

N° d'ordre :

N° de série :

**République Algérienne Démocratique et Populaire**



**Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique**

**UNIVERSITÉ ECHAHID HAMMA LAKHDAR D'EL OUED**

**FACULTÉ DES SCIENCES EXACTES**

**DEPARTEMENT D'INFORMATIQUE**

**Mémoire de fin d'étude**

**MASTER ACADEMIQUE**

Domaine: Mathématiques et Informatique

Filière: Informatique

Spécialité: Systèmes Distribués et Intelligence Artificiel (SDIA)

**Thème**

**Construction d'une ontologie à partir d'une base  
de données pour la mise en oeuvre des  
mécanismes d'interopérabilité sémantique :  
Application sur un progiciel de gestion intégrée.**

Présenté par: - Kamel Maaloul  
- Laïche Largot

Soutenu devant le jury composé de

M.ZOUBEIDI MEROUANE	MA(B)	Rapporteur	Univ. d'El Oued
M. GHENABZIA AHMED		Co.Rapporteur	Univ. d'El Oued
M.LADJDEL BRAHIM	MAA	Président	Univ. d'El Oued
M.LAAOUID ABDELKADER	MAA	Examineur	Univ. d'El Oued

Année universitaire 2016-2017.

## Remerciement

*Nous remercions tout d'abord notre Dieu qui nous a donné  
la force et la volonté pour élaborer ce travail.*

*Nous adressons nos vifs remerciements à notre encadreur  
**‘Mr Merouane Zoubeidi ‘**, qui n’a pas cessé de nous  
donner les conseils et les bonnes orientations et nous prive  
pas de son temps et aussi on remercie **‘Ghenabzia Ahmed  
‘**, qui nous écoute avec grande patience pendant toute la  
période de préparation malgré ses contraintes.*

*Sans son aide, notre travail n'aurait pas vu la lumière.  
Notre reconnaissance va aussi à tous ceux qui ont collaboré  
à notre formation en particulier les enseignants du  
département d'Informatique,  
Universitaire Echahide Hamma Lakhdar  
El-Oued*

*Aussi à nos collègues de la promotion 2016-2017 On  
remercie également tous ceux qui ont participé de près ou  
de loin à élaborer ce travail*



## Résumé

Le PGI (Progiciel de Gestion Intégrée) est devenu la pierre angulaire des systèmes d'information des entreprises. Cet outil permet une gestion homogène et cohérente du système d'information (SI) de l'entreprise. On peut donc définir une solution de gestion intégrée tels que *SAGE*, *SAP*, *ORACLE* ...également dénommée ERP (Entreprise Resource Planning) ou PGI comme étant :

- Une solution logicielle qui permet de gérer plusieurs processus fonctionnels ou opérationnels d'une organisation (gestion des RH, gestion comptable et /ou financière, des achats, des approvisionnements, de la production, des stocks, des vente, de la relation client, de la chaîne logistique, etc.).
- Les processus gérés sont intégrés de manière modulaire (module fonctionnel par module fonctionnel), tout en partageant une base de données unique et commune.
- La solution doit couvrir au moins deux domaines fonctionnels différents (par exemple, RH et comptabilité/finance, ou encore comptabilité/finance et gestion commerciale...).

L'objectif de ce mémoire est de développer une partie de l'ontologie d'un ERP tel que *SAGE*. Pour ce faire, il est utile de commencer par la construction des ontologies des domaines pour chaque module fonctionnel. Nous souhaitons accomplir des manipulations intra-ontologiques, consistant à un ensemble d'actions de mise à jour, leur manipulation, leur interrogation en prenant en compte leur consistance.

## **Mots clés :**

PGI (ERP), Ontologie, Interopérabilité sémantique.

## **Abstract**

PGI (Software package of Integrated Management) became the cornerstone of the systems of information of firms. This tool allows a homogeneous and consistent management of the system of information (IF) from the firm. They can therefore define a resolution of integrated management such as SAGE, SAP, ORACLE also called ERP (Undertaken Resource Planning) or PGI as being:

- A software resolution which allows to manage several functional or operational processes of an organization (management of RH, accounting management and / or financial, of purchases, of supplies, of production, of supplies, of blows, relation customer, logistical chaine, etc).
- Managed processes are inserted in a modular way (functional module by module functional), while sharing a common and unique database.
- Resolution must cover two different functional domains at least (for instance, RH and accountancy / finance, or else accountancy / finance and commercial management ...

Objective of this memoryis to develop a party of the ontology of ERP such as SAGE. In order to do that, it is useful to begin with the building of the ontologies of domains for every functional module. We like to fulfil intra-ontological manipulations, consisting in-group of actions of updating, their manipulation, their questioning by taking into account their consistency.

## **Keywords:**

PGI (ERP), Ontology, semantic Interopérabilité.

## الملخص

أصبحت أداة تخطيط موارد المؤسسات (الإدارة المتكاملة للبرمجيات) هي الحجر الأساس لنظم المعلومات للشركات. تقوم هذه الأداة بالتسيير المتجانس والمتناسك لنظام المعلومات لإدارة الشركة. ولذلك يمكننا تحديد حل لإدارة متكاملة مثل أوراكل، ساب،...

تعرف أيضا باسم (تخطيط موارد المؤسسة) أو تخطيط موارد المؤسسات بأنها:

- حل برنامج الذي يسمح لك بإدارة عدة الوظيفية أو التشغيلية المنظمة عملية (الإدارة، والمحاسبة والموارد البشرية و/أو المالية، المشتريات، اللوازم، الإنتاج، مخزون، مبيعات، وعلاقة العميل (، توريد سلسلة، إلخ).
- يحتوي تدار العمليات تتكامل بطريقة نمطية (وحدة وظيفية بوحدة وظيفية)، أثناء مشاركة قاعدة بيانات مشتركة واحدة.
- يحتوي الحل يجب أن يشمل اثنين على الأقل من مختلف المجالات الوظيفية (على سبيل المثال، الموارد البشرية، والمالية، أو حتى المالية والإدارة...

والغرض من هذه المذكرة هي تطوير جزء من الانطولوجيا لنظام تخطيط موارد المؤسسات. وللقيام بذلك لابد من البدء ببناء الانتولوجيا للمجالات لكل وحدة وظيفية. كما نريد تحقيق الانطولوجيا العامة التي تتكون من مجموعة من الإجراءات الحديثة لكن الاستعلام، مع مراعاة توافقها. الكلمات الرئيسية: تخطيط موارد المؤسسات (ERP)، علم الوجود، إمكانية التشغيل المتداخل الدلالية

## الكلمات المفتاحية:

تخطيط موارد المؤسسات، الأنطولوجيا، العمل المشترك الدلالي.

## Liste de tableaux

Table I.1: Comparaison des méthodologies de construction d'ontologies .....	11
Tableau II.1: Principaux ERP (Source: AMR Research, oct. 2016).....	42
Tableau III.1: les classes et leur description.....	55
Tableau III.2: Les Propriétés de Type de Données et leur domain.range.....	55/56
Tableau III.3: Les propriétés d'objets et leur domain.range .....	57

## ***LISTE DES FIGURES***

Figure I. 1 : Cycle de vie d'une ontology dans Methontology .....	13
Figure I. 2 : Approche d'une ontology simple.....	21
Figure I. 3 : Approche d'ontology multiple.....	22
Figure I. 4 : Approche d'ontology hybride.....	22
Figure II.1 : Organisation d'un ERP .....	31
Figure II.2 : Architecture technique d'un ERP .....	34
Figure II.3 : Architecture modulaire ERP .....	35
Figure II.4 : Processus de module logistique (Negoce).....	36
Figure II.5 : Processus de commande de vente .....	37
Figure II.6 : Processus stock .....	38
Figure II.7 : Réapprovisionnement sur seuil .....	38
Figure II.8 : Réapprovisionnement périodique.....	39
Figure II.9 : Processus achat.....	39
Figure II.10 : Processus de commande d'Achat .....	40
Figure II.11 : Répartition des licences ERP par secteur d'activité en 2005 .....	42
Figure III.1 : Fonctionnalités d'Odoo.....	49
Figure III.2 : L'éditeur d'ontology Protégé.....	53
Figure III.3 : Graphe OWL.....	57
Figure III.4 : Développement mapping en JenaAPI.....	58
Figure III.5 : Résultat du mapping(1).....	59
Figure III.5 : Résultat du mapping(2).....	59

## Sommaire

Introduction général .....	1
----------------------------	---

### **ChapitreI: Ontologies: état de l'art et intégration**

1. Introduction .....	5
2. Définitions d'une ontologie .....	5
3. Pourquoi les ontologie ? .....	7
4. Notion de base .....	8
4.1 Les concepts .....	8
4.2 Les rôles .....	8
4. 3 Les axiomes .....	8
5. Développement des ontologies .....	9
5. 1 Principes de développement.....	9
5. 2 Méthodologies de développement.....	10
5. 2. 1 Tove .....	11
5. 2. 2 Enterprise .....	12
5. 2. 3 Methontology .....	12
6. Formalismes de représentation .....	14
6.1 Frames .....	14
6. 2 Graphes conceptuels .....	15
6. 3 Logiques de description .....	15
7. Langages de spécification d'ontologies .....	15
7 .1 KIF.....	16
7. 2 KL-ONE.....	16
7. 3 RDF(S) Resource Description Framework and RDF schema .....	16
7. 4 OIL (Ontology Interchange Language) .....	16
7. 5 DAML+OIL (DARPA Agent Markup Language +OIL) .....	17
7. 6 OWL (Web OntologyLanguage) .....	17
7. 7 Modèle orienté objet.....	18
8. Environnement et outils de modélisation des ontologies .....	19
9. Systèmes de raisonnement.....	21
10. Intégration des ontologies .....	21
10. 1 Alignement des ontologies .....	23

10. 2 Fusion des ontology .....	23
10. 3 Mapping des ontologies .....	23
11. Hétérogénéité .....	25
12. Conclusion .....	27

## **ChapitreII: Généralité sur le progiciel de gestion intégré (ERP)**

1. Introduction .....	29
2. Le système intégré de gestion .....	29
2.1. La gestion intégrée.....	29
2.2. Le principe de l'intégration.....	30
3. L'ERP, Historique et définitions.....	31
3.1. Historique de l'ERP .....	31
3.2. Définition de l'ERP .....	31
3.3. Organisation d'un ERP.....	31
3.4. Caractéristiques d'un ERP.....	32
4. Les ERP: Pour qui? Pourquoi? .....	33
4.1. Les ERP: Pour qui?.....	33
4.2. Les ERP: Pourquoi?.....	33
4.3. Intérêts de l'ERP dans l'entreprise.....	34
5. Architecture technique d'un ERP.....	35
6. Architecture modulaire d'un ERP .....	35
6.1. Le module finance.....	36
6.2. Le module logistique .....	36
6.3. Module e-commerce.....	41
7. Principaux éditeurs d'ERP leurs parts de marché.....	41
7.1. Les ERP propriétaires .....	41
7.2. Les ERP Open Source .....	42
8. Domaine d'application d'un ERP .....	43
9. Les difficultés de l'implantation.....	43
10. Les étapes de l'implantation de l'ERP .....	44
11. Positionnement des ERP par rapport aux nouveaux concepts.....	44
11.1. L'ERP et le SCM .....	45
11.2. L'ERP et le E-business.....	45
11.3. L'ERP et le CRM .....	45
11.4. L'ERP et L'EAI .....	45
12. Exemple d'un ERP : Tiny ERP .....	46
12.1. Présentation de Tiny ERP .....	46
12.2. Module de Tiny ERP.....	46
12.3. Avantage Tiny ERP.....	46
13. Conclusion.....	47

## **Chapitre III: Implémentation et réalisation**

I - Introduction .....	49
II - Odoo.....	49

1-Odoo : Définition .....	49
2 - Odoo : fonctionnalités .....	50
3 - Pourquoi choisir Odoo ?.....	50
III -module RH et RH holidays.....	51
1 - module RH.....	51
2 - module RHHolidays.....	51
3 – Classe UML.....	52
IV –Présentation Sémantique OWL.....	53
IV –1 – Protégé .....	53
IV –2 – Les Classes (concepts) .....	53
IV –3 – Les Propriétés de Type de Données .....	55
IV –4 – Les Propriétés d’Objets .....	57
IV –5 – Graphe OWL.....	58
IV –6 – Mapping RDB TO OWL.....	59
V - Conclusion.....	61
Conclusion Générale.....	62
Bibliographie.....	63

## **1- Contexte général**

L'évolution technologique d'une part et les mutations économiques d'autre part ont bouleversé le domaine des systèmes d'information. Pour profiter de cette nouvelle technologie (réseau, multimédia, équipements mobiles, etc.), les techniques de modélisation des ERP et des données en particulier ont connu-elles aussi une évolution et de nouveaux concepts sont apparus : Bases de données objets, bases de données distribuées, Data Warehouse, systèmes d'information géographiques, systèmes d'information multimédias, systèmes d'information coopératifs, etc.

Cette variété de technologies a donné naissance à des applications dont les sources de données sont hétérogènes : fichiers classiques, bases de données navigationnelles (hiérarchiques et codées), bases de données relationnelles, bases de données objets et des données avec des structures plus complexes telles que les images, les séquences vidéo, données géographiques, etc.

Dans le nouveau contexte économique, les entreprises se fusionnent et adhèrent à des groupes pour préserver leurs intérêts communs et faire face aux nouvelles contraintes imposées par le marché. Cette restructuration et réorganisation a un impact direct sur le ERP. Pour que ces entreprises puissent communiquer et coopérer de manière efficace, leurs systèmes d'information propriétaires doivent continuer à fonctionner mais doivent coopérer eux aussi. La coopération des ERPs est devenue aujourd'hui plus qu'une nécessité vu le besoin d'échanger des informations dans un environnement ouvert et dynamique caractérisé par l'utilisation d'Internet, messagerie électronique, téléphonie mobile, etc.

L'interopérabilité est considérée comme un moyen, parmi d'autres, pour assurer la coopération des ERPs.

## **2- Problématique**

Le système d'information est construit grâce à plusieurs méthodes et technologies : plateformes, protocoles de communication, environnements et outils de développement, modèles de conception des données, types de SGBD, etc., la complexité pour faire interopérer des systèmes d'information réside au niveau de cette diversité de méthodes de conception et de technologies d'implémentation.

Pour les données en particulier, les systèmes d'information disposent de sources de données très intéressantes couvrant l'ensemble des domaines d'activité et pourront servir pour développer des applications d'analyse qui exploitent cette richesse d'informations distribuées sur ces différents systèmes d'information. Néanmoins, ces

sources de données préexistantes ont été construites dans des contextes différents, avec des taxonomies différentes, des choix de conception différents, des modèles de données différents et des systèmes de gestion de bases de données différents

Une coopération entre ces différents systèmes propriétaires peut être considérée elle-même comme un système d'information dont le contenu est hétérogène et dispersé géographiquement.

De manière générale, la problématique peut se résumer comme suit : comment traiter une requête dont les éléments de réponse se trouvent dispersés sur différents ERPs ?

Différentes formes d'hétérogénéité apparaissent dans le processus d'interopérabilité: hétérogénéité matérielle due à la variété et l'incompatibilité des plateformes matérielles, l'hétérogénéité logicielle provoquée par la présence de systèmes d'exploitations différents (Unix, Linux, Windows, Solaris, etc.) et des outils d'implémentation différents (PHP, JAVA,C++, .NET, etc.), hétérogénéité des SGBD implique une divergence au niveau des systèmes de gestion de bases de données sur lesquels ont été implantées les sources de données (Oracle, SQL Server, DB2, Informix, Interbase, etc.).

Dans ce mémoire, nous nous intéressons particulièrement à L'interopérabilité sémantique qui vise à tenir compte de la dimension "sémantique" ou "sens" des données et des services disponibles dans les différents systèmes d'information mérite encore une meilleure maîtrise. Malgré le nombre important de travaux effectués, les solutions d'interopérabilité sémantique proposées restent insuffisantes et la problématique continue à rester d'actualité.

### **3- Objectifs**

Comme la sémantique est un aspect important dans la coopération des systèmes d'information, l'objectif est donc de faire un état de l'art sur les solutions proposées pour résoudre les conflits sémantiques, d'évaluer ces approches sur un certain nombre de critères dont la modélisation de la sémantique, de faire un bilan et de proposer une nouvelle approche qui prend en considération l'importance de la sémantique. En effet, les ontologies, qui ont connu une prolifération ces dernières années, constituent un outil intéressant qui valorise la sémantique des informations et qui permet la réconciliation des différents systèmes sur un vocabulaire commun.

## **4- Travail du mémoire**

Le travail de mémoire consiste à présenter une approche pour Construction d'une ontologie à partir de base de données pour la mise en oeuvre de mécanismes d'interopérabilité sémantique : Application sur un progiciel de gestion intégrée.

- Nous avons choisi d'introduire une ontologie dans chaque ERP pour la représentation de son contexte local. Chacune des ontologies locales est connectée vers une ontologie universelle qui jouera le rôle de dictionnaire commun aux différents systèmes, ce qui permet de modéliser la sémantique de manière plus précise et permet aux différents systèmes d'avoir un vocabulaire commun. Au niveau global, nous proposons la construction d'une ontologie globale, exprimant le domaine d'application par fusion des ontologies locales.

## **5- Organisation du mémoire**

Le présent mémoire est structuré comme suit :

**Chapitre 1 :** nous décrivons la notion d'ontologie, les méthodologies de développement (Tove, Enterprise et Methontology), les formalismes de représentation (les frames, les graphes conceptuels et les logiques de description) et les langages de spécification (KIF, KL-ONE, RDF, RRD(S), OIL, DAML-OIL et OWL). Ensuite, nous mentionnons les approches d'intégration des ontologies tout en citant le problème d'hétérogénéité.

**Chapitre 2 :** présentation générale sur le Progiciel de Gestion Intégré ERP, commençant par les définitions, les intérêts et ses avantages dans l'entreprise, ainsi que les différentes architectures et ses domaines d'application.

**Chapitre 3 :** dernier chapitre présente la conception et l'implémentation de système réalisé.

Enfin, nous concluons ce projet par une conclusion générale.

## ***Chapitre I :***

# ***Les ontologies : état de l'art et intégration***

## **I - Introduction**

Nées des besoins de représentation des connaissances, Les ontologies sont étudiées par les chercheurs travaillant en Intelligence Artificielle, sur la représentation de la connaissance, et maintenant, sur le Web sémantique et sont à l'heure actuelle au cœur des travaux menés en Ingénierie des Connaissances (IC). Visant à établir des représentations à travers lesquelles les machines puissent manipuler la sémantique des informations, la construction des ontologies demande à la fois une étude des connaissances humaines et la définition de langages de représentation, ainsi que la réalisation de systèmes pour les manipuler.

À l'heure actuelle, les ontologies constituent une solution très prisée quand à la capitalisation et au partage de connaissances dans le domaine des environnements informatiques. Elles aient un rôle normatif analogue aux terminologies mettre en place un vocabulaire commun et faire usage de représentations et concepts partagés, afin de permettre l'interopérabilité des documents et faciliter l'élaboration de connaissances, leur approche formelle de la sémantique les distingue également de manière nette de ces dernières. Les ontologies sont des architectures de concepts, non des listes organisées de termes.

Au fur et à mesure des expérimentations, des méthodologies de construction d'ontologies et des outils de développement adéquats sont apparues. Émergeant des pratiques artisanales initiales, une véritable ingénierie s'est constituée autour des ontologies, ayant pour but leur construction mais plus largement leur gestion tout au long d'un cycle de vie. Les ontologies apparaissent ainsi comme des composants logiciels s'insérant dans les systèmes d'information en leur apportant une dimension sémantique qui leur faisait défaut jusqu'ici.

