



HAL
open science

Multi-utilisation de données complexes et hétérogènes : application au domaine du PLM pour l'imagerie biomédicale

Cong Cuong Pham

► **To cite this version:**

Cong Cuong Pham. Multi-utilisation de données complexes et hétérogènes : application au domaine du PLM pour l'imagerie biomédicale. Mécanique [physics.med-ph]. Université de Technologie de Compiègne, 2017. Français. NNT : 2017COMP2365 . tel-01982301

HAL Id: tel-01982301

<https://theses.hal.science/tel-01982301v1>

Submitted on 15 Jan 2019

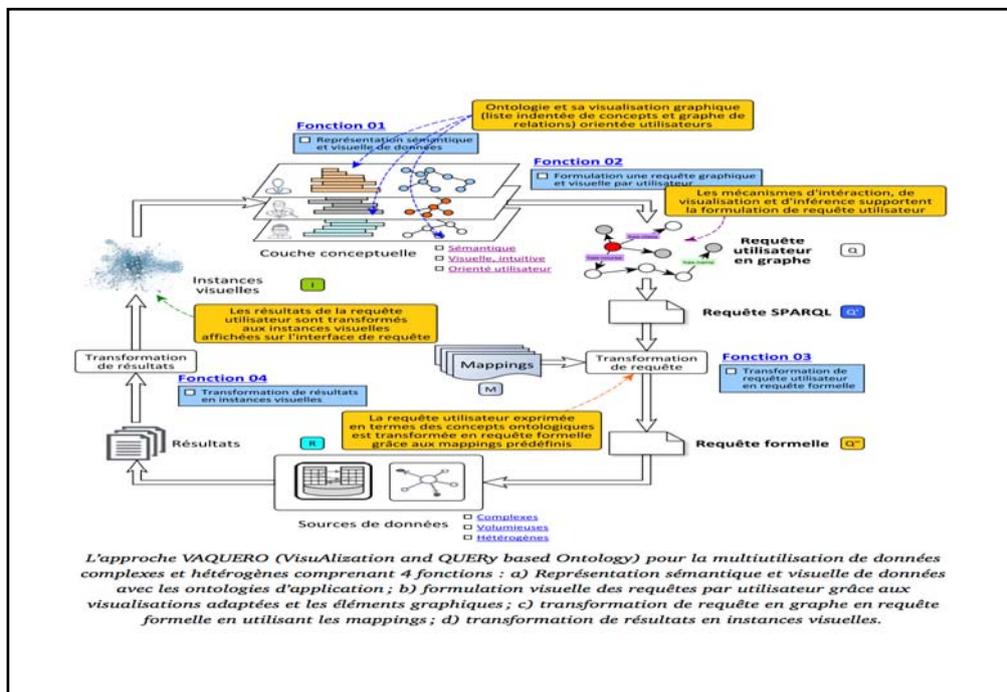
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Par Cong Cuong PHAM

Multi-utilisation de données complexes et hétérogènes : application au domaine du PLM pour l'imagerie biomédicale

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenu le 15 juin 2017

Spécialité : Mécanique Avancée : Unité de recherche en Mécanique - Laboratoire Roberval (UMR-7337)

D2365

UNIVERSITÉ DE TECHNOLOGIE DE COMPIEGNE
ÉCOLE DOCTORALE

THÈSE

Doctorat
« **Mécanique avancée** »

par

Cong Cuong PHAM

MULTI-UTILISATION DE DONNÉES COMPLEXES ET HÉTÉROGÈNES :
APPLICATION AU DOMAINE DU PLM POUR L'IMAGERIE BIOMÉDICALE

Thèse dirigée par Alexandre DURUPT et Nada MATTA

Thèse soutenue le 15 Juin 2017 devant le jury composé de :