Dans ce chapitre, nous décrivons la notion d'ontologie, les méthodologies de développement (Tove, Enterprise et Methontology), les formalismes de représentation (les frames, les graphes conceptuels et les logiques de description) et les langages de spécification (KIF, KL-ONE, RDF, RRD(S), OIL, DAML-OIL et OWL). Ensuite, nous mentionnons les approches d'intégration des ontologies tout en citant le problème d'hétérogénéité.

### **1. Définitions d'une ontologie**

**Ontologie** est un terme grec composé des mots :

« Ontos= être » et « Logia=discours (langage+raison) ».

Bien que la communauté d'I.A. semble s'accorder sur l'utilisation et le sens du mot « ontologie », il n'existe pas de définition formelle unanimement acceptée. Une ontologie est vue comme un moyen de « décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances ».

En informatique et d'après l'encyclopédie Wikipédia, « une ontologie est un ensemble structuré de concepts. Les concepts sont organisés dans un graphe dont les relations peuvent être :

- des relations sémantiques ;
- des relations de composition et d'héritage (au sens objet). »

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné.

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

De nombreuses définitions ont été proposées pour donner le sens exact au terme ontologie. Ce mot est utilisé dans différents contextes avec des définitions variées, mais aucune de ces définitions ne s'est explicitement imposée. Nous avons recensé les définitions suivantes :

[27], a définit une ontologie comme suit :

« Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire » (01).

Dans le cadre d'IA, neches montre que l'élaboration d'une ontologie s'effectue par des directives relativement floues. Ce qui signifie qu'on peut utiliser des règles pour combiner les termes et les relations entre eux pour déduire d'autres termes et relations.

Selon Gruber, « Une ontologie est une spécification explicite et formelle d'une conceptualisation partagée» (14).

« Ontologie » est un terme emprunté à la philosophie qui implique une branche de la philosophie qui traite la nature et l'organisation de la réalité. Plusieurs définitions d'ontologies sont données dans la dernière décade, mais celle qui caractérise l'essentiel d'une ontologie est fondée sur la définition reliée dans [08] : « Une ontologie est une **spécification formelle**, et **explicite** d'une **conceptualisation partagée** ». Conceptualisation réfère à un modèle abstrait de certains phénomènes dans le monde qui identifie les concepts appropriés de ce phénomène. Explicite signifie que le type de concepts utilisés et les contraintes sur leur utilisation doivent être explicitement définis. Formelle réfère au fait qu'une ontologie doit être compréhensible par la machine, c'est-à-dire cette dernière soit capable d'interpréter la sémantique de l'information fournie. Partagée indique que l'ontologie supporte la connaissance consensuelle, et elle n'est pas restreinte à certains individus mais accepté par un groupe [09].

Grüniger [46], définit une ontologie selon les composants de base, les entités et les propriétés. Cette permet de distinguer une ontologie qui est un ensemble définie de concepts et leurs relations, d'un modèle qui est un ensemble d'instance de ces concepts.

« Une ontologie est une description formelle d'entités et leurs propriétés, relations, contraintes et comportement » (1995).

En 1997, Guarino [173] accentue l'ambiguïté de la conceptualisation. La spécification des ontologies est partielle car une conceptualisation ne peut toujours être entièrement formalisée dans un cadre logique, du fait d'ambiguïtés ou qu'aucune représentation de leur sémantique n'existe dans le langage de représentation choisi.

« Les ontologies sont des spécifications partielles et formelles d'une conceptualisation commune » (1997).

D'autres définitions se basent sur la notion de partage des connaissances liés à un domaine :

- " Une ontologie est une description concise et non ambiguë de principales entités appropriées d'un domaine d'application et de leurs relations potentielles entre eux." (S. Schulze-Kremer).

- " Une ontologie est un vocabulaire commun pour les chercheurs qui doivent partager les informations dans un domaine (Noy et al.)

"Une ontologie est une spécification d'un vocabulaire des symboles non-logiques et inclut : le type des entités, les attributs et les propriétés, les relations et les fonctions, et les contraintes." (R. Fikes).

Nous concluons que les définitions du terme ontologie adonnent dans la littérature scientifique. Les définitions dans leur diversité offrent des points de vue à la fois différents et complémentaires.

Les taxonomies ou les thésaurus sont aussi des ontologies car elles se cantonnent à décrire des liens sémantiques du type "est une sorte de" et son inverse "est représenté par" ou, plus spécifiquement, "est une sous-classe de." Des ontologies plus complexes permettent la représentation de liens sémantiques plus spécifiques, par exemple, "est localisé dans,". Mais surtout les ontologies les plus abouties permettent également l'intégration de propriétés particulières, de règles d'utilisation et de contraintes.

## **2. Pourquoi les ontologie?**

Les ontologies ont pour objectif de représenter de façon générique et réutilisable la sémantique d'un domaine [04]. A l'origine, elles fournissent pour un domaine particulier, une compréhension commune utilisable pour partager de la connaissance ou des données. Actuellement,

Une ontologie définit un vocabulaire commun pour les chercheurs qui ont besoin de partager l'information dans un domaine. Elle inclut des définitions lisibles en machine des concepts de base de ce domaine et de leurs relations.

Pour quelles raisons développer une ontologie ? En voici quelques-unes [11] :

- Partager la compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels.
- Permettre la réutilisation du savoir sur un domaine.
- Expliciter ce qui est considéré comme implicite sur un domaine.
- Distinguer le savoir sur un domaine du savoir opérationnel.
- Analyser le savoir sur un domaine.

### 3. Notion de base

Une ontologie contient les primitives terminologiques du domaine (le vocabulaire conceptuel) ainsi que des axiomes qui restreignent l'interprétation des primitives. Le vocabulaire conceptuel est structuré en un ensemble de concepts et un ensemble de relations existantes entre ces concepts.

#### 3.1 Les concepts

Les concepts sont appelés aussi termes ou classes de l'ontologie, correspondent aux abstractions pertinentes d'un segment de la réalité (le domaine du problème), retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie. Selon [13] ces concepts peuvent être classifiés selon plusieurs dimensions :

- 1) niveau d'abstraction (concret ou abstrait) ;
- 2) atomicité (élémentaire ou composée) ;
- 3) niveau de réalité (réel ou fictif).

Un concept peut représenter un objet matériel, une notion, une idée.

#### 3.2 Les rôles

Les rôles traduisent les associations (pertinentes) existant entre les concepts présents dans le segment analysé de la réalité. Ces relations incluent les associations suivantes: 1) Sous-classe-de (généralisation – spécialisation) ; 2) Partie-de (agrégation ou composition) ; 3) Associée-à ; 4) Instance-de, etc. Ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres.

Un rôle est caractérisé par un terme, une extension qui est l'ensemble des tuples d'instances liés par cette relation, et une intension qui spécifie d'une part les concepts pouvant être liés par cette relation et d'autre part la sémantique d'usage de la relation.

### 4.3 Les axiomes

Les axiomes constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine traduites par l'ontologie.

Les axiomes sont employés pour modéliser les phrases qui sont toujours vraies. Ils peuvent être inclus dans une ontologie pour plusieurs buts, tels que définir la signification de composants d'ontologie, définir les contraintes complexes sur les valeurs des attributs, les arguments de relations, etc., vérifier l'exactitude d'informations indiquées dans l'ontologie ou déduire une nouvelle information.

L'utilisation des axiomes sert à définir le sens des entités, mettre des restrictions sur la valeur des attributs, examiner la conformité des informations spécifiées ou en déduire de nouvelles.

## 5. Développement des ontologies

La construction d'une ontologie va donc consister d'une part, à identifier les concepts d'un domaine de connaissance et la sémantique qui leur est associée et, d'autre part, à représenter cette sémantique de façon indépendante des diverses applications dans lesquelles pourra être utilisée l'ontologie. En effet, une ontologie n'est pas opérationnelle, au sens où elle n'inclue pas de mécanismes de raisonnement, puisqu'elle doit justement être indépendante de tout objectif opérationnel. Le langage cible doit donc permettre de représenter les différents types de connaissances (connaissances terminologiques, faits, règles et contraintes) et de manipuler ces connaissances à travers des mécanismes adaptés à l'objectif opérationnel du système conçu.

### 5.1. Principes de développement

Il existe un ensemble de critères et de principes qui ont fait leurs preuves dans le développement des ontologies et qui peuvent être résumés comme suit [06] :

- **Clarté et Objectivité** : L'ontologie doit fournir la signification des termes définies en fournissant des définitions objectives ainsi qu'une documentation en langue naturelle.
- **Complétude** : Une définition exprimée par des conditions nécessaires et suffisantes est préférée à une définition partielle (définie seulement par une condition nécessaire et suffisante).
- **Cohérence** : Une ontologie cohérente doit permettre des inférences conformes à ces définitions.
- **Extensibilité monotone maximale** : De nouveaux termes généraux et spécialisés devraient être inclus dans l'ontologie d'une façon qui n'exige pas la révision des définitions existantes.
- **Engagements ontologiques minimaux** : Ce principe invite à faire aussi peu de réclamations que possible au sujet du monde représenté. L'ontologie devrait spécifier le moins possible la signification de ses termes, donnant aux parties qui s'engagent dans cette ontologie la liberté de spécialiser et d'instancier l'ontologie comme elles le désirent.
- **Principe de distinction ontologique** [10] : les classes dans une ontologie devraient être disjointes. Le critère utilisé pour isoler le noyau de propriétés considérées comme invariables pour une instance d'une classe est appelé le critère d'Identité.
- **Modularité** [05] : Ce principe vise à minimiser les couplages entre les modules.
- **Diversification des hiérarchies** [05] : Ce principe est adopté pour augmenter la puissance fournie par les mécanismes d'héritage multiple. Si suffisamment de connaissances sont représentées dans l'ontologie et que suffisamment de différentes classifications de critères sont utilisées, il est plus facile d'ajouter de nouveaux concepts (puisque'ils peuvent être facilement spécifiés à partir des concepts et des classifications

de critères pré-existants) et de les faire hériter de propriétés de différents points de vue.

**-Distance sémantique minimale** [05]. Il s'agit de la distance minimale entre les concepts enfants de mêmes parents. Les concepts similaires sont groupés et représentés comme des sous-classes d'une classe, et devraient être définis en utilisant les mêmes primitives, considérant que les concepts qui sont moins similaires sont représentés plus loin dans la hiérarchie.

**--Normaliser les noms** [05]. Ce principe indique qu'il est préférable de normaliser les noms aussi autant que possible.

Cet ensemble de critères est généralement accepté pour guider le processus d'ingénierie ontologique.

## **5.2.Méthodologiesdedeveloppement**

Le développement des ontologies est une activité de modélisation qui est habituellement effectuée par les ingénieurs ontologique (également appelés les ontologists) qui ont la compréhension suffisante du domaine. Il est facile avec des langues de représentation de la connaissance. Quant à la conception des systèmes basés sur la connaissance, l'aide des experts du domaine est exigée pour construire et valider des ontologies.

La construction d'une ontologie est une activité complexe et longue. Par conséquent, les méthodologies ont émergé basé sur les expériences acquises dans la construction de grandes ontologies. Visent à faire au développement des ontologies plus un processus de technologie plutôt qu'à un art. Un aperçu récent des méthodologies de développement d'ontologies est présenté dans [11]. Les méthodologies plus complètes pour le développement d'ontology sont récapitulées dans la Table 2. 1. à l'exception de la méthode de capture d'ontology IDEF5 [09] qui est une partie de la méthodologie de modélisation d'entreprise complète IDEF (Integrated Computer Aided ManufacturingDEFinition), les autres méthodologies résultent des projets importants visant à construire de grandes ontologies :

<b>TOV</b>	<b>Enterprise</b>	<b>Methontology</b>	<b>IDEF</b>
Scenarios de motivation	objectif: niveau de formalité	Objectif et portée	Objectif et besoins
Besoins (questions informelles de compétence)	objectif : masse d'information i.e. liste des concepts appropriés	acquisition de connaissance par l'analyse des textes et des interviews d'expert	objectif : limites de l'ontologie collection des données basée sur les techniques
specification formelle des objects, attributs and relations	definition formelle des termes	Conceptualisation : spécification des concepts, instances, relations basée sur	Analyse de données : produire une listedes objets.
Besoins formels (questions formelles de compétence)		Intégration à partir des autres ontologies implémentation	Ontologie initiale Développement : définition des "proto- concepts"
specification des axioms	Formal specification of		raffinement de l'ontologie dans un
Evaluation	Évaluation	Evaluation + documentation	Teste andvalidation

**Table I.1 : Comparaison des méthodologies de construction d'ontologies.**

- Projet TOVE (Toronto Virtual Enterprise) project [05],
- L'ontologie Enterprise [Uschold and Gruninger, 05],
- La methodeMethontology [07].

Les étapes principales qui peuvent être dérivées de ces méthodologies comprennent ce qui suit :

- Identification de la tâche pour laquelle l'ontologie est développée ;
- Définition des besoins de l'ontologie : objectif et portée ;
- Spécifications informelle : construire une spécification informelle des concepts ;
- Codage : Représenter formellement les concepts et les axiomes dans un langage ;
- Évaluation de l'ontologie.

L'extensibilité et la maintenance des ontologies sont considérées par la plupart des auteurs mais ne sont pas une partie des méthodologies.

### 5.2.1 Tove

TOVE (Toronto Virtual Enterprise) développé par l'université de Toronto. Cette méthodologie repose sur les expériences de développement d'une entreprise [04]

[03]. Elle s'appuie également sur les principales étapes suivantes pour le développement d'une ontologie :

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

## 5. 2. 2Enterprise

Uschold et Grüninger proposent le squelette d'une méthode basé sur l'expérience de construction d'ontologies dans le domaine de la gestion des entreprises [02]. La méthode Enterprise repose sur les quatre (4) étapes suivantes :

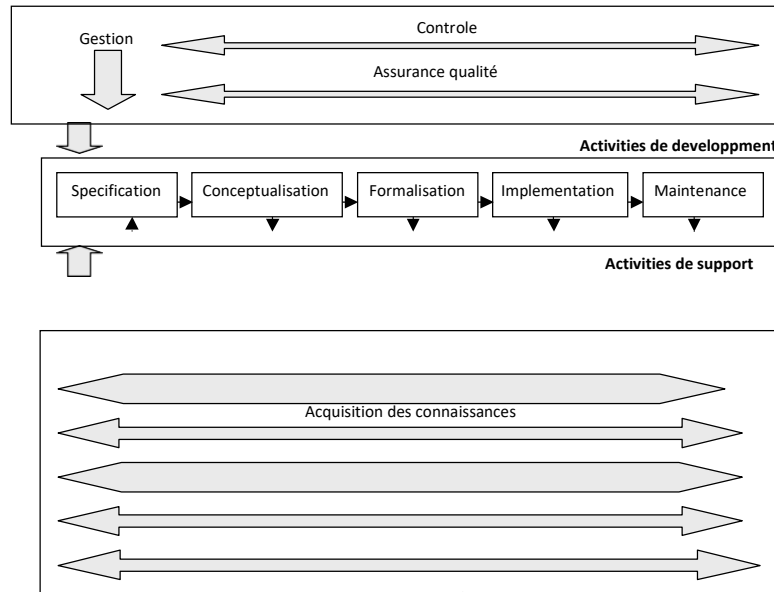
- Identifier le rôle et la portée de l'ontologie.
- Réaliser l'ontologie où les activités suivantes sont distinguées : identifier les concepts et les relations fondamentaux, définir provisoirement ces éléments, coder l'ontologie dans un langage adapté, intégrer les ontologies existantes.
- Evaluer l'ontologie.
- Rédiger une documentation et une trace des actions réalisées lors des différentes étapes.

Les étapes de la méthode Enterprise sont décrites de façon abstraite. Les techniques utilisées pour les activités ne sont pas précises (ex : comment identifier les concepts fondamentaux ? quel langage utilisé pour représenter l'ontologie ?).

## 5.2. 3Methontology

Methontology se situe entre le GL (Génie Logiciel) et l'IC (Ingénierie des Connaissances) [06] [07]. Elle identifie une séquence d'activités de gestion, de développement et de support pour le développement de l'ontologie. Dans Methontology, nous distinguons les étapes (cf. figure 2. 1) :

### Activités de gestion



**Figure I. 1 : Cycle de vie d'une ontology dans Methontology.**

- Spécification : Le développement d'une ontology commence par la définition de son domaine et de son porté. Cela est basé sur la réponse à certaines questions :
  - Quel est le domaine que couvre l'ontology ?
  - A quoi servir cette ontologie ?
  - Quels types de questions l'ontology doit fournir des réponses ?
  - Qui va utiliser l'ontology ?
  - etc.

Les reponses à ces questions peuvent changer durant le processus de developpement de l'ontology. Elles permetent de limiter la portée de l'ontology.

- Conceptualisation : Elle consiste à identifier et à structurer les connaissances du domaine à partir des sources d'informations. L'acquisition de ces connaissances peut s'appuyer à la fois sur l'analyse de documents et l'interview des experts du domaine. Une fois que les concepts sont identifiés par leurs termes. La semantique de ces concepts est décrite par un langage semi-formel à travers leurs propriétés et leurs instances.
- Formalisation : Cette étape consiste à formaliser le modèle conceptuel obtenu dans l'étappécédentepar un formalisme de représentation d'ontooogie telle que DL.
- Implementation : Cette étape requiert le choix d'un langage permettant de représenter les connaissances terminologiques et les axiomes puis de manipuler ces représentations pour raisonner.
- Maintenance : il s'agit d'une maintenance corrective ou évolutive (nouveaux besoins de l'utilisateur), ce qui permet l'évolution et la validation de l'ontology. Cette activitée est réalisée par le constructeur et des experts du domaine. La

validation se base sur les services d'inférences associés au DL, offerts par des raisonneurs.

## 6. Formalismes de représentation

Les modèles de représentation de connaissance utilisés en ingénierie ontologique peuvent être regroupés selon les paradigmes conceptuels qu'ils réifient. Sont ainsi distingués :

- a) Les graphes
  - i) Les frames
  - ii) Les graphes conceptuels.
  - iii) Orienté-Web : RDF et RDF Schéma
- b) Logique
  - i) Du premier ordre : KIF
  - ii) Descriptive: KL-One, OIL, DAML+OIL, OWL
- c) Orienté objet : UML + OCL

Chacun de ces modèles de représentation est implémenté dans un ou plusieurs langages implémentant une partie ou la totalité du modèle, en particulier des langages adaptés à la sémantique. Les langages implémentant ces modèles sont souvent opérationnels, c'est-à-dire qu'ils offrent des mécanismes de raisonnement, et servent alors à représenter des ontologies déjà opérationnelles. Le modèle de la logique de description étant celui que nous utilisons dans nos travaux, nous en détaillerons la présentation.

### 6.1 Frames

Le modèle des Frames est un classique de l'Intelligence Artificielle et a été initialement proposé comme langage de représentation d'ontologies par T. GRUBER [48]. Le principe de ce modèle est de décomposer les connaissances en classes (ou frames) qui représentent les concepts du domaine. À un frame est rattaché un certain nombre d'attributs (slots), chaque attribut pouvant prendre ses valeurs parmi un ensemble de facettes (facets) [10]. Une autre façon de présenter ces attributs est de les considérer comme des relations binaires entre classes dont le premier argument est appelé domaine (domain) et le deuxième portée (range) [15]. Des instances des classes, correspondant à l'extension de chaque concept, peuvent être ajoutées, ainsi que des fonctions qui sont des types particuliers de relations liant un ensemble de classes à une valeur calculée à partir des valeurs des attributs des classes. La spécification de propriétés conceptuelles des attributs (ou relations) recourt à des formules de la logique du premier ordre. La sémantique de la subsomption est purement extensionnelle : un frame F1 est plus spécifique qu'un frame F2 si toute instance de F1 est instance de F2. F-Logic est l'exemple le plus connu de langage opérationnel à base de frames [08]. L'Open Knowledge Base Connectivity (OKBC [11]), protocole et API de requête et d'interfaçage entre bases de connaissances, utilise également le modèle de frame. Le KnowledgeInterchange Format (KIF) est un langage, non opérationnel, implémentant le

modèle des frames en logique du premier ordre pour la représentation de connaissance [12].