SANDRA BRINGAY	PR, Université Paul Valéry Montpellier III	(Rapporteur)
LAURENT GENESTE	PR, Ecole Nationale d'Ingénieur de Tarbes	(Rapporteur)
JULIE LE CARDINAL	PR, Centrale Supélec	(Examineur)
JULIEN LE DUIGOU	EC, Université de Technologie de Compiègne	(Examineur)
FRÉDÉRIC NOEL	PR, Grenoble INP	(Examineur)
GUILLAUME DUCELLIER	EC, Université de Technologie de Troyes	(Examineur)
ALEXANDRE DURUPT	MdC, Université de Technologie de Compiègne	(Co Directeur)
NADA MATTA	EC HDD, Université de Technologie de Troyes	(Co Directeur)
PHILIPPE BOUTINAUD	DR, Fealinx	(Invité)

RÉSUMÉ

L'émergence des technologies de l'information et de la communication (*TIC*) au début des années 1990, notamment internet, a permis de produire facilement des données et de les diffuser au reste du monde. L'essor des bases de données, le développement des outils applicatifs et la réduction des coûts de stockage ont conduit à l'augmentation quasi exponentielle des quantités de données au sein de l'entreprise. Plus les données sont volumineuses, plus la quantité d'interrelations entre données augmente. Le grand nombre de corrélations (*visibles* ou *cachées*) entre données rend les données plus entrelacées et complexes. Les données sont aussi plus hétérogènes, car elles peuvent venir de plusieurs sources et exister dans de nombreux formats (*texte, image, audio, vidéo, etc.*) ou à différents degrés de structuration (*structurées, semi-structurées, non-structurées*). Les systèmes d'information des entreprises actuelles contiennent des données qui sont plus massives, complexes et hétérogènes.

L'augmentation de la complexité, la globalisation et le travail collaboratif font qu'un projet industriel (*conception de produit*) demande la participation et la collaboration d'acteurs qui viennent de plusieurs domaines et de lieux de travail. Afin d'assurer la qualité des données, d'éviter les redondances et les dysfonctionnements des flux de données, tous les acteurs doivent travailler sur un référentiel commun partagé. Dans cet environnement de multi-utilisation de données, chaque utilisateur introduit son propre point de vue quand il ajoute de nouvelles données et informations techniques. Les données peuvent soit avoir des dénominations différentes, soit ne pas avoir des provenances vérifiables. Par conséquent, ces données sont difficilement interprétées et accessibles aux autres acteurs. Elles restent inexploitées ou non exploitées au maximum afin de pouvoir les partager et/ou les réutiliser.

L'accès aux données (ou la *recherche de données*), par définition est le processus d'extraction des informations à partir d'une base de données en utilisant des requêtes, pour répondre à une question spécifique. L'extraction des informations est une fonction indispensable pour tout système d'information. Cependant, cette dernière n'est jamais facile car elle représente toujours un goulot majeur d'étranglement pour toutes les organisations ([Soylu et al. 2013](#)). Dans l'environnement de données complexes, hétérogènes et de multi-utilisation de données, fournir à tous les utilisateurs un accès facile et simple aux données devient plus difficile pour deux raisons :

- **Le manque de compétences techniques** : Pour formuler informatiquement une requête complexe (les *requêtes conjonctives*), l'utilisateur doit connaître la structuration de données, c'est-à-dire la façon dont les données sont organisées et stockées dans la base de données. Quand les données sont volumineuses et complexes, ce n'est pas facile d'avoir une compréhension approfondie sur toutes les dépendances et interrelations entre données, même pour les techniciens du système d'information. De plus, cette compréhension n'est pas forcément liée au savoir et savoir-faire du domaine et il est donc, très rare que les utilisateurs finaux possèdent les compétences suffisantes.
- **Différents points de vue des utilisateurs** : Dans l'environnement de *multi-utilisation de données*, chaque utilisateur introduit son propre point de vue quand il ajoute des

nouvelles données et informations techniques. Les données peuvent être nommées de manières très différentes et les provenances de données ne sont pas suffisamment fournies. Par conséquent, ces données sont difficilement interprétées et accessibles par les autres acteurs et elles restent donc inexploitées ou non exploitées au maximum pour le partage et la réutilisation.

Les travaux de thèse présentés dans le manuscrit visent à améliorer la multi-utilisation de données complexes et hétérogènes par des acteurs métiers en leur fournissant un accès sémantique et visuel aux données.