## 6. 2 Graphes conceptuels

Introduit par J. SOWA au début des années 80 [14], le modèle des GraphesConceptuels (GC) appartient à la famille des réseaux sémantiques [14]. Les réseaux sémantiques modélisent les connaissances sous forme de graphes, les noeuds étant associés à des concepts et les arêtes à des relations. Ce modèle se prête bien à des présentations graphiques des connaissances mais présente, comme nous allons le voir, certains problèmes particuliers de représentation.

Le modèle des GCs se décompose en deux (2) parties :

- Une partie terminologique dédiée au vocabulaire conceptuel des connaissances à représenter, c'est-à-dire les types de concepts<sup>3</sup>, les types de relations et les instances des types de concepts<sup>4</sup>. Cette partie correspond à la représentation du modèle conceptuel mais intègre également des connaissances sur la hiérarchisation des types de concepts et de relations ;
- Une partie assertionnelle dédiée à la représentation des assertions du domaine de connaissance étudié.

## 6.3 LogiquesdeDescription

Les Logiques de Description (DL) combinent des représentations intentionnelles et extensionnelles des connaissances [04] :

\_ Une TBox (T pour terminologique) contient les déclarations des primitives conceptuelles organisées

En concepts et rôles (relations binaires entre concepts dotées d'un domaine et d'une portée similaires aux attributs des frames). Ces déclarations décrivent les propriétés des concepts et des rôles et constituent donc une définition intentionnelle des connaissances ;

\_ Une ABox (A pour assertionnel) contient les déclarations d'individus, instances des concepts définis dans la TBox.

La spécification de propriétés des concepts et rôles utilise des constructeurs ensemblistes et logiques. Plusieurs modèles de DL peuvent être utilisés en fonction des constructeurs nécessaires à la modélisation. Le fait d'utiliser tous les constructeurs ou seulement une partie d'entre eux joue sur la décidabilité et la complexité des calculs de raisonnement. Loom [14] et CLASSIC [15] sont des exemples de modèles opérationnels implémentant ce modèle. Il est de plus utilisé dans les langages de représentation de connaissance OIL et OWL développés pour le Web.

## 7. Langages de spécification d'ontologies

Plusieurs langages de spécification d'ontologies ont été développés pendant ces dernières années. Parmi ces langages, nous citons :

### 7.1 KIF

KIF (Knowledge Interchange Format) [01] est un langage basé sur les prédicats du premier ordre avec des extensions pour représenter des définitions et des méta-connaissances. Tant que la logique du premier ordre est un langage de bas niveau pour l'expression d'ontologies, l'outil Ontolingua [140] permet aux utilisateurs de construire des ontologies KIF à un niveau plus élevé de la description par l'importation des définitions des ontologies prédéfinies.

### 7.2 KL-ONE

KL-ONE [10] est un langage basé sur la logique de description [09]. Il est une formalisation de représentation de la connaissance à base de cadres (« frame-based »). Ce système maintient la définition des concepts par un simple nommage, et l'indication de la correspondance des concepts dans une hiérarchie de généralisation/spécialisation. De nouveaux termes peuvent être définis par des opérations de conjonction des concepts. Par exemple l'opérateur « and » peut être utilisé pour préciser qu'un nouveau concept est une spécialisation commune de plusieurs autres concepts. De nouveaux rôles peuvent être introduits pour représenter les relations qui peuvent exister entre des individus dans le domaine modélisé. Les définitions des concepts peuvent inclure des restrictions sur les valeurs possibles, sur les nombres de valeurs, ou sur le type de valeurs qu'un rôle peut avoir pour un concept.

### 7.3 RDF(S) Resource Description Framework and RDF schema

RDF [02] « Resource Description Framework » est un formalisme graphique pour représenter des méta-données. Il est basé sur la notion de triplet (sujet, prédicat, objet). Le sujet et l'objet sont des ressources liées par le prédicat. RDF utilise la syntaxe XML, mais il ne donne aucune signification spécifique pour le vocabulaire comme sous classe de, ou le type. Les primitives de modélisation offertes par RDF sont très basiques. RDF Schéma [02] est un langage qui étend RDF avec un vocabulaire de termes et les relations entre ces termes, par exemple Class, Property, type, subclassOf, subPropertyOf, range et domain. RDFS est reconnu comme un langage d'ontologie qui définit :

- Des classes et des propriétés.
- Les sous-classes, les super-classes, les sous-propriétés, et les super-propriétés.
- Le domaine de définition et le domaine image des propriétés.

### 7.4 OIL (Ontology Interchange Language)

Dans l'optique d'une utilisation d'ontologies sur le Web, le langage RDF(S) a été enrichi par l'apport du langage OIL (Ontology Interchange Language) qui permet d'exprimer une sémantique à travers le modèle des frames tout en utilisant la syntaxe de RDF(S). OIL offre de nouvelles primitives permettant de définir des classes à l'aide de

mécanismes ensemblistes issus des logiques de description (intersection de classes, union de classes, complémentaire d'une classe). Il permet également d'affiner les propriétés de RDF(S) en en contraignant la cardinalité ou en en restreignant la portée [01].

Le langage OIL a été fusionné avec le langage DAML pour former le DAML+OIL. DAML (Darpa Agent Markup Language5) est conçu pour permettre l'expression d'ontologies dans une extension du langage RDF. Il offre les primitives usuelles d'une représentation à base de frames et utilise la syntaxe RDF [03]. L'intégration de OIL rend possible les inférences compatibles avec les logiques de description, essentiellement les calculs des liens de subsomption.

### 7.5 DAML+OIL (DARPA Agent Markup Language +OIL)

Beaucoup de travaux ont été faits dans le domaine de la représentation des connaissances parmi lesquels on peut citer les plus importants : SHOE, OntoBroker [104], OIL [105], et encore DAML + OIL [103] qui a remplacé DAML – ONT6. DAML + OIL est un langage construit sur des normes précédentes du W3C telles que RDF et RDF Schéma, et étend ces langages avec des primitives de modélisation plus riches. DAML+OIL a été conçu à partir du langage d'ontologie DAML-ONT (DARPA Agent ModellingLanguage-Ontology, Octobre 2000) en vue de combiner plusieurs composants du langage OIL [105]. OIL7 « OntologyInferenceLanguage » est une représentation basée sur le Web, et une couche d'inférence pour des ontologies. Il combine les primitives de modélisation des langages à base de cadres (frames) avec la sémantique formelle et le raisonnement fournis par la logique de description.

### 7.6 OWL(WebOntologyLanguage)

Développé par le groupe de travail sur le Web Sémantique du W3C, OWL peut être utilisé pour représenter explicitement les sens des termes des vocabulaires et les relations entre ces termes. OWL vise également à rendre les ressources sur le Web aisément accessibles aux processus automatisés [10], d'une part en les structurant d'une façon compréhensible et standardisée, et d'autre part en leur ajoutant des méta-informations. Pour cela, OWL a des moyens plus puissants pour exprimer la signification et la sémantique que XML, RDF, et RDF-S [12]. De plus, OWL tient compte de l'aspect diffus des sources de connaissances et permet à l'information d'être recueillie à partir de sources distribuées, notamment en permettant la mise en relation des ontologies et l'importation des informations provenant explicitement d'autres ontologies [06].

#### (i) Pourquoi OWL?

XML [16] fournit une syntaxe pour des documents structurés, mais n'impose aucune contrainte sémantique à la signification des documents. RDF est un modèle de données pour représenter les objets (Ressources) et les relations entre eux, fournissant une sémantique simple pour ce modèle qui peut être représenté dans une syntaxe XML. RDF Schema [02] est un langage de définition de vocabulaires pour la description de propriétés et de classes représentées par des ressources RDF. RDFS permet de définir des graphes de triplets RDF, avec une sémantique de généralisation / hiérarchisation de ces propriétés et de ces classes. OWL ajoute des vocabulaires pour la description des propriétés et des classes, des relations entre classes (exemple disjointness), des cardinalités, des caractéristiques de propriétés (symmetry), et des classes énumérées.

OWL est développé comme une extension du vocabulaire de RDF et il est dérivé du langage d'ontologies DAML + OIL [03].

(ii) **Souslangages de OWL**

OWL a trois sous langages de plus en plus expressifs : OWL Lite, OWL DL, et OWL Full :

- **OWL Lite** : supporte les utilisateurs ayant besoin principalement d'une hiérarchie de classification et des contraintes simples (un ensemble est limité à 0 ou 1 élément, par exemple). Il a une complexité formelle inférieure à celle d'OWL DL. OWL Lite supporte seulement un sous-ensemble de constructions du langage OWL.

- **OWL DL** : D'après son nom OWL DL utilise la logique de description DL [16 abdulghafour]. Il supporte les utilisateurs qui réclament l'expressivité maximale tout en retenant la complétude informatique (toutes les conclusions sont garanties d'être calculables), et la possibilité de décision (les calculs finiront en un temps fini). Il inclut toutes les constructions du langage OWL, qui ne peuvent être utilisées que sous certaines restrictions.

- **OWL Full** : a été défini pour les utilisateurs qui veulent une expressivité maximale et une liberté syntaxique de RDF sans des garanties informatiques. OWL Full permet à une ontologie d'augmenter la signification du vocabulaire prédéfini (RDF ou OWL). Il est peu probable que n'importe quel logiciel de raisonnement soit capable de supporter le raisonnement complet de chaque caractéristique d'OWL Full. Autrement dit, en utilisant OWL Full en comparaison avec OWL DL, le support de raisonnement est moins prévisible puisque l'implémentation complète d'OWL Full n'existe pas actuellement.

OWL Full et OWL DL maintiennent le même ensemble de constructions d'OWL. La différence se situe dans les restrictions sur l'utilisation de certaines de ses caractéristiques et sur l'utilisation des caractéristiques de RDF. OWL permet le mélange libre d'OWL avec RDF Schéma et, comme RDFS, n'impose pas une séparation stricte des classes, des propriétés, des individus, et des valeurs de données. OWL Full peut être vu comme étant une extension de RDF, tandis que OWL Lite et OWL DL peuvent être vus comme des extensions d'une vue restreinte de RDF. Alors les utilisateurs de RDF devraient se rendre compte qu'OWL Lite n'est pas simplement une extension de RDFS. OWL Lite met des contraintes sur l'utilisation du vocabulaire de RDF (par exemple, disjointness des classes, des propriétés, etc.). OWL Full est conçu pour la compatibilité maximale de RDF. Quant à opter OWL DL et OWL Lite, il faut considérer si les avantages du OWL DL/Lite (par exemple, support de raisonnement) l'emportent aux restrictions de DL/Lite à l'utilisation des constructions de OWL et de RDF.

## 7.7 Modèle orienté objet

La vaste communauté d'utilisateurs et le support commercial pour les normes orientées objet justifient l'investigation des techniques de modélisation objet pour le développement des ontologies [106]. Par conséquent, on peut exploiter la grande croissance des technologies orientées objet en adoptant UML + OCL comme un formalisme alternatif pour la représentation des ontologies.

### a) UML

UML est un langage et une notation graphique associée pour l'analyse et la modélisation orientée objet. Le paradigme de modélisation orientée objet est devenu la technique principale dans le développement de logiciel de l'industrie basé sur la vue largement acceptée que la modélisation orientée objet s'adapte bien aux modèles intuitifs du

monde [07].

#### b) OCL

OCL (Object ConstraintLanguage) est un langage qui permet d'écrire des expressions et des contraintes sur des modèles orientés objet. Une expression est une indication ou une spécification d'une valeur. Une contrainte est une restriction sur une ou plusieurs valeurs sur un modèle orienté – objet ou une partie de ce modèle.

Divers langages de contraintes ont été utilisés dans des modèles orientés objets, par exemple Syntropy [10], Catalysis [18] et BON [11] et des langages de programmation Eiffel [13]. OCL est le langage standard qui fait partie d'UML mis par l'OMG comme un standard pour l'analyse et le modèle orienté – objet [11]. Dans les diagrammes des classes, OCL peut être utilisé pour contraindre des valeurs d'attributs et des instances possibles de relations. Les expressions peuvent être utilisées dans un modèle UML afin de :

- spécifier la valeur initiale d'un attribut ou association « end ».
- spécifier la règle de dérivation pour un attribut ou une association « end ».
- spécifier le corps d'une opération.

Il existe 4 types de contraintes :

- Invariant : est une contrainte qui affirme qu'une condition est satisfaite par toutes les instances de la classe, type, ou interface.
- Précondition : une restriction qui doit être vraie au moment où l'opération va être exécutée.
- Postcondition : une restriction qui doit être vraie au moment où l'opération a fini son exécution.
- Guard : une contrainte qui doit être vraie avant qu'un état de transition soit déclenché.

## 8. Environnements et outils de modélisation des ontologies

Un ensemble d'environnements d'ingénierie ontologique ont été développés afin de systématiser l'ingénierie des ontologies. Les plus connus sont : ONTOLINGUA, ONTOSAURUS, ODE (Ontology Design Environment), PROTÉGé WIN (maintenant connu sous le nom de PROTÉGé 2000) et enfin, TADZEBAO et WEBONTO. Ces outils font l'objet d'une étude comparative des outils d'ingénierie ontologique proposée par l'équipe WonderTools de l'Université d'Amsterdam. D'autres outils recensés dans méritent d'être cités : HOZO, KAON (anciennement connu sous le nom d'OntoEdit) et OILED [13] :

- **KAON** (KarlsruheOntology and Semantic Web) est un environnement open source modulaire, basé dans Java, destiné à la conception, au développement et à la gestion d'ontologies. L'environnement intègre les modules suivants :
- API, Query, Serveurs (d'ontologie et d'application), Générateur de portails web (basés sur les ontologies), Éditeur d'ontologie (construction et maintenance).
- **HOZO** du MizLab de l'Université d'Osaka ; Hozo est un environnement composé d'un éditeur et d'un serveur d'ontologies. L'éditeur est développé en applets Java afin de pouvoir fonctionner comme un client via l'Internet. Hozo gère les ontologies et ses instances pour chaque programmeur. Chacun peut lire et copier

toutes les ontologies et les instances présentes dans Hozo, mais ne peut pas modifier celles développées par d'autres. La vérification de la consistance d'une instance se fait en utilisant les axiomes définis dans l'ontologie. Hozo gère l'exportation des ontologies et modèles en format XML, RDF, DAML+OIL.

- **ODE**du laboratoire d'Intelligence Artificielle de l'Université de Madrid. Les principaux avantages d'ODE sont le module de conceptualisation pour construire des ontologies et le module pour construire des modèles conceptuels ad hoc.
- **ONTOLINGUA**de l'Université Stanford ; Le serveur Ontolingua est le plus connu des environnements de construction d'ontologies en langage Ontolingua. Il consiste en un ensemble d'outils et de services qui supportent la construction en coopération d'ontologies, entre des groupes séparés géographiquement.
- **ONTOSAURUS** de l'Information Science Institute de l'Université de SouthernCalifornia. Ontosaurus consiste en un serveur utilisant LOOM comme langage de représentation des connaissances, et en un serveur de navigation créant dynamiquement des pages HTML qui affichent la hiérarchie de l'ontologie ; le serveur utilise des formulaires HTML pour permettre à l'utilisateur d'éditer l'ontologie. Destructeurs du LOOM en Ontolingua, KIF, KRSS et C++, ont été développés.
- **OILED**de l'Information Management Group de l'Université de Manchester développé par Sean Bechhofer et Gary Ng se veut un éditeur freeware d'ontologies, destiné à supporter le développement d'ontologies de petites et moyennes tailles, basées sur le standard DAML+OIL. OILED n'est pas un environnement de développement d'ontologies offrant des fonctionnalités supportant le cycle complet de conceptualisation et d'opérationnalisation.
- **PROTEGE-2000** du département d'Informatique Médicale de l'Université Stanford ; Protégé-2000 successeur de PROTEGEWIN, est un outil, une plateforme et une librairie d'ontologies, qui permettent : 1) de construire une ontologie du domaine, 2) de personnaliser des formulaires d'acquisition de connaissances et 3) de transférer la connaissance du domaine.
- **WEBONTO** du Knowledge Media Institute de l'Open University ; WebOnto et TADZEBAO sont des outils complémentaires. Tadzebao permet aux ingénieurs des connaissances de tenir des discussions sur les ontologies, en mode synchrone et asynchrone. WebOnto supporte la navigation collaborative, la création et l'édition d'ontologies sur le Web.

## 9. Systèmes de raisonnement

- Il est également possible de raisonner sur les ontologies en utilisant un moteur d'inférence général tel que FACT [18], ou des outils d'inférence spécifiques au Web sémantique basés sur des DL tels que RACER [17]. Ces deux outils peuvent être facilement intégrés à Protégé. Les logiques de description permettent de définir les bases logiques des différents formalismes de représentation de la

connaissance tant sur le plan de la représentation que sur le raisonnement.

- Dans notre travail, notre choix est porté sur le système RACER (RenamedABox and Concept Expression Reasoner). Ce dernier se présente sous la forme d'un serveur qui peut être accédé par le protocole TCP ou http. C'est un système de représentation de connaissances. Il implémente des tableaux de calcul hautement optimisé pour un DL très expressive. Il interprète les documents OWL et offre des services de raisonnement aussi bien pour le niveau terminologique que pour le niveau assertionnel de l'ontologie. Il permet aussi de vérifier la consistance et la relation de subsomption entre les concepts et d'autres tests plus élaborés sur les instances et les rôles.

## 10. Intégration des ontologies

- Le mot intégration est utilisé avec plusieurs significations dans le domaine d'ontologie. [01] définit trois différentes significations :

a) **Intégration** : Construction d'une nouvelle ontologie réutilisant (en assemblant, étendant, spécialisant, ou adaptant) d'autres ontologies disponibles. Ces différentes ontologies font partie de la nouvelle ontologie.

b) **Fusion** : Construction d'une ontologie par la fusion de différentes ontologies dans une seule ontologie qui les unifie.

c) **Utilisation** : Construction d'une application utilisant une ou plusieurs ontologies pour spécifier ou implémenter un système à base de connaissance.

La distribution et l'hétérogénéité de l'information ont créé le besoin de l'intégration pour améliorer l'accès aux données et assurer une recherche pertinente. Différentes approches existent et sont évaluées dans [11] vis-à-vis de quatre caractéristiques principales :

Le rôle et l'architecture des ontologies affectent profondément le formalisme de représentation d'une ontologie. Les ontologies sont utilisées pour la description explicite de la sémantique de l'information source. Ces ontologies sont employées selon différentes méthodes :

a) **Approche d'une ontologie simple** : Cette approche utilise une ontologie globale fournissant un vocabulaire partagé pour la spécification de la sémantique. Toutes les sources d'informations sont reliées à une ontologie globale. Une approche importante pour ce type d'intégration des ontologies est SIMS [19]. Un modèle indépendant de chaque source d'information doit être décrit pour ce système en reliant les objets de chaque source au modèle du domaine global.

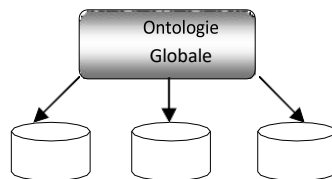


Figure I. 2 : Approche d'une ontologie simple.

b) **Ontologies multiples** : Chaque source d'information est décrite par sa propre ontologie. L'avantage est que chaque ontologie source peut être définie sans prendre en considération les autres sources ou les autres ontologies. Mais le manque d'un vocabulaire commun conduit à une difficulté extrême pour comparer différentes ontologies sources. Pour surmonter ce problème, un formalisme de représentation supplémentaire définissant le mapping inter-ontologie est fourni. Ce dernier identifie sémantiquement les termes correspondants de différentes ontologies sources. Ce mapping est difficile à définir à cause de divers problèmes d'hétérogénéité sémantique qui peuvent se produire. L'approche OBSERVER [14] est utilisée pour ce type.

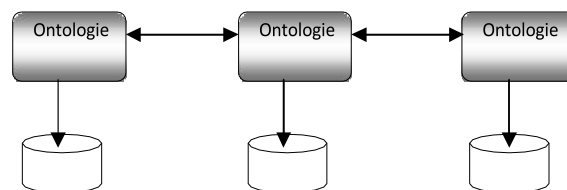


Figure I. 3 : Approche d'ontologie multiple

c) **Approche hybride** : Comme l'approche multiple, la sémantique de chaque source est décrite par sa propre ontologie. Mais pour rendre les sources d'ontologies comparables,

elles sont construites sur un vocabulaire partagé global [15] [16]. Ce vocabulaire partagé comprend les termes de bases ou primitives d'un domaine. Afin de construire des termes complexes, ces primitives sont combinées par des opérateurs, et les termes seront comparés plus facilement qu'une approche multiple. Parfois, ce vocabulaire partagé est représenté sous forme d'ontologie [10]. Dans COIN [14], la description locale d'une information, qu'on appelle contexte, est simplement un vecteur de valeurs d'attributs. Dans METACOTA [15], chaque source d'information est annotée par un label qui indique la sémantique de l'information. Ce label combine les termes primitifs du vocabulaire partagé. Dans BUSTER [07], le vocabulaire partagé est une ontologie générale. Une source d'ontologie est un raffinement de l'ontologie générale. L'avantage de cette approche est que des nouvelles sources peuvent être ajoutées sans le besoin de la modification du vocabulaire partagé. Elle maintient encore l'acquisition et l'évolution des ontologies.

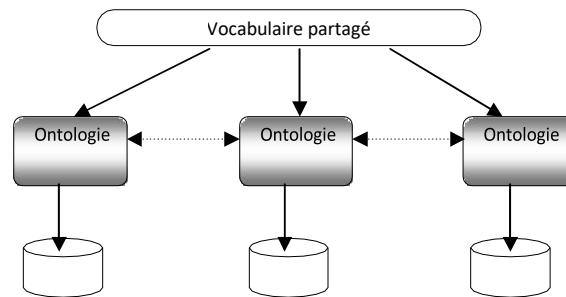


Figure I. 4 : Approche d'ontologie hybride.

## 10. 1 Alignement des ontologies

L'alignement d'ontologies consiste à trouver des correspondances entre les connaissances spécifiées dans les deux ontologies, de manière à pouvoir les exploiter conjointement dans le même système. En pratique, il s'agit d'identifier des concepts (ou des relations) de la première ontologie avec des concepts (ou des relations) de la seconde, ou de trouver des liens conceptuels (subsumption,...) entre eux.

## 10. 2 Fusion des ontologies

Contrairement à l'alignement où les deux ontologies de départ restent intactes, la fusion d'ontologies consiste, à partir de deux ontologies, à en créer une troisième qui

intègre les connaissances spécifiées dans les deux premières.

### 10.3 Mapping des ontologies

La relation d'une ontologie avec son environnement joue un rôle essentiel dans l'intégration de l'information. Mappings réfère à la mise en correspondance d'éléments d'ontologies entre eux ou avec des éléments d'autres types de ressources sémantiques.

Les deux plus importantes sortes de mapping utilisées pour l'intégration de l'information sont le mapping entre les ontologies et l'information qu'elles décrivent, et le « mapping » entre les différentes ontologies utilisées dans un système.

a) **Connexion aux sources d'informations** : différentes approches générales sont utilisées pour établir une connexion entre ontologies et sources d'informations :

- Ressemblance de la structure : une approche directe est de produire une simple copie de structure de la source d'information et l'encoder dans un langage qui rend possible le raisonnement automatique. Cette approche est implémentée par le médiateur SIMS [13] et le système TSIMMIS [12].

- Définition des termes : afin de clarifier la sémantique des termes dans un schéma de base de données, des approches comme BUSTER [10] utilisent les ontologies pour définir des termes supplémentaires de la base de données ou de son schéma.

- Enrichissement de la structure : combine les deux approches précédentes. Un modèle logique est construit qui ressemble à la structure de l'information source et contient des définitions additionnelles des concepts. Les systèmes qui utilisent l'enrichissement de la structure pour l'intégration de l'information sont OBSERVÉS [14], Kraft [01], PICSEL [18] et DWQ [19].

- Méta-annotation : une nouvelle approche est l'utilisation des méta-annotations pour ajouter la sémantique sur la source d'information. Cette approche est devenue préminente dans le besoin d'intégrer des informations présentes sur le WORLD WIDE WEB où l'annotation est un moyen naturel pour l'ajout de la sémantique. Des approches développées pour être utilisées sur le WWW sont Ontobroker [13], etSHOE [04].

b) **Mapping inter-ontologies** : le problème de mapping entre différentes ontologies est bien reconnu dans l'ingénierie de la connaissance. Des approches générales sont utilisées dans les systèmes d'intégration de l'information :

- Les mappings définis : une approche commune est de fournir la possibilité de définir des mappings. Elle est utilisée dans Kraft [10] où les translations entre différentes ontologies sont réalisées par des agents médiateurs spéciaux qui peuvent être adaptés pour établir la traduction entre différentes ontologies et même entre différents langages. Cette approche permet une grande flexibilité, mais elle ne peut pas assurer la préservation de la sémantique : l'utilisateur est libre de définir des mappings arbitraires même s'ils n'ont pas de significations ou s'ils produisent des conflits.

- Relations lexiques : Ces approches étendent le modèle de la logique de description par des relations inter-ontologies empruntées à la linguistique, par exemple « synonyme », « disjoint », « hyponyme », et « hyperonyme » qui sont utilisés dans le système OBSERVER [11].
- Connaissance de base haut niveau : afin d'éviter la perte de la sémantique, on doit rester dans le langage de la représentation formelle en définissant des « mappings » entre différentes ontologies. Une méthode directe pour rester dans le formalisme est de relier toutes les ontologies utilisées à une ontologie simple du haut – niveau. Tant que cette approche permet d'établir des connections entre des concepts de différentes ontologies en termes des superclasses communes, elle ne permet pas la correspondance directe.
- Correspondance sémantique : pour surmonter l'ambiguïté causée par le « mapping » indirect (comme l'approche précédente), une solution est d'identifier des correspondances sémantiques entre les concepts des différentes ontologies. Ces approches comptent sur un vocabulaire commun pour la définition des concepts des différentes ontologies.

En ce qui concerne notre approche de mapping d'ontologie, nous avons adopté le concept de sémantique bridge. Ce dernier est approprié pour représenter les mapping entre les ontologies d'application développées. Sémantique bridge considère différentes dimensions. Chacune d'elles décrit un aspect particulier [03] :

- **Entité** : cette dimension reflète le type d'entités d'ontologie à lier (par exemple les classes, les propriétés et les relations).
- **Cardinalité** : cette dimension reflète le nombre d'entités d'ontologie à lier (1:1, 1 : n, n : m).
- **Structurel** : cette dimension reflète la manière dont les liens élémentaires peuvent être combinés dans des mapping plus complexes (spécialisée, abstrait, composé ou alternative).
- **Contrainte** : cette dimension reflète les contraintes appliquées pendant la phase d'exécution aux instances de l'ontologie source.
- **Transformation** : cette dimension reflète comment les instances de l'ontologie source sont transformées pendant le processus de mapping.

Ces différentes approches traitant de l'intégration des ontologies pour un domaine ont permis la distinction des problèmes d'hétérogénéité entre les différentes ontologies.

## **11. Hétérogénéité**

Cependant, beaucoup de projets opérationnels et de recherche ont abouti à la création

des ontologies, parfois pour un même domaine. Ces ontologies ne sont homogènes ni dans leur structure, ni dans leur sémantique, et l'hétérogénéité est aperçue lors de la mise en commun de la connaissance représentée avec ces ontologies. Un certain nombre de recherches tente à étudier les causes du problème d'inconsistance ou de discordance entre différentes ontologies. Ceci a amené à la mise en oeuvre de multiples approches qui visent à traiter ce type de problème. Ces approches mettent en considération les différents cas qui peuvent y exister. Wiederhold [16] a distingué dans son approche les cas où la consistance, pour un domaine, n'est pas maintenue. Il a référé les différences entre des ontologies aux différences de la nomination et de la portée en terme des noms et de la sémantique des méta-informations sur les attributs qui apparaissent dans le schéma, et des noms et la sémantique qui apparaissent comme des valeurs dans le contenu de la source d'information. Quatre différences y sont déterminées :

- 1- Deux items d'attributs dans deux ontologies ont deux noms différents est le cas le plus commun et le plus facile à résoudre. Une simple table peut être utilisée pour maintenir la correspondance désirée, et rassembler l'information.
- 2- La différence de portée des concepts manipulés est plus insidieuse, et doit être déterminée par une analyse du contenu. La résolution exige l'établissement, la validation, et le traitement des règles.
- 3- Des différences relatives aux unités de mesure prises par des nombres sont communes. Une conversion peut être établie par une formule (un mètre = pieds/0.305). Dans ce cas des règles ou des tables doivent être introduites.
- 4- Des portées sur des attributs sont subjectives. Le terme «table» a une signification différente dans le domaine des meubles dans le domaine des bases de données. Les différences dans la portée mènent aux différences dans la mise en référence qui fait leur résolution pourtant plus critique.

L'approche Mitra [157] traite l'hétérogénéité sémantique entre multiples sources. Afin d'établir une correspondance entre les termes des ontologies, cette approche explore les discordances et leurs solutions en utilisant des graphes objets. Quatre types de discordance sont déterminés :

- 1- Discordance de la sémantique des termes : si un terme de deux ontologies différentes réfère à des concepts différents.
- 2- Discordance de structure : un même terme dans une source correspond à plusieurs termes dans une autre source, et cause qu'un noeud dans un graphe s'approprie à plusieurs noeuds dans un autre. Cette approche permet ce type de correspondance.
- 3- Discordance des instances : une instance d'une classe dans une source n'est pas une

instance de la même classe d'une autre source. Les règles d'articulation doivent spécifier explicitement quelles correspondances on veut générer dans ces cas.

4- Discordance de granularité : si on a deux noeuds correspondants, et le grand-père d'un noeud correspond avec le père de l'autre. Autrement dit, dans le premier graphe un concept est organisé dans une hiérarchie plus élaborée que dans le deuxième graphe.

Le besoin de traduire une représentation symbolique d'une connaissance en différents formats était le motif de la création du système « Ontomorph » par Chalupsky [158]. Ce système permet de traiter les problèmes qui peuvent se produire lors d'une telle traduction. Le scénario est de traduire les différents modèles du même domaine. Ces modèles se diffèrent à travers de multiples dimensions. Selon Chalupsky, six types de discordances communes qui peuvent être trouvées sont identifiés :

1- Syntaxe du langage de représentation de la connaissance (KR) : deux ontologies peuvent avoir des syntaxes et des conventions syntaxiques différentes.

2- Expressivité du langage de KR : chaque langage supporte l'expressivité d'une manière différente des autres langages. Par exemple, la négation qui peut être représentée par un langage mais pas dans un autre. Dans ce cas, des choix difficiles doivent être pris sur la manière de traduire les idiomes représentatifs.

3- Conventions de modélisation : même si le langage de représentation de la connaissance pour une base de connaissance source et, une autre cible sont les mêmes, ça n'empêche pas d'avoir des différences à cause de la manière de modélisation d'un domaine particulier. Par exemple, la classe person ne peut avoir deux sous-classes « male » et « femelle », ou on peut ajouter une relation « sexe » pour déterminer le sexe de la personne.

4- Modèle de recouvrement et granularité : Des modèles diffèrent par leur recouvrement d'un domaine particulier, et la granularité avec laquelle des distinctions sont faites. C'est la raison la plus forte pour la fusion des ontologies. Par exemple, une ontologie modélise les voitures mais pas les camions, une autre représente les camions sous des catégories.

5- Paradigmes de représentation : différents paradigmes sont utilisés pour représenter des concepts comme le temps, l'action, le plan, la causalité.

6- Systèmes d'inférence : une raison pour que des modèles semblent différents est due à leur construction pour produire des inférences désirées avec un moteur d'inférence particulier.

Enfin, nous constatons la classification des types d'hétérogénéité entre les ontologies en deux classes : les discordances liées à la syntaxe, et celles liées à la sémantique. Ces dernières sont beaucoup plus complexes, et leur résolution nécessite de grandes recherches. Cependant, cette approche ne prend pas en considération l'étude du langage OWL.

## **12. Conclusion**

Dans ce chapitre, l'ontologie est présentée comme un outil essentiel pour représenter le monde réel par la conception des concepts de ce monde dans un modèle représentatif, décrit par les classes y existantes, et les propriétés qui définissent les relations établies entre ces concepts. Le standard OWL s'est montré capable de jouer un rôle du pivot, grâce à sa compatibilité avec d'autres langages DAML+OIL, RDF, et RDFS.

Les approches d'intégration des ontologies nous ont apporté des points forts sur l'étude structurelle et les mécanismes de « mapping ». Cette intégration cause l'apparition des problèmes d'hétérogénéité entre différentes ontologies dans un même domaine. L'étude de ces types de problèmes nous a rendus conscient des difficultés que nous pouvons rencontrer durant notre travail.

***Chapitre II :***

***Généralité sur le progiciel de gestion  
intégré (ERP)***

### **1. Introduction**

Au cours des années 90, de nombreux dirigeants expriment leur insatisfaction face aux solutions informatiques disponibles. Ils se voient régulièrement proposer des solutions de gestion trop souvent peu adaptées et / ou inadaptées à la réalité de leurs besoins. Leurs attentes, toujours valides, veulent essentiellement que les systèmes d'information permettent d'une part d'organiser l'intégration des flux d'information entre les unités dans un même processus transversal de gestion, et d'autre part de définir et diffuser un langage commun au sein de l'entreprise via une standardisation détaillée de l'information liée aux différentes fonctions. Ceci aurait amené plusieurs entreprises à implanter un système ERP au coeur de leur système global d'information [RONG 2000].

Les progiciels de gestion intégrés (PGI) désignés souvent par le terme anglais ERP acronyme pour "Enterprise Resource Planning" connaissent un véritable succès auprès des entreprises et leur mise en œuvre constitue une refonte du système d'information, mais surtout une remise à plat des procédures de gestion au sein de l'organisation. C'est ainsi que la quasi-totalité des grandes entreprises mondiales sont déjà équipées d'un ERP et de plus en plus de PME cherchent à construire un système informatique unifié qui s'appuie sur ce progiciel. [26].

Cependant, la technologie ERP n'est pas une solution magique, ses bénéfices sont le résultat direct d'une préparation et d'une implantation efficace accompagnée d'une utilisation appropriée. Ainsi, la réussite de l'adoption d'un ERP repose essentiellement sur une préparation réfléchie de la part de l'entreprise. [20].

### **2. Le système intégré de gestion**

#### **2.1. La gestion intégrée**

Dans une entreprise, les fonctions sont interconnectées et interagissent entre-elles pour essayer d'atteindre l'objectif final préalablement fixé. C'est ainsi que le profit dégagé, qu'il soit de court terme ou de long terme, n'est que le reflet d'un équilibre général au sein de l'entreprise grâce à une harmonisation des tâches des différentes fonctions. L'entreprise se présente ainsi comme un système complexe où chacune de ses composantes alimente les autres par des données nécessaires au fonctionnement de l'ensemble du système.

D'ailleurs, selon [PERO, 2004], chaque sous système de l'organisation vise à travers le développement des compétences spécifiques à répondre le mieux aux demandes de

L'environnement (fournisseurs, clients,...). En outre, il est appelé à s'intégrer avec les autres services afin d'atteindre les objectifs de l'entreprise. En fait, cette intégration requise va favoriser les échanges des informations entre les services et donc va rendre plus cohérente l'entreprise.

Par ailleurs, l'instauration de cette gestion intégrée ne peut être conçue sans la mise en place d'un système d'information dont la mission est de faciliter les échanges des informations entre les différents acteurs de l'entreprise. En effet, le système d'information est défini selon comme étant « un système qui utilise des technologies de l'information pour saisir, transmettre, stocker, retrouver, manipuler ou afficher de l'information dans un ou plusieurs processus de gestion ».

Il est à noter que pour assurer l'intégration des principaux processus de l'entreprise, ce système d'information doit être cohérent c'est à dire garantissant l'unicité de l'information et l'accès à celle-ci à partir de toutes les fonctions de l'entreprise. Il s'agit donc d'instaurer une base de données unique pour pouvoir assurer une gestion intégrée.

### **2.2. Le principe de l'intégration**

Lorsque l'entreprise dispose de plusieurs systèmes d'information constitués d'applications spécifiques séparées, ces diverses applications ne peuvent communiquer qu'à travers des interfaces. Cependant, les techniques d'interfaçage n'arrivent pas à résoudre complètement cette désintégration des systèmes d'information. En effet, même en présence de ces techniques, la qualité des informations transmises entre les différents domaines de l'entreprise reste très pauvre : ces informations sont parfois incohérentes et toujours disponibles avec retard [03].

D'ailleurs d'après [04], « l'intégration a pour objectif la coopération des applications au sein d'un système unique et pour caractéristique une base de données logique unique ». Il s'en suit que l'intégration implique l'existence d'une base de données unique partagée par toutes les applications. En effet, [07] c'est à travers l'unicité de la base de données qu'on peut éviter les risques d'incohérence et de redondance existant lors des multiples opérations de traitement de données et harmoniser par la suite les différents processus de l'entreprise.

L'intégration désigne donc à la fois l'homogénéisation et la mise en cohérence du système d'information de gestion [09] et présente deux caractéristiques opérationnelles majeures à savoir l'unicité du référentiel et l'homogénéité des processus.

En effet, [07] le principe de l'intégration consiste à n'enregistrer une information qu'une seule fois tout en prenant compte, dès le début, de tous les traitements possibles de cette information à chaque étape du processus de la gestion.

C'est pour cela qu'on doit au préalable concevoir un système complet permettant de fournir les informations nécessaires au fonctionnement de l'ensemble des grandes fonctions de l'entreprise.

### **3. L'ERP, Historique et définitions**

#### **3.1. Historique de l'ERP**

Le concept de progiciel de gestion intégré existe depuis longtemps. En effet, dès les débuts de la Gestion de Production Assistée par Ordinateur (GPAO), certains professionnels avaient introduit des séries d'applications intégrées comprenant, outre la gestion de production proprement dite, la gestion commerciale et la gestion comptable. On a alors parlé pendant quelque temps de triptyque GP-GCCF-GA (Gestion de Production, Gestion Commerciale Comptable et Financière, Gestion Administrative). Les Américains utilisent le sigle MRP (Manufacturing Resource Planning), Planification des Moyens de Fabrication. [24]

Entre 1972 et 1974, les créateurs de SAP à l'origine du concept d'ERP avaient imaginé « la conception d'un outil de gestion standard adaptable à toutes les entreprises » .[05]

Dans les années 80, le concept de MRP II remplace les MRP, c'est l'époque où les progiciels commencent à s'imposer face aux développements spécifiques. Et c'est en 1990 qu'apparaît l'appellation ERP (Enterprise Resource Planning) introduite par Gartner Group, que le CXP traduit en français par Progiciel de Gestion Intégré (PGI) [12]

#### **3.2. Définition de l'ERP**

Un ERP est un logiciel qui permet de gérer l'ensemble des processus d'une entreprise en intégrant l'ensemble des fonctions de cette dernière comme la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais aussi la vente, la distribution, l'approvisionnement, la commerce électronique » (Définition du grand dictionnaire terminologique de l'Office québécois de la langue française (OLF)) [10]

Quant à [REIX 2002], il définit l'ERP en tant qu'une « une application informatique paramétrable, modulaire et intégrée, qui vise à fédérer et à optimiser les processus de gestion de l'entreprise en proposant un référentiel unique et en s'appuyant sur des règles de gestion standard ». Cette définition souligne le caractère standard de ce progiciel.

En outre, selon [JONE 2006], le terme ERP renvoie à l'infrastructure logicielle qui assure non seulement la cohésion interne dans l'ensemble de l'entreprise mais qui soutient aussi les processus commerciaux externes auxquels participe l'entreprise.

Il est important de noter qu'il n'existe pas un consensus sur une définition unique d'un ERP. Par ailleurs, la définition proposée par Willis-Brown s'avère la plus complète, « L'ERP est un système intégré qui permet à l'entreprise de standardiser son système d'information pour

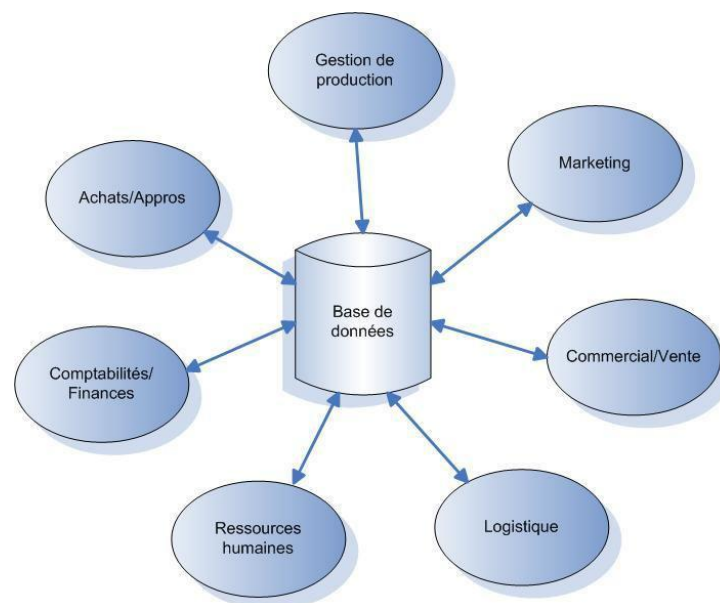
relier et automatiser ses processus de base. Il fournit aux employés les informations nécessaires pour diriger et contrôler les activités essentielles de l'entreprise le long de la chaîne logistique, de l'approvisionnement à la production/ exploitation jusqu'à la vente et à la livraison au client final. Les employés n'entrent qu'une seule fois les informations, qui sont alors mises à la disposition de tous les systèmes de l'entreprise. ». [10]

### 3.3. Organisation d'un ERP

Le principe fondateur d'un ERP est de construire des applications informatiques correspondant aux diverses fonctions citées précédemment de manière modulaire sachant que ces modules sont indépendants entre eux, tout en partageant une base de données unique et commune au sens logique. [20].

L'autre principe qui caractérise un ERP est l'usage de ce qu'on appelle un moteur de workflow et qui permet, lorsqu'une donnée est enregistrée dans le SI, de la propager dans les modules qui en ont l'utilité, selon une programmation prédéfinie. [10].

Ainsi, on peut parler d'ERP lorsqu'on est en présence d'un SI composé de plusieurs applications partageant une seule et même base de données, par le biais d'un système automatisé prédéfini et éventuellement paramétrable, un moteur de workflow. (Figure II.1)



**Figure II.1 : Organisation d'un ERP [20]**

Il doit permettre d'assurer le contrôle global de l'entreprise (contrôle de gestion, pilotage de production, suivi des processus). Ainsi il permet d'automatiser et d'intégrer les processus, de partager les données, de produire et d'accéder à l'information en temps réel [20].

### 3.4. Caractéristiques d'un ERP

[LEQU 1999] résume l'ensemble des caractéristiques d'un ERP comme suit :

1. Gestion effective de plusieurs domaines de l'entreprise par des modules intégrés ou des progiciels susceptibles d'assurer une intégration ainsi qu'une collaboration des processus ;
2. Existence d'un référentiel unique de données ; le référentiel est défini comme étant l'ensemble des références des données ainsi que les indications nécessaires pour retrouver les données elles-mêmes sur une base de données ;
3. Adaptation rapides aux règles de fonctionnement (professionnelles, légales ou résultat de l'organisation interne de l'entreprise) ;
4. Uniformisation des interfaces homme-machine (mêmes écrans, mêmes boutons, même famille de barres de menu, mêmes touches de fonctions et de raccourcis, etc) ;
5. Existence d'outils de développement ou de personnalisation de compléments applicatifs.

### **4. Les ERP : Pour qui ? Pourquoi ?**

#### **4.1. Les ERP : Pour qui ?**

Les ERP sont principalement destinés aux grandes entreprises ou multinationales du fait d'un coût important. Cependant, le marché des ERP tend à se démocratiser vers les PME/PMI. Certains éditeurs conçoivent un ERP uniquement pour ce type de structure. Enfin, il existe des ERP open source ce qui revient moins cher, puisqu'il n'y a pas de coût de licence (ils sont gratuits). En revanche, il faut inclure dans le calcul du coût d'acquisition total, les frais de maintenance et l'assistance technique. [20]

#### **4.2. Les ERP : Pourquoi ?**

Concrètement, les avantages de la mise en place d'un ERP sont les suivants [20] & [23] & [21] :

1. L'intégrité et l'unicité du SI : c'est à dire qu'un ERP permet une logique et une ergonomie unique à travers sa base de données, elle aussi unique au sens "logique". Ceci se traduit par le fait qu'il peut exister plusieurs bases de données "physiques" mais celles-ci respectent la même structure. En bref, un ERP permet d'éviter la redondance d'information entre différents SI de l'entreprise.
2. Communication interne et externe facilitée par le partage du même système d'information.
3. Meilleure coordination des services et donc meilleur suivi des processus (meilleur suivi de commande ou meilleure maîtrise des stocks par exemple).
4. Normalisation de la gestion des ressources humaines (pour les entreprises gérant de nombreuses entités parfois géographiquement dispersées).

5. L'utilisateur a la possibilité de récupérer des données de manière immédiate, ou encore de les enregistrer.
6. Les mises à jour dans la base de données sont effectuées en temps réel et propagées aux modules concernés.
7. Un ERP est un outil multilingue et multidevise, il est donc adapté au marché mondial, en particulier aux multinationales.
8. Pas d'interface entre les modules, il y a synchronisation des traitements et optimisation des processus de gestion.
9. La maintenance corrective est simplifiée car celle-ci est assurée directement par l'éditeur et non plus par le service informatique de l'entreprise. (Celui-ci garde néanmoins sous sa responsabilité la maintenance évolutive : amélioration des fonctionnalités, évolution des règles de gestion, etc.).
10. Maîtrise des coûts et des délais de mise en œuvre et de déploiement.
11. Mise à disposition, des cadres supérieurs, d'indicateurs nettement plus fiables que lorsqu'ils étaient extraits de plusieurs systèmes différents.

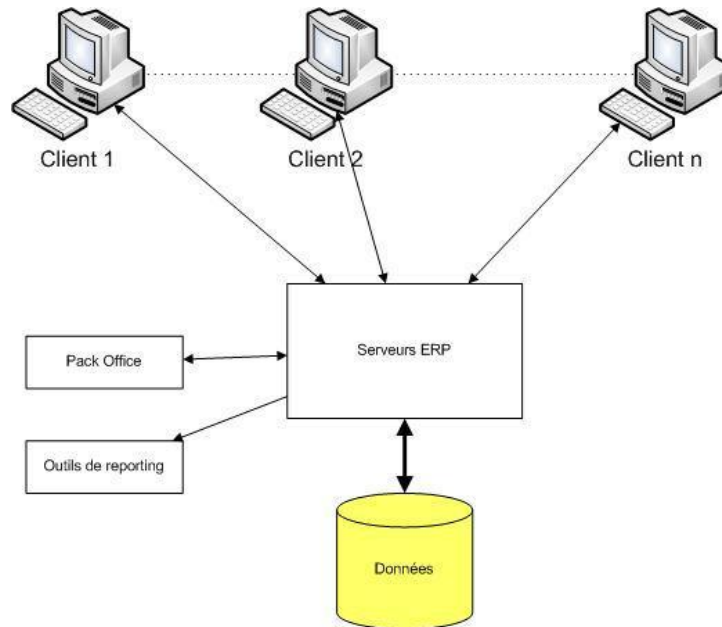
### **4.3. Intérêts de l'ERP dans l'entreprise**

Les différentes études, enquêtes et ouvrages qui ont été réalisés aux USA et en France sur la mise en œuvre des ERP ont permis de mettre en avant les axes les plus fréquents d'amélioration cités par les entreprises pour justifier de leur investissement dans un ERP. Ainsi nous citons [28] :

1. Au niveau de l'ensemble des fonctions de l'entreprise un pilotage par des indicateurs et une meilleure connaissance des structures de coûts.
2. Au niveau des ventes, un accroissement du nombre d'offre qui doit permettre de générer des revenus complémentaires.
3. Au niveau financier, l'optimisation des fonds propres avec, par exemple, une diminution des stocks et une amélioration des délais de paiement.
4. Au niveau des achats, une optimisation des conditions d'achat par une meilleure capacité à négocier, avec une diminution des temps d'approvisionnement.
5. Au niveau de la production, la maîtrise des coûts de production et des niveaux de stock pour accroître la productivité et la réactivité.
6. Au niveau de la communication entre les différents services de l'entreprise, une meilleure circulation des informations par l'existence d'une base unique.

### **5. Architecture technique d'un ERP**

Concernant le déploiement d'un ERP, la plupart du temps est basé sur une architecture client/serveur comme le décrit dans la figure II.2



**Figure II.2 : Architecture technique d'un ERP [20]**

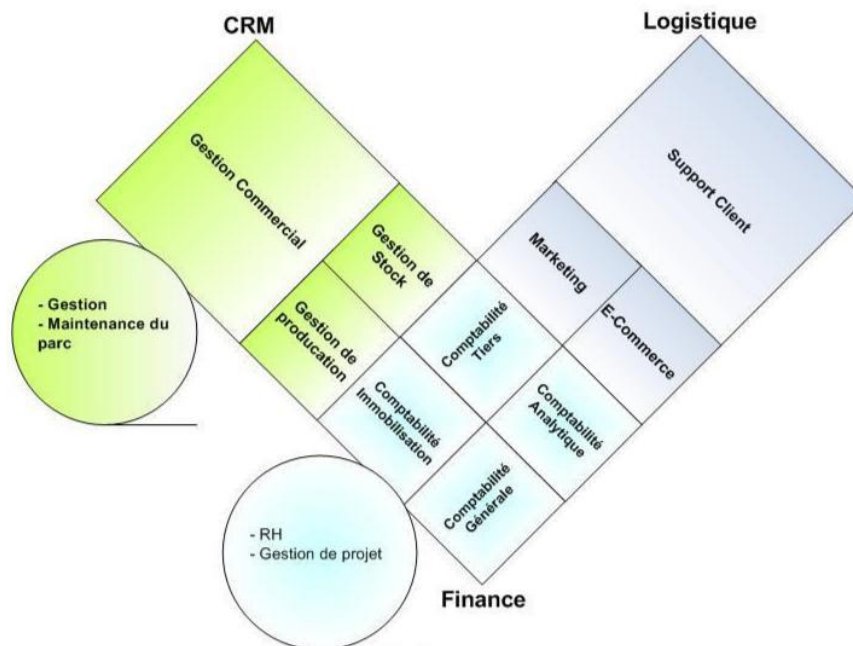
L'ERP est donc sur serveur. La majorité des ERP sont couplés à une base de données ORACLE. De plus, les ERP sont compatibles pack Office, en particulier pour PowerPoint et Excel. En effet, le premier étant utile pour personnaliser les bureaux ERP en fonction de l'entreprise et le second pour effectuer les imports/exports de données. Enfin, les ERP sont aussi compatibles avec des outils de reporting (Crystal Report en général). Le reporting étant utilisé en particulier pour le module de gestion relation client. [20]

### **6. Architecture modulaire d'un ERP [20]**

Un ERP est un ensemble dont toutes les parties fonctionnent les unes avec les autres d'où l'ergonomie et l'unicité des informations et donc la cohérence du SI.

Un ERP est modulaire dans le sens où il est possible de n'avoir qu'une ou plusieurs applications en même temps, ou peu à peu. Les applications modulaires telles que les ERP permettent d'être sûr de la compatibilité des modules entre eux, ils s'imbriquent comme des blocs de Lego et fonctionnent ensemble (pas de vérification de compatibilité à effectuer).

La figure II.3 c'est un exemple d'architecture modulaire qui tend à représenter tous les ERP :



**Figure II.3 : Architecture modulaire ERP**

L'architecture modulaire schématisée dans la figure ci-dessus intègre plusieurs modules retouchant aux grandes fonctions d'une entreprise que l'on peut détailler de la manière suivante : le module finance, logistique et e-commerce.

### 6.1. Le module finance

Le module finance se divise en cinq sous modules. Il rassemble les données pertinentes (en particulier les données comptables de l'entreprise) pour établir sur un plan international des comptes annuels. Ce module effectue du contrôle de gestion et des prévisions concernant les objectifs de l'entreprise.

De même, il permet de faire de la comptabilité tiers, analytique, générale, de gérer les immobilisations (gestion des investissements), les ressources humaines, soit l'administration du personnel, des frais de déplacement et du temps de travail ainsi que la gestion de paie et les besoins en postes.

Ce module est complexe car il faut avoir de bonnes connaissances en comptabilité.

## 6.2. Le module logistique

Le module logistique ou encore appelé Négocce, est le module le plus convoité par les entreprises car il permet de gérer tout ce qui se rapporte aux ventes/achats, en particulier la gestion des stocks qui coûtent chers aux entreprises.

Ce module gère les commandes clients et les livraisons. Il assure une liaison directe avec le compte de résultats et le système de production. Il permet l'optimisation des processus de workflow, la gestion précise des stocks, des contrôles qualités et factures. Suite au contrôle qualité, il y a coordination et déclenchement des mesures correctives. Enfin, il est possible de planifier/gérer/suivre la maintenance du matériel.

Les processus généraux à paramétrer dans le module Negoce sont décrits ci-dessous : (Figure II.4)

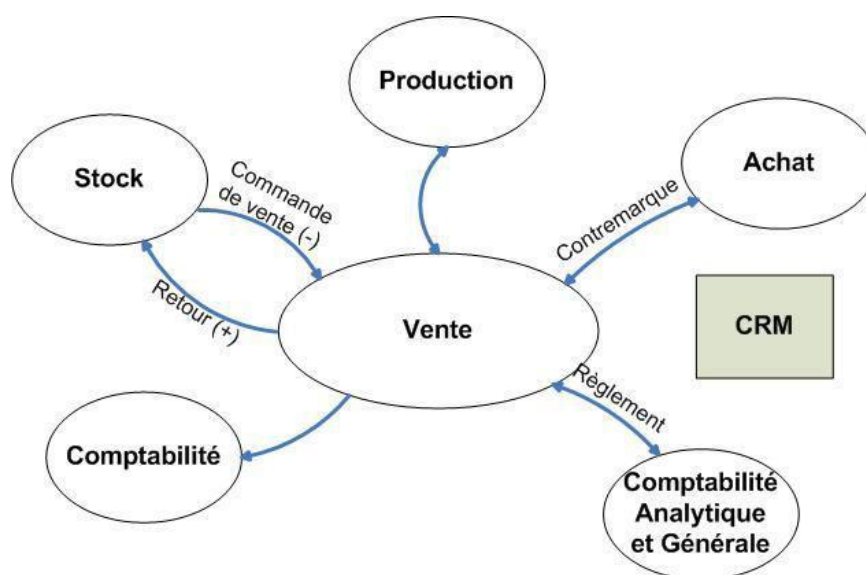


Figure II.4 : Processus de module logistique (Negoce)

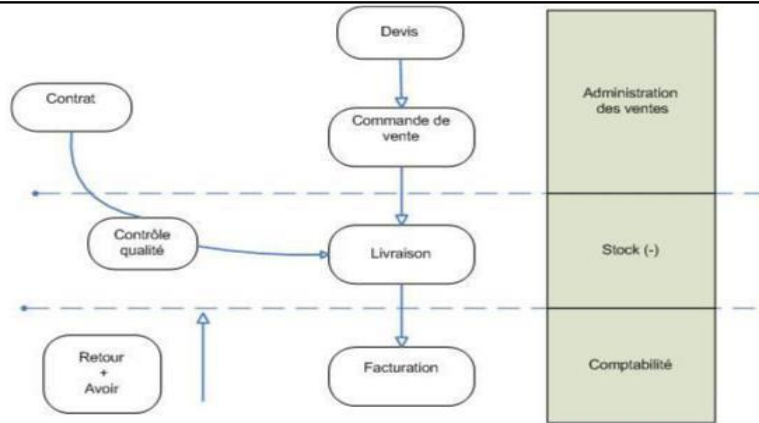
Une commande de vente correspond à une vente, donc il y a décrémentation du stock et inversement lors d'un retour. Entre le processus de vente et achat, il y a contreremarque, c'est-à-dire une fabrication d'un article après la vente. Enfin, entre le processus vente et comptabilité générale et analytique, il y a les règlements clients ou encore les avoirs.

Ceci correspond donc aux processus de bases à connaître afin de paramétrer un ERP.

Les tiers impliqués dans une vente sont : les clients, l'utilisateur (personnel/ représentant), le transporteur, les prospects et les facteurs/affacteurs (société qui gère la facturation).

### 6.2.1. Le processus vente, la commande de vente

Pour effectuer une commande de vente, c'est-à-dire une vente, il faut avoir en données de bases : les articles, les clients et les fournisseurs. Ci-dessous (Figure II.5) est décrit le processus de commande de vente à connaître afin d'effectuer, dans l'ordre, une commande de vente.



**Figure II.5 : Processus de commande de vente**

Une vente peut-être de plusieurs types :

- Commande issue d'un devis validé totalement ou partiellement.
- Commande de vente normale avec certains accès, par exemple, fiche client, conditions tarifaires, accès aux stocks, etc.
- Commande de vente de type marché (avec contrat).

### 6.2.2. *Le processus stock*

Lors d'une vente, d'un achat ou autres, il y a différents mouvements internes qui ont une influence sur le stock à savoir :

- Les entrées diverses
- Les sorties diverses
- Les changements d'emplacement (d'un article)
- Les inventaires
- Les contrôles qualité
- Les transferts inter sites
- Les changements de numéro de lot ou type de numérotation
- Les déclarations de production

Une description schématique qui modélise le processus stock au sein de l'ERP : (Figure II.6)

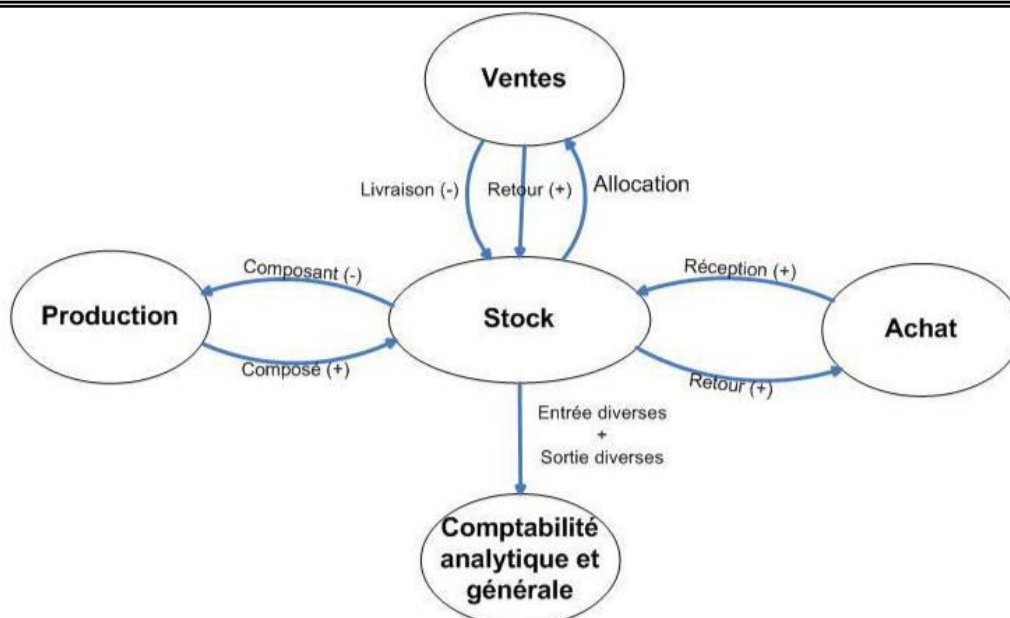


Figure II.6 : Processus stock

Dans un ERP, les stocks sont gérés de manières automatiques. Le parametreur doit, par ailleurs, définir en fonction de l'entreprise le type de gestion de stock ainsi que les quantités minimales et maximales. Il existe deux types de gestion de stocks :

### Réapprovisionnement sur seuil

Le réapprovisionnement sur seuil consiste à paramétrer un message automatique, appelé suggestion, qui va se déclencher lorsque le seuil, fixé par le paramétreur est atteint. Cette méthode est en fonction de la demande d'un article.

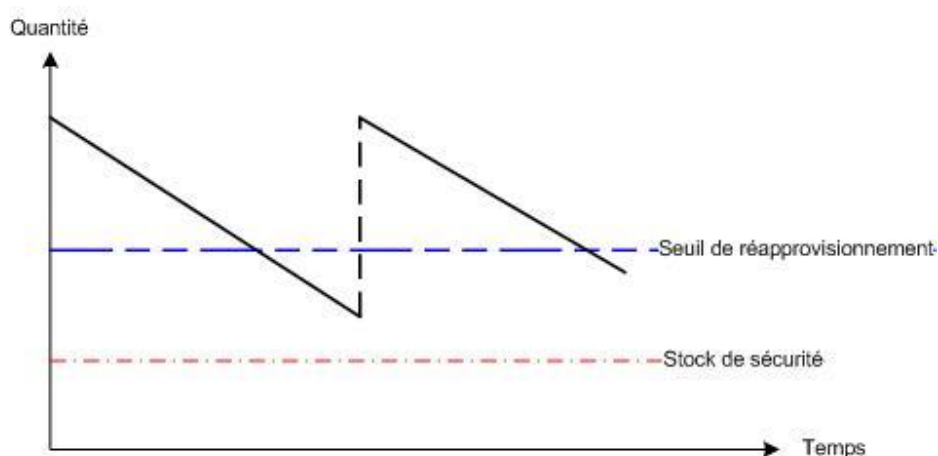


Figure II.7 : Réapprovisionnement sur seuil

### Réapprovisionnement périodique

Cette méthode de réapprovisionnement est périodique, c'est-à-dire qu'après un temps T, paramétré, une demande de réapprovisionnement suggérée est effectuée à travers une commande de vente. Cette méthode est utilisée lorsque la consommation d'article, c'est-à-dire les sorties de stock, sont fréquentes.

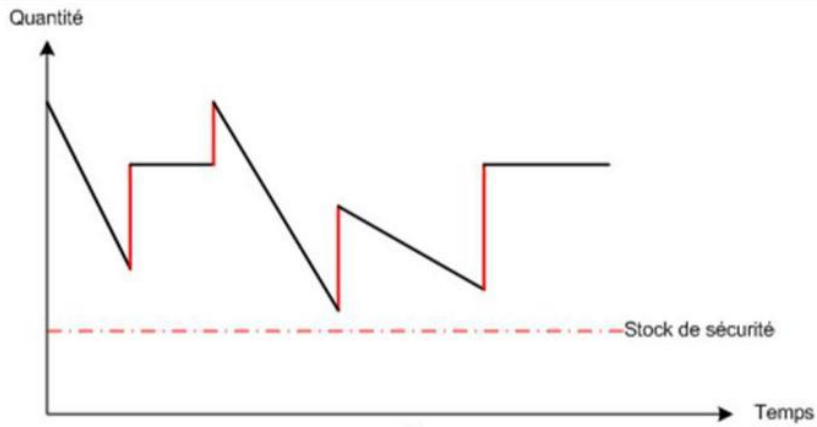


Figure II.8 : Réapprovisionnement périodique

### 6.2.3. Le processus achat, la commande d'achat

Le dernier processus à connaître afin de comprendre la logique d'un ERP et le paramétrage est le processus achat ou commande d'achat : (Figure II.9)

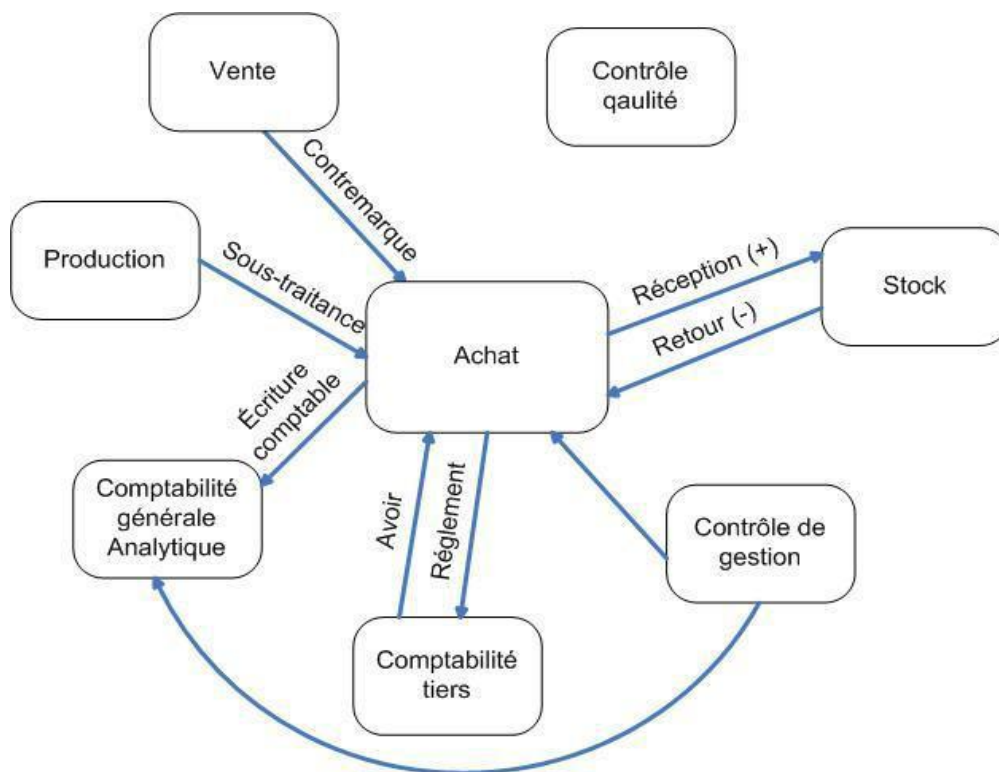
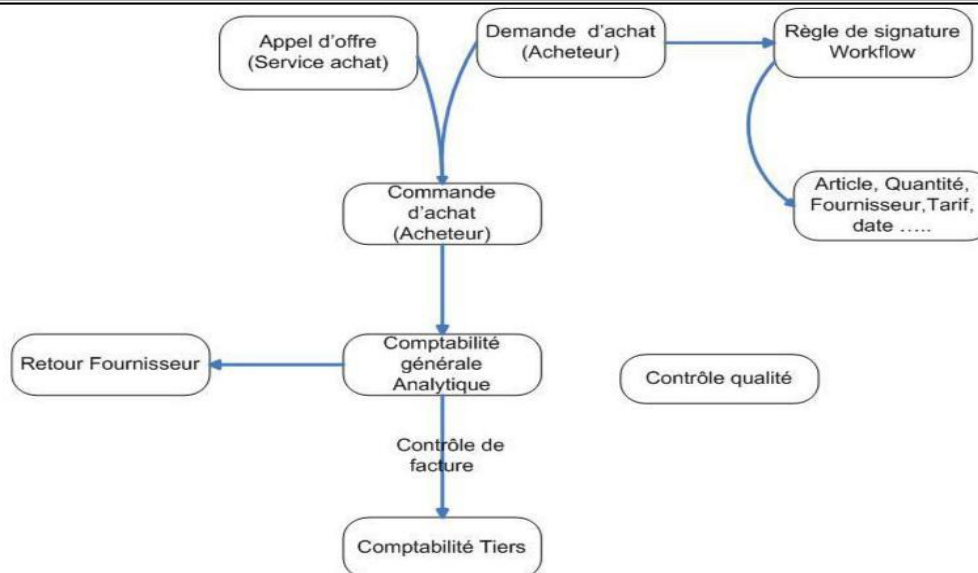


Figure II.9 : Processus achat

Enfin, le flux à suivre pour paramétrer un ERP concernant la commande d'achat est modélisé ci-dessous.



**Figure II.10 : Processus de commande d'Achat**

Pour effectuer une commande d'achat, il faut connaître les articles, les fournisseurs, la quantité à commander pour chaque article ainsi qu'un tarif de vente. Par conséquent, les données de base à manipuler sont donc les fournisseurs, les articles, et les tarifs d'achat car ces derniers peuvent varier en fonction de certains critères (période, quantité, etc.).

### 6.3. Module e-commerce

Ce module permet de faire du e-commerce. C'est un logiciel de Gestion Relation Client (GRC) plus communément appelé Customer Relationship Management (CRM). Il permet d'effectuer les statistiques voulues sur tous types de données (en particulier clients, vendeurs, production, fournisseurs,...). Ensuite ses résultats sont analysés en fonction des besoins de l'entreprise. En particulier pour ce module, un outil de reporting est associé. Enfin, ce module permet, suite aux résultats, d'effectuer de l'e-mailing, des offres marketing.

## 7. Principaux éditeurs d'ERP leurs parts de marché

Le marché de l'ERP représente une vraie manne pour les prestataires de services informatiques. Il est devenu aujourd'hui, le marché le plus porteur de l'informatique.

On distingue deux sortes d'ERP : les ERP propriétaires et les ERP OpenSource.

### 7.1. Les ERP propriétaires

Aujourd'hui, il existe de nombreux ERP propriétaires mais seuls quelques éditeurs internationaux restent vraiment dans la course.

Les trois premiers sont : SAP (leader mondial), Oracle/PeopleSoft et SAGE.

Le progiciel SAP a remporté rapidement un succès important auprès des grandes entreprises en proposant un progiciel multilingue et multidevises. SAP s'intéresse aussi au

marché des PME, en pleine croissance en proposant sa suite BusinessOne, pour les entreprises de 2 à 250 salariés. [16]

SAP est une application client-serveur. Ses modules couvrent l'ensemble des fonctions de gestion de l'entreprise et chaque module couvre des besoins complets de gestion. Certaines entreprises implémentent tous les modules fonctionnels de SAP, ou seulement quelques-uns. SAP R/3 est entièrement paramétrable. Par ailleurs, grâce à son environnement de développement, SAP R/3 peut être adapté à des besoins spécifiques. [16]

Voici la table qui présente les principaux ERP propriétaire et leurs parts de marché [17] :

Les vendeurs d'ERP dans le monde en 2005 et les prévisions 2006					
(En millions de dollars)					
Acteur	Chiffre d'Affaire 2015	Part de marché 2015	Chiffre d'Affaire 2016	Part de marché 2016	Evolution
SAP	10 668	42 %	12 470	43 %	+17 %
Oracle	5 080	20 %	6 670	23 %	+29 %
Sage Group	1 524	6 %	1 450	5 %	+10 %
Microsoft	1 016	4 %	1 160	4 %	+18 %
SSA Global	762	3 %	870	3 %	+3 %
Autres	6 350	25 %	6 380	22 %	+0 %
<b>Total</b>	<b>25 400</b>	<b>100 %</b>	<b>29 000</b>	<b>100 %</b>	<b>+14 %</b>

Tableau II.1: principaux ERP (Source: AMR Research, oct. 2016)

### 7.2. Les ERP Open Source [23]

L'implémentation d'un progiciel Open Source revient moins chère, puisqu'il n'y a pas de coût de licence. En revanche il faut inclure, dans le calcul du coût d'acquisition total, les frais de maintenance et de l'assistance technique.

Parmi les principaux progiciels Open Source nous trouvons : Aria, Compiere, ERP5, Fistera, OFBiz (Open for Business), PGI Suite, Value Enterprise, Tiny ERP.

Les ERP d'aujourd'hui tirent parti du Web. Les utilisateurs peuvent accéder à ces systèmes au moyen d'un navigateur. Ces progiciels sont de plus en plus orientés vers l'extérieur et sont capables de communiquer avec les clients, les fournisseurs et d'autres organisations.

## 8. Domaine d'application d'un ERP [CIRE 2006]

Toutefois, il est entrain de devenir une application sectorielle, c'est-à-dire que les éditeurs adaptent de plus en plus leurs ERP à un secteur d'activité spécifique.

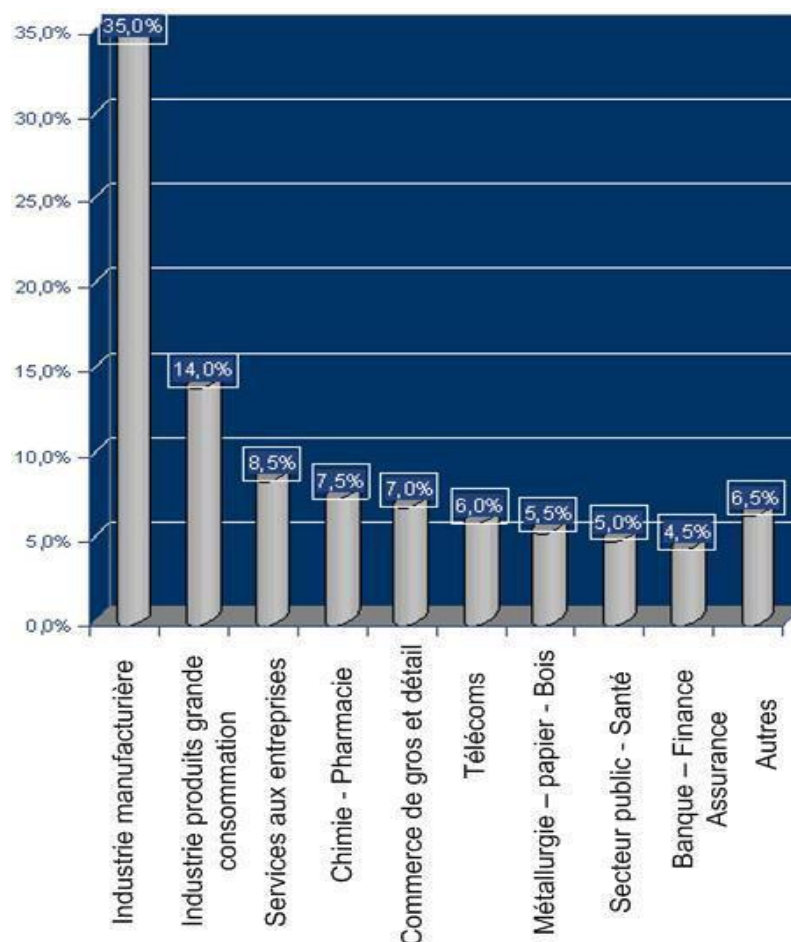


Figure II.11 : Répartition des licences ERP par secteur d'activité en 2005 [CIRE 2006]

En 2006, presque toutes les grandes entreprises disposent d'un ERP.

La nouvelle cible privilégiée est désormais les PME-PMI, l'administration publique et les secteurs de la santé

## 9. Les difficultés de l'implantation

[DAVE 1998] a mentionné plusieurs cas d'entreprises qui ont échoué à la mise en place d'un ERP. Parmi ces entreprises, on trouve Dell Computer et Dow Chemical.

Parmi les problèmes qu'on peut rencontrer lors de l'implantation de l'ERP on cite :

- Les problèmes relatifs à l'intégration : en essayant d'adapter l'ERP à la plate forme existante, les entreprises trouvent des difficultés à intégrer le progiciel aux systèmes existants, aux bases de données existantes et aux systèmes de communications spécifiques à chaque entreprise [20].

- La modification du progiciel : selon [20], la modification du progiciel est déconseillée. D'ailleurs, très souvent les éditeurs refusent de faire des changements à cause des coûts élevés engendrés par les développements spécifiques et leur maintenance.

- La résistance au changement exprimée par le personnel : comme l'implantation d'un ERP engendre des changements organisationnels, cette résistance représente une sorte de manifestation d'un refus du modèle organisationnel véhiculé par l'ERP. D'ailleurs, cette résistance au changement peut constituer l'élément clé de l'échec de la mise en place de l'ERP et elle est essentiellement due à ce que les utilisateurs sont souvent mal préparés à cette tâche [10].

- Le manque de compétences [22] : vu leur complexité, les projets ERP demandent des expertises importantes d'origine souvent externe à l'entreprise. D'ailleurs, les spécialistes en ERP sont essentiels afin de surpasser les problèmes relatifs à la conception et à l'implantation de l'ERP. Cependant, l'équipe projet doit comporter aussi bien des compétences internes qu'externes à l'entreprise. En effet, le fait de recourir fréquemment aux consultants externes à cause de l'indisponibilité du personnel interne engendre des coûts non négligeables.

### **10. Les étapes de l'implantation de l'ERP**

Vu les difficultés de l'implantation de l'ERP, la mise en place de ce progiciel devrait passer par différentes étapes. [20] ont développé un modèle d'implantation comportant 4 phases :

1. **Phase de conception** : Cette phase constitue une phase de planification où les principales décisions sont prises : la sélection du progiciel, l'identification du chef de projet, l'approbation du budget et du plan de projet.
2. **Phase d'implantation** : Cette phase comporte l'analyse de la situation actuelle de l'entreprise, la mise en place du nouveau système et sa configuration aux systèmes existants et les tests.
3. **Phase de stabilisation** : Cette phase vient juste après avoir rompu avec l'ancien système. C'est la phase pendant laquelle le système se stabilise pour permettre de détecter les différentes anomalies et de les corriger éventuellement.
4. **Phase d'amélioration continue** : L'amélioration continue consiste par exemple dans la maintenance du système, l'introduction de nouvelles fonctionnalités au système, le support des utilisateurs, les mises à jour éventuelles, etc.

### **11. Positionnement des ERP par rapport aux nouveaux concepts**

Les concepts de : Supply Chain Management (SCM), Customer Relationship Management (CRM), E-business, Enterprise Application Integration (EAI) concurrencent les ERP et risquent de les dépasser. Les ERP restent leaders en termes d'intégration [02].

### 11.1. L'ERP et le SCM

La comparaison entre ces deux outils peut être résumée dans le tableau qui suit :

	ERP	SCM
Fonctionnalité	Pouvoir tout couvrir et intégrer (gestion commerciale, industrielle, financière, etc)	Optimiser la gestion logistique, de distribution, de production, etc
Principes d'optimisation	Cherchent des solutions faisables et l'optimisation individuelle des fonctions	Recherche des simulations et des horizons variés pour identifier le meilleur scénario et visent à optimiser l'ensemble des processus pour la prise en compte simultanée de plusieurs contraintes.
Finalité	La gestion de l'entreprise par des fonctions opérationnelles et de planification	Aider à la décision, à la simulation et à l'optimisation de l'ensemble des composants de la supplychain

**Tableau 1 : Comparaison entre ERP et SCM (source : [25])**

### 11.2. L'ERP et le E-business

Le principal point fort du e-business est sa grande ouverture. Il permet l'accès immédiat et à faible coût à une grande quantité d'information. Du fait de cette richesse, les ERP doivent commencer à s'y adapter et à l'intégrer de façon à avoir des accès à travers le net, et pour se faire, doivent avoir des applications qui soient compatibles avec ce net. Le e-business constitue donc un virage important à prendre pour les ERP. [25]

### 11.3. L'ERP et le CRM

Le CRM est un véritable foisonnement d'outils parfois disparates portant sur les besoins et les offres pour mettre le client au cœur de l'entreprise. Il diffère en cela de l'ERP, dans la mesure où il met en œuvre des applications e-business qui aident à connaître et à anticiper les attentes des clients actuels et futurs. L'intégration joue un rôle fondamental et permet aux ERP de se maintenir face à des outils plus spécifiques. [25]

### 11.4. L'ERP et L'EAI

L'EAI est une structure informatique dont l'objectif est l'interfaçage de bases de données existantes. L'EAI serait le modèle rêvé de l'intégration et de la diversité. Il traiterait en effet toutes les applications en une seule et combinerait pour se faire les meilleures applications de chaque domaine : SCM, ressources humaines, CRM, GPAO, etc. [25]

## **12. Exemple d'un ERP : Tiny ERP**

### **12.1. Présentation de Tiny ERP**

Tiny ERP est un progiciel de gestion intégré libre (GPL) entièrement développé en Python. Il intègre toutes les étapes de la gestion d'entreprises et est utilisé dans des petites et moyennes entreprises (1 à 200 employés).

Tiny ERP est le progiciel de gestion le plus téléchargé au monde avec plus de 500 installations par jour. Il est disponible en 16 langues et possède un réseau de partenaires et contributeurs mondial. Plus de 200 développeurs participent au projet sur le système de développement collaboratif Tiny Forge.

### **12.2. Module de Tiny ERP**

Les différents modules sont :

- Comptabilité : financière et analytique, gestion des débiteurs et créancier, TVA, multi-devises, multi-sociétés, plans comptable virtuels, ...etc.
- Stock : par lot et/ou séries, gestion FIFO/LIFO, multi-entrepôts,...
- Ventes : différentes logistiques, réapprovisionnement,
- Production : méthode MRP 2, production, réapprovisionnement, Produits : avec variantes, listes de prix et contrats
- Projets : tâches, industriels ou services, allocation des ressources, Achats : contrôle qualité, analyse des fournisseurs,
- CRM : segmentations, SMS/Fax/Email automatisé, campagnes marketing, ...

Ainsi que des modules spécifiques à certains secteurs nichent :

- Gestion de parcs informatiques,
- Ventes à l'enchère,
- Huissiers de justices, ...

### **12.3. Avantage Tiny ERP**

Grâce à son extrême modularité, les développements collaboratifs ont pu être parfaitement intégrés, permettant à chaque entreprise de choisir parmi une grande liste de fonctions disponibles. Comme la plupart des logiciels libres, l'accessibilité, la flexibilité et la simplicité sont les maîtres mots du développement. En effet, on ne peut plus se permettre de former les utilisateurs pendant plusieurs mois sur le système puisque ceux-ci peuvent le télécharger et l'utiliser directement.

Nous trouverons ainsi des modules pour tous les types de besoins, permettant à chaque entreprise de construire son système sur mesure par le simple assemblage et paramétrage de modules.

Tiny ERP est le seul système de gestion qui est à la fois utilisé par de grands comptes mais également par des indépendants. Cette diversité est à l'image de la solution : un rapport élégant entre l'étendue fonctionnelle du logiciel et une très grande accessibilité.

Et cette diversité se retrouve également dans les secteurs et métiers qui utilisent le logiciel : agro-alimentaire, textiles, sociétés de services, production, pharmaceutique, libraires, huissier de justice, ventes publiques, informatique, négoce, etc.

### **13. Conclusion**

En conclusion, les ERP offrent de nouvelles opportunités à l'entreprise pour la gestion et la prise en charge de plusieurs périodes (pour les exercices comptables par exemple), plusieurs devises, plusieurs langues pour les utilisateurs et clients, plusieurs législations, plusieurs axes d'analyse en informatique décisionnelle.

Mais l'implantation comporte plusieurs risques : des risques organisationnels (le progiciel et l'organisation de l'entreprise doivent cohabiter), de mise en œuvre (au niveau formation utilisateur), fonctionnels (fonctions offertes par le progiciel par rapport aux fonctions attendues), techniques, contractuels entre l'éditeur et l'entreprise et enfin des risques économiques du fait de l'investissement. Il convient donc d'étudier minutieusement les conditions de réussite de la mise en œuvre d'un projet ERP. [06]

Un rappel d'une bonne marche à suivre pour la réussite de la mise en œuvre d'un projet ERP :

- Bien étudier le SI de l'entreprise et le cahier des charges.
- Effectuer une étude de faisabilité afin de voir ce qu'il y a à paramétrer et ce qui existe déjà par défaut.
- Ne pas hésiter à modéliser les processus et flux de gestion et surtout les suivre tout au long des étapes du paramétrage.
- Tester les flux paramétrés, avec l'entreprise, en particulier les différents utilisateurs.

Dans le chapitre suivant nous allons introduire les concepts liés aux systèmes Multi-Agents.

***Chapitre III :***  
***Conception et Implémentation***

## **I - Introduction**

Actuellement, il existe de nombreuses bases de données géographiques (BDG) qui couvrent la même région du monde réel à des données ou représentations géométriques différentes. La multiplicité et l'indépendance des bases de données géographiques posent aujourd'hui un certain nombre de problèmes. Le fait qu'il n'existe aucun lien entre les objets géographiques qu'il implique : une répétition des opérations de mises à jour, un manque de cohérence entre les différentes BD, et une impossibilité d'effectuer des analyses à plusieurs niveaux de détail.

L'explicitation des ontologies s'effectue au moyen de langages est une des principales décisions à prendre dans le procédé de développement d'ontologies consiste à choisir le langage dans lequel l'ontologie sera exprimée et utilisée. Comme l'éventail des choix possibles concernant les langages de représentation et de spécification d'ontologies étant très large, il convient de choisir celui qui respecte ces exigences et critères techniques : la lisibilité, la possibilité de faire des inférences c'est-à-dire permettre le traitement informatique des données en vue de calculer les déductions logiques possibles.

L'implémentation consiste à traduire l'ontologie formelle de l'étape de formalisation ou d'évolution, dans un langage opérationnel de définition d'ontologies. Notre choix porte sur le langage OWL pour la codification de l'ontologie, et cela pour toutes les fonctionnalités sémantiques que permet OWL et qui sont plus riches que celles de DAML+OIL. D'autre part, OWL est considéré comme un langage standard de définition des ontologies pour le Web. Par ailleurs, OWL fait partie du formalisme des logiques de descriptions. Sa sémantique peut être définie via une transcription vers la logique de descriptions SHIQ.

## **II- Odo**

### **II-1-Odo : Définition**

Odo, anciennement OpenERP et Tiny ERP, est initialement un progiciel open-source de gestion intégré comprenant de très nombreux modules permettant de simplifier la gestion d'entreprise dans son ensemble. Le logiciel est utilisé par plus de deux millions d'utilisateurs pour gérer leurs entreprises à travers le monde. Odo est le système ERP open-source le plus populaire. À l'origine un ERP, le logiciel s'est vu étendre ses fonctionnalités à des applications de front office (CMS, e-commerce, blogs, forums, news, événements, live chat,

job offers...).

Odoo est une suite d'applications business 100% intégrée, de nouvelle génération et Open Source. Ce modèle moderne de conception logicielle permet une évolutivité rapide, universelle et à moindres coûts. En effet, la solution est ainsi soutenue à la fois par un éditeur et par une large communauté de développeurs et d'utilisateurs dans le monde. D'ailleurs, Camptocamp fait partie des plus importants contributeurs sur la solution, et ceci depuis 2006.

## II-2- Odoo: fonctionnalités



Figure III.1 : Fonctionnalités d'Odoo

## II-3 Pourquoi choisir Odoo ?

1. Solution ERP Open Source : le code source est à tout moment disponible, votre ERP n'est donc pas une boîte noire inaccessible ;
2. Simplicité d'utilisation : solution 100% web et ergonomie attractive et intuitive ;
3. Étant donné son architecture moderne et innovante, c'est un ERP flexible, autorisant une mise en place en plusieurs phases en fonction de l'évolution de vos besoins ;
4. Solution ERP est très facilement inter-opérable avec n'importe quel logiciel et/ou service tiers (notamment via le connecteur générique développé par Camptocamp);

5. Repose d'une part sur un éditeur solide comptant plus de 300 collaborateurs, une communauté dynamique et structurée proposant plus de 8'000 modules communautaires et un réseau mondial de plus de 600 partenaires certifiés.

### **III -module RH et RH holidays**

#### **III-1 module RH**

ERP dispose d'un module RH (ressources humaines) consistant à gérer tout ce qui est en rapport avec l'environnement du salarié qu'il soit ouvrier, employé ou cadre. Un progiciel hébergé sur le Web conçu pour répondre aux attentes des professionnels. Un outil gestion entreprise complet et performant abordant un ensemble de sujets lié au domaine social avec des mises à jour ponctuelles. Le module des ressources humaines concerne les autres aspects des RH, comprenant moins de contraintes légales mais plus sujet à une politique de gestion. La gestion de la carrière du travailleur, de son recrutement jusqu'à la pension, y est en particulier développée. Sont ici gérées les données personnelles (âge, adresse, famille...).

En effet, la gestion des Ressources Humaines détermine la stratégie de fonctionnement de l'entreprise et le suivi des performances car elle comprend :

- La gestion du recrutement et des affectations et La gestion sociale.
- La gestion des rémunérations et des carrières et L'appréciation de la performance.

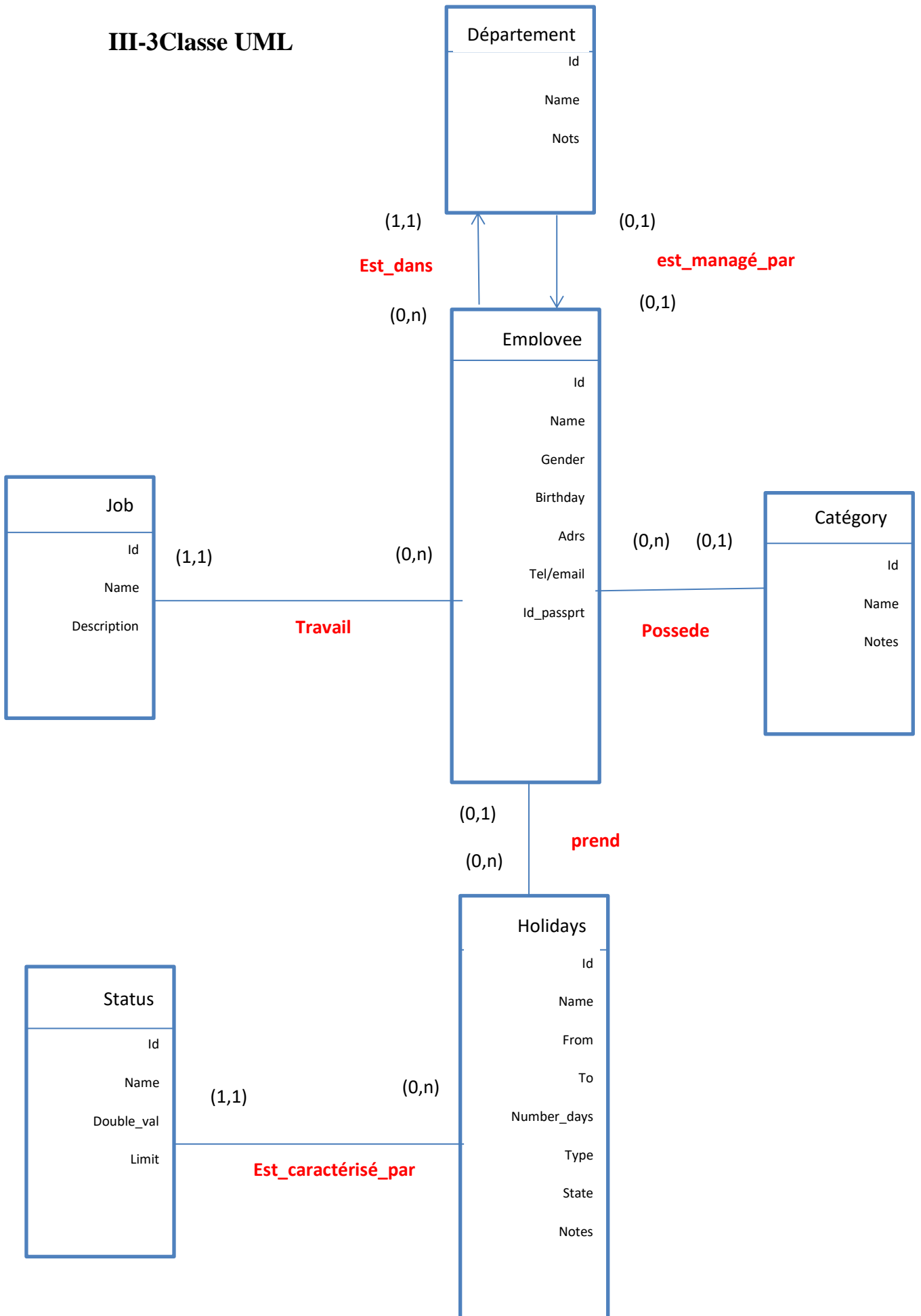
Mais cette définition est loin d'être exhaustive, étant donné que la gestion RH s'étend aussi aux domaines suivants :

- La responsabilité sociale de l'entreprise et La gestion du temps de travail de chaque salarié.
- La gestion du dialogue social et des connaissances et L'hygiène et la sécurité.

#### **III-2module RHholidays**

Le module de gestion des congés d'Odoo est aussi simple qu'efficace. Chaque employé se connecte et réalise ses demandes de congés. Les responsables de service sont ensuite en charge d'approuver ou de refuser les congés. La gestion des congés vous renseigne sur la disponibilité de vos employés. Les employés saisissent leurs souhait de congés qui est ensuite soumis à l'approbation de leur supérieur hiérarchique. Le calendrier de chaque employé est mis à jour.

III-3Classe UML



## **IV-Présentation Sémantique OWL**

Pour la spécification de l'ontologie, le langage OWL est le plus adéquat, pour l'implémentation, nous utilisons Protégé comme un éditeur d'ontologie qui génère un fichier en sortie sous le format OWL basé sur la syntaxe Xml compréhensible et traitable par la machine.

Le langage d'ontologie Web OWL [25] est conçu pour décrire et représenter un domaine de connaissance spécifique, en définissant des classes de ressources ou objets et leurs relations ; ainsi que de définir des individus et affirmer des propriétés les concernant et de raisonner sur ces classes et individus dans la mesure où le permet la sémantique formelle du langage OWL.

OWL est un standard basé sur la logique de descriptions [11], il est construit sur RDF et RDFS et utilise la syntaxe RDF/XML.

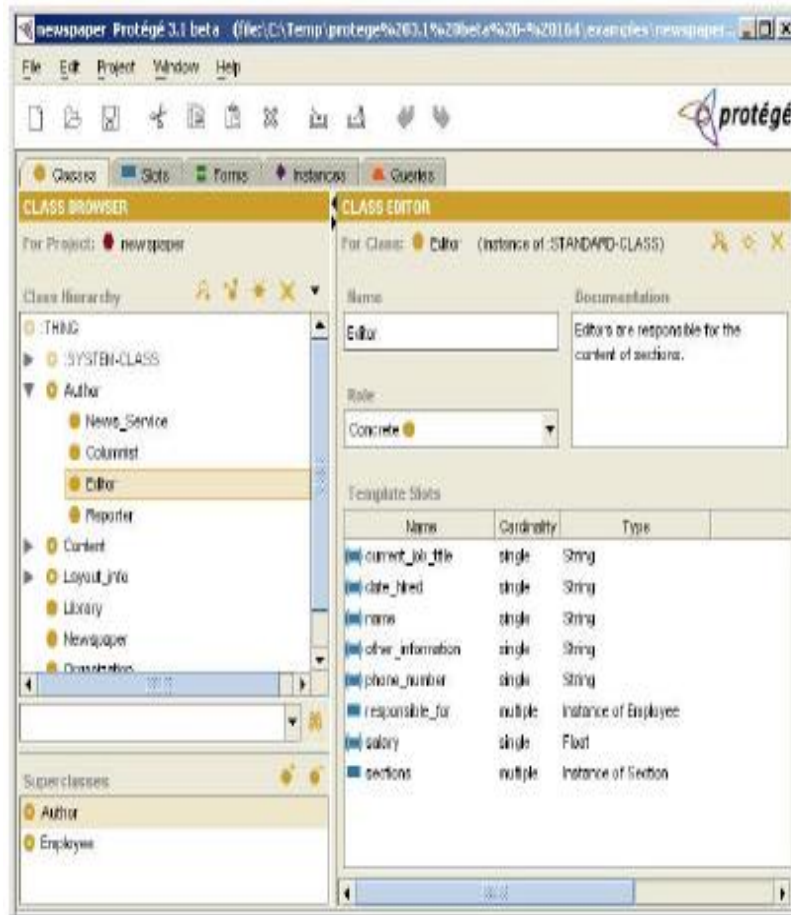
Le langage OWL permet d'étendre les technologies de base (XML, RDF, RDFS) pour apporter :

- ✓ Plus d'interopérabilité (équivalences) ;
- ✓ Plus de raisonnements (logique de description) ;
- ✓ Plus d'évolution (intégration d'ontologies).

Les ontologies OWL se présentent, généralement, sous forme de fichiers texte et de documents OWL.

### **IV-1-Protégé**

Protégé est un système auteur pour la création d'ontologies. Il a été créé à l'université Stanford et est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique. Protégé est développé en Java. Il est gratuit et son code source est publié sous une licence libre (la Mozilla Public License). Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, etc. Il possède plusieurs concurrents tels que Hozo, OntoEdit et Swoop. Il est reconnu pour sa capacité à travailler sur des ontologies de grandes dimensions.



PROTEGE permet la création, la visualisation et la manipulation d'ontologies dans différents formats.

L'onglet "Classes" permet de définir les concepts de l'ontologie comme une hiérarchie de classes et de leur affecter des propriétés. L'onglet "Slots" permet de spécifier ces propriétés.

Des onglets supplémentaires peuvent être développés par les utilisateurs pour répondre à leurs besoins spécifiques.

Figure III.2 : L'éditeur d'ontologie Protégé

#### IV-2- Les Classes (concepts)

Une classe définit un groupe d'individus possédant des caractéristiques similaires. L'ensemble des individus d'une classe est désigné par le terme extension de classe, chacun de ces individus étant alors une instance de la classe. Chaque individu du monde OWL est un membre de la classe *owl : Thing*. Chaque classe définie par l'utilisateur est donc implicitement une sous-classe d'*owl : Thing* [25][20].

CLASSES	Description
EMPLOYEE	Salarié(e) qui travaille dans un bureau, une administration, un magasin, ou chez un particulier
CATEGORY	Ensemble de personnes ou de choses de même nature : Les catégories de fonctionnaires
JOB	Un emploi, consiste à utiliser des personnes actives de la population à des activités économiques.
DEPARTEMENT	Un département peut être une circonscription administrative, ou une division d'un ensemble administratif plus grand.
HOLIDAYS	Période de repos, durant un certain nombre de jours, accordée à des employés dans une entreprise.
STATUS	Position occupée par un individu dans une certaine hiérarchie

Tableau III.1 : les classes et leur description

### IV-3- Les Propriétés de Type de Données

Les propriétés de types de données (*owl:DatatypeProperty*), qui sont des relations entre des instances de classes, des littéraux RDF [11] et des types de donnée du schéma XML[3]. Les propriétés de types de données recouvrent des littéraux RDF ou des types simples définis conformément aux types de données du schéma XML. Le langage OWL utilise la plupart des types de données intégrés du schéma XML [25][20][3].

**domain** : classes pour lesquelles est définie la propriété,

**range** : classes reliées par la propriété au domain.

Domain	Propriétés de Type de Données	range
EMPLOYEE	NAME	STRING
EMPLOYEE	GENDER	STRING
EMPLOYEE	BIRTHDAY	DATA

EMPLOYEE	ADRESSE	STRING
EMPLOYEE	MOBILE_PHONE	STRING
EMPLOYEE	WORK_MAIL	STRING
EMPLOYEE	PASSPORT_ID	STRING
EMPLOYEE	NOTES	STRING
CATEGORY	NAME	STRING
CATEGORY	NOTES	STRING
JOB	NAME	STRING
JOB	DESCRIPTION	STRING
DEPARTEMENT	NAME	STRING
DEPARTEMENT	NOTES	STRING
HOLIDAY	NAME	STRING
HOLIDAY	FROM	DATETIME
HOLIDAY	TO	DATETIME
HOLIDAY	NUMERO_OF_DAYS	DATETIME
HOLIDAY	TYPE	STRING
HOLIDAY	STATE	STRING
HOLIDAY	NOTES	STRING
STATUS	NAME	STRING
STATUS	DOUBLE_VALIDATION	BOOLEAN
STATUS	LIMIT	BOOLEAN

Tableau III.2 : Les Propriétés de Type de Données et leur domain.range

## IV-4- Les Propriétés d'Objets

Les propriétés d'objets (*owl : ObjectProperty*), qui sont des relations entre les instances de deux classes. Il existe plusieurs façons de restreindre la relation, lorsque nous définissons une propriété. Nous pouvons définir un domaine et une image. Nous pouvons définir la propriété comme une spécialisation (sous-propriété) d'une propriété existante.

DOMAIN	Propriétés d'Objets	RANGE
HOLYDAYS	EST_PREND_PAR	EMPLOYEE
EMPLOYEE	TRAVAIL	JOB
EMPLOYEE	POSSEDE	CATEGORIE
EMPLOYEE	EST_DANS	DEPARTEMENT
DEPARTEMENT	Est_MANAGE_PAR	EMPLOYEE
HOLYDAYS	EST_CHARACTERISE_PAR	STATUS

**Tableau III.3 : Les propriétés d'objets et leur domain.range**

IV-5-Graphe OWL

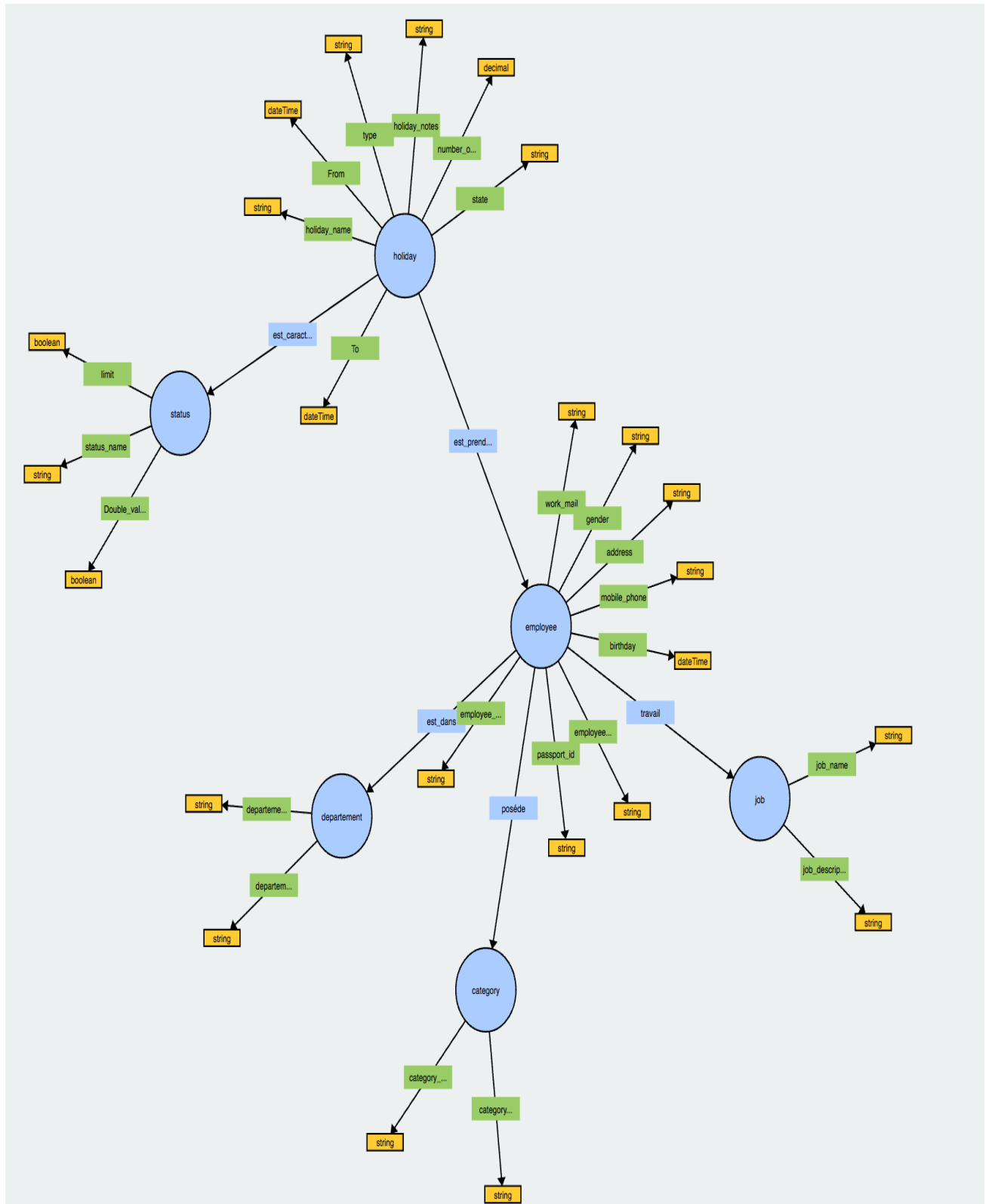


Figure III.3: Graphe OWL

## IV-6- Mapping RDB TO OWL

Il existe plusieurs outils et langages pour la gestion des ontologies. Parmi ces outils NetBeans IDE et Jena.

Jena est un API java open source développé par le laboratoire de HewlettPackard permettant la lecture et la manipulation des ontologies décrites en RDFS ou en OWL. Jena est une bibliothèque de classes Java qui facilite le développement d'applications pour le web sémantique :

- Manipulation de déclarations RDF.
- Lecture et écriture RDF/XML, Notation 3.
- Stockage en mémoire ou sur disque de connaissances RDF.
- Interpréteur SPARQL.
- Gestion d'ontologies: RDF-Schema, OWL.
- Logiciel libre (licence Apache) développé (initialement) par HP.

```

151  OntClass employee = m.getOntClass( NS + "employee" );
152  OntProperty emplName = m.getDatatypeProperty( NS + "employee_name" );
153  OntProperty emplNotes = m.getDatatypeProperty( NS + "employee_notes" );
154  OntProperty emplGender = m.getDatatypeProperty( NS + "gender" );
155  OntProperty emplBirthday = m.getDatatypeProperty( NS + "birthday" );
156  OntProperty emplAddress = m.getDatatypeProperty( NS + "address" );
157  OntProperty emplMobile = m.getDatatypeProperty( NS + "mobile_phone" );
158  OntProperty emplMail = m.getDatatypeProperty( NS + "work_mail" );
159  OntProperty emplPassport = m.getDatatypeProperty( NS + "passport_id" );
160  OntProperty emplDepartement = m.getObjectProperty( NS + "est_dans" );
161  OntProperty emplCategory = m.getObjectProperty( NS + "posède" );
162  OntProperty emplTravail = m.getObjectProperty( NS + "travail" );
163
164  s = c.createStatement();
165
166  rs=s.executeQuery("select id,name_related,notes,gender,birthday,mobile_phone,work_email,passport_id,department_id,job_id from employee");
167
168  while(rs.next())
169  {
170
171      Individual emplInst = m.createIndividual( NS + "employee"+rs.getObject(1), employee );
172
173      Individual emplDep=m.getIndividual(NS+"departement"+rs.getObject(9).toString());
174      //Individual emplCat=m.getIndividual(NS+"category"+rs.getObject(10).toString());
175      Individual emplJob=m.getIndividual(NS+"job"+rs.getObject(10).toString());
176
177      emplInst.addProperty( emplName, ResourceFactory.createTypedLiteral(rs.getObject(2).toString(), XSDDatatype.XSDstring));
178      emplInst.addProperty( emplNotes, ResourceFactory.createTypedLiteral(rs.getObject(3).toString(), XSDDatatype.XSDstring));
179      emplInst.addProperty( emplGender, ResourceFactory.createTypedLiteral(rs.getObject(4).toString(), XSDDatatype.XSDstring));
180      emplInst.addProperty( emplBirthday, ResourceFactory.createTypedLiteral(rs.getObject(5).toString(), XSDDatatype.XSDstring));
181      emplInst.addProperty( emplAddress, ResourceFactory.createTypedLiteral(rs.getObject(6).toString(), XSDDatatype.XSDstring));
182      emplInst.addProperty( emplMobile, ResourceFactory.createTypedLiteral(rs.getObject(7).toString(), XSDDatatype.XSDstring));
183      emplInst.addProperty( emplPassport, ResourceFactory.createTypedLiteral(rs.getObject(8).toString(), XSDDatatype.XSDstring));
184      emplInst.addProperty( emplDepartement, emplDep);
185      //emplInst.addProperty( emplCategory, emplCat);
186      emplInst.addProperty( emplTravail, emplJob);
187
188

```

Figure III.4 : Développement mapping en JenaAPI

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
  <owl:Ontology rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr"/>
  <owl:Class rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#employee"
  />
  <owl:Class rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#job"/>
  <owl:Class rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#status"
  />
  <owl:Class rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#holiday"
  />
  <owl:Class rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#category"
  />
  <owl:Class rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#departement"
  />
  <owl:ObjectProperty rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#est_prend_par">
    <rdfs:range rdf:resource="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#employee"/>
    <rdfs:domain rdf:resource="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#holiday"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#est_caractérisé_par">
    <rdfs:range rdf:resource="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#status"/>
    <rdfs:domain rdf:resource="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#holiday"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/

```

Figure III.5 : Résultat du mapping(1)

```

<employee rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#employee2"
">
  <est_dans rdf:resource="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#departement4"/>
  <address>\N</address>
  <gender>\N</gender>
  <mobile_phone>antoine@openerp.com</mobile_phone>
  <birthday>\N</birthday>
  <passport_id>\N</passport_id>
  <employee_name>Antoine Langlais</employee_name>
  <travail>
    <job rdf:about="http://www.semanticweb.org/macbookpro/ontologies/2017/4/oddo_hr#job2">
      <job_description>Bachelor, master or engineering degree in Computer Science or
        equivalent by experience
        Preferably at least 1 years of experience
        Interest for enterprise application development
        Customer-minded
        Willing to travel abroad occasionally for short term missions.
        Passion for the Internet and its culture
        Quick and autonomous learner, problem-solving personality, enthusiastic when faced with
        technical challenges
        Team spirit and good communication
        Required skills:
        Good knowledge of object oriented programming, object modeling, relational databases, Unix/
        Linux platform
        Fluent in English, especially read and written
        Nice-to-have skills:
        Good knowledge of Python
        Good knowledge of HTML and Javascript
        Knowledge of UML-like modeling
        Good language skills, other than English (Dutch and French preferred, others welcome)
    </job_description>
  </travail>

```

Figure III.6 : Résultat du mapping(2)

## **V - Conclusion**

Dans ce travail, nous avons essayé de décrire un cadre général capable de gérer l'interopérabilité sémantique entre différents ERPs basés sur les ontologies. Nous avons appliquées certaines techniques dans notre approche pouvant assurer la coopération entre les ontologies (données) pour réaliser l'interopérabilité sémantiques, tels que la fusion et le mapping, elles sont utilisées pour permettre le partage des données entre des bases de connaissance hétérogènes et aussi bien la réutilisation des informations de ces bases et de minimiser la perte d'informations afin d'assurer une coopération d'un système d'information interopérable.

## **Conclusion général**

Dans ce travail, nous avons présenté notre approche fondée sur des techniques qui sont offertes pour réaliser l'interopérabilité sémantique. Les principales étapes de notre proposition sont :

- la définition d'une architecture sémantique basée sur les ontologies.
- la définition et la création de l'ontologie.
- la génération de l'ontologie locale par la technique de fusion d'ontologies.
- la génération de l'ontologie globale à partir des ontologies locales par la technique de mapping.
- l'interrogation du modèle par un langage de requête SPARQL.

L'intégration sémantique concerne les données, les applications et les processus. Ceci nécessite une approche basée ontologie pour le résoudre.

Les ontologies peuvent jouer un rôle très important dans la représentation de la sémantique des informations. Nous avons défini les notions de base en particulier l'ontologie, les concepts et relations, le corpus d'un domaine d'application et nous avons présenté quelques méthodologies et outils de construction dans l'ingénierie ontologique. Comme les domaines d'application de l'ingénierie ontologique sont très variés, nous avons fait une projection sur le domaine qui nous intéresse qui est l'interopérabilité des systèmes d'information. Nous avons présenté les approches mono-ontologie, multi-ontologies et l'approche hybride qui se présente comme une bonne solution dans un système d'interopération.

Nous proposons donc, une ontologie pour chaque système participant dans la coopération (ontologie locale), une ontologie universelle comme vocabulaire commun qui permet de réduire les divergences terminologiques des ontologies locales et par conséquent de simplifier l'identification des correspondances entre les concepts.

Nous avons vu en premier temps les notions de base de l'ontologie, les méthodologies de développement, les formalismes de représentation et les langages de spécification. Ensuite, présentation générale sur le Progiciel de Gestion Intégré ERP. Le dernier chapitre donne une vue de fond de notre système, à travers les outils de programmation ainsi que les structures de données et les méthodes utilisés.

## **Bibliographie**

- [01] Psyché V. (2003). État de l'art sur l'ontologie - application au téléapprentissage. Rapport technique, Montréal : Télé-université, Centre de recherche LICEF.
- [02] Bechhofer, S., Horrocks, I., Goble, C. et Stevens, R. (2001). OilEd: a Reason-able Ontology Editor for the Semantic Web. Paper presented at the Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, Vienna. Springer-Verlag.
- [03] Farquhar A., Fikes R. ET Rice J. (1996). The Ontolingua Server: Tool for Collaborative Ontology Construction. Paper presented at the Proc. of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, Alberta, Canada.
- [04] Swartout B., Patil R., Knight K. ET Russ T. (1997). Towards Distributed Use of Large Scale Ontologies. Spring Symposium Series on Ontological Engineering, pp.138-148.
- [05] Pinto, H. S., Gomez-Perez, A., and Martins, J. P. (1999). Some issues on ontology integration. In Proceedings of the Workshop on Ontologies and Problem Solving Methods during IJCAI-99, Stockholm, Sweden.
- [06] B. Meyer. Eiffel: the language. Prentice-Hall, 1992.
- [07] Robert M. MacGregor. Using a description classifier to enhance deductive inference. In Proceedings Seventh IEEE Conference on AI Applications, pages 141–147, 1991.
- [08] A. Borgida, Brachman a R. J., D. L. McGuinness, and L. A. Resnick. Classic: A structural data model for objects. In ACM SIGMOID International Conference on Management of Data, Portland, Oregon, USA, 1989.
- [09] M.R. Quillian. Semantic Memory. In Semantic Information Processing, pages 227–270. MIT Press, 1968.
- [10] J. Sowa. Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, 1984.
- [11] M. Kifer, G. Lausen ET J. Wu. Logical Foundations of Object-Oriented and FrameBased Languages. Journal of the ACM, 42(4):741–843, 1995.

- [12] M. L. Ginsberg. Knowledge Interchange Format: the KIF of death. *AI Magazine*, 12(3):57–63, 1991.
- [13] A. Gomez-Perez, M. Fernandez-Lopez Et O. Corcho. *Ontological Engineering*. Springer-Verlag, Advanced Information and Knowledge Processing, 2003.
- [14] V.K. Chaudhri, A. Farquhar, R. Fikes, P.D. Karp ET J.P. Rice. OKBC: a Programmatic Foundation for Knowledge Base Interoperability. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98)*, pages 600–607. MIT Press, 1998.
- [15] D. Fensel, I. Horrocks, F. Van Harmelen, S Decker, M. Erdmann ET M. Klein. Oil in anutshell. In *Proceedings of European Knowledge Acquisition Workshop (EKAW'2000)*.
- [16] T. Bray, J. Paoli, C.M. Sperberg-McQueen and E. Maler, Editors. *Extensible Markup Language (XML) 1.0, Second Edition*, *World Wide Web Consortium*. 6 October 2000. <http://www.w3.org/TR/REC-xml>.
- [17] S. Agrawal, S. Chaudhuri, L. Kollar, A. Marathe, V. Narasayya, and M. Syamala. Database tuning advisor for microsoftsql server 2005. In *Proceedings of the International Conference on Very Large Databases, (VLDB'04)*, pages 1110–1121, 2004.
- [18] M. Christel and K. Kang. *Issues in requirements elicitation*. Technical report, 1992.
- [19] Yigal Arens, Chun-Nan Hsu, and Craig A. Knoblock. Query processing in the SIMS information mediator. In *Advanced Planning Technology*. AAAI Press, California, USA, 1996.
- [20] F-A. Blain, «Présentation générale des ERP et leur architecture modulaire », <http://fablain.developpez.com/tutoriel/presenterp/>, 20/03/2006.
- [21] S. BAUDRY, «Développement sur l'ERP OfbizNéogia », Rapport de stage, Polytechnique de l'Université de Tours, 2005.
- [22] L. Gharsallah, « Impact de l'ERP sur la performance : cas d'IGL », Mastère Professionnel Université de Sfax, 2007.
- [23] C. Lassiette, « Les progiciels de gestion intégrée, Article, Centre de Recherche en Economie et Gestion (CREG) Paris », [http://www.creg.acversailles.fr/article.php3?id\\_article=209,27\09\2007](http://www.creg.acversailles.fr/article.php3?id_article=209,27\09\2007).

[24] J-L. Lequeux, «Manager avec les ERP », Editions d'organisation, 1998.

[25] Nadjib Mesbahi, « Une approche à base d'agent pour la modélisation d'un progiciel de gestion integer(ERP) », SIIE'2009 : 2ème Conférence Internationale "Systèmes d'Information et Intelligence Economique" 12-14 Février 2009 à Hammamet - Tunisie, avec une évaluation favorable, 2009.

[26] L. Neyer S. Mourlon, «Tout ce que nous avons voulu savoir sur les ERP», Les Annales des Mines de Paris, 2002.

[27], R., Fikes, R.E., Finin, T., Gruber, T., Senator, T., Swartout, W.R., « Enablingtechnology for knowledge sharing », AI Magazine, 1991

[28] PGPP, « Progiciel de Gestion pour la PME Principaux ERP propriétaires et Open Source », version 1.0, 2006.