Nous constatons que, bien que la conception initiale des bases de données ait pris en compte la logique du domaine (en utilisant le *modèle entité-association* par exemple), il est de pratique courante de modifier cette conception afin de l'adapter à des besoins techniques spécifiques. Par conséquent, la conception finale est souvent une forme qui diverge de la structure conceptuelle originale et il existe une nette distinction entre les connaissances techniques nécessaires pour extraire des données et celles que les acteurs métiers ont pour interpréter, traiter et utiliser les données (Soylu et al. 2013).

Après une étude approfondie des éléments bibliographiques portant sur les outils de gestion de données, la représentation des connaissances, les techniques de visualisation et les technologies du Web Sémantique (Berners-Lee et al. 2001), etc. afin de fournir à différents acteurs métiers d'accéder aux données, nous proposons d'utiliser une représentation plus compréhensive et plus déclarative des données. Cette représentation doit être au niveau sémantique, conceptuel et intégrer des notions du domaine qui sont proches de celles des acteurs métiers. Cette demande peut être assurée en utilisant l'ontologie (Studer et al. 1998, Borst 1997, Grüninger et al. 1995), le pivot du Web Sémantique et de l'Ingénierie des connaissances. Les requêtes utilisateurs formulées sur la représentation sémantique sont donc les requêtes ontologiques. Dans notre approche, nous profitons également des avantages de la visualisation (Keller et Tergan 2005) et des outils d'interactions pour que la formulation des requêtes ontologiques soit visuelle, dynamique et interactive. La méthodologie d'interrogation de données que nous proposons se constitue de quatre fonctions principales :

1. **Représentation sémantique de données.**
2. **Formulation d'une requête graphique et visuelle par utilisateur.**
3. **Transformation de requête utilisateur en requête formelle.**
4. **Transformation de résultats en instanciations visuelles.**

Nous présentons en détail les démarches nécessaires pour mettre en place cette méthodologie. Elle propose un cadre de construction des ontologies et la mise en place de mappings entre ces dernières et le modèle de données.

La Neuroimagerie est un domaine pluridisciplinaire qui étudie le fonctionnement du cerveau à l'aide de techniques issues de l'imagerie médicale (Allanic 2015). Grâce aux nouvelles techniques d'imagerie, les données disponibles à traiter sont de plus en plus complexes et diversifiées. Les données dérivées peuvent facilement atteindre cinq fois le volume de données brutes. Initialement développé par l'industrie manufacturière, le [Product Lifecycle Manage-](#)

ment (PLM) est un intégrateur d'outils et de technologies qui rationalise le flux d'informations au travers des différentes étapes du cycle de vie du produit (John Stark 2015, Terzi et al. 2010). Les solutions PLM permettent de gérer efficacement toutes les données reliées au produit en fournissant des droits d'accès aux données et des mécanismes de contrôle de révision, des modifications et des évolutions effectuées sur les données. Ils contiennent aussi des méthodes supplémentaires permettant d'avoir un environnement collaboratif d'ingénierie. Grâce à ces points forts, les solutions PLM peuvent être utilisées pour gérer la complexité et l'hétérogénéité des données d'Imagerie Biomédicale (Allanic 2015).

Le PLM pour Neuroimagerie est le cadre d'application de la thèse. Pour tester la faisabilité de la méthodologie proposée sur de véritables exemples, nous développons un système de requête nommé *VisuAlisation et QUERy based Ontology* (VAQUERO) permettant aux différents acteurs métiers de l'*Imaging Group Neurofunctional, CEA CNRS Université de Bordeaux (GIN)* de pouvoir formaliser des requêtes selon différents profils d'utilisateur (Pham et al. 2016a;b). Les travaux ont été financés dans le cadre du projet ANR BIOMIST (BIO Medical research Imaging SemanTic data management) impliquant l'*Université de Technologie de Compiègne (UTC)*, l'*Université de Technologie de Troyes (UTT)*, l'Université de Bordeaux, et la société Fealinx (porteur du projet) (Allanic et al. 2017).

ABSTRACT

The emergence of [Information and Communication Technologies \(ICT\)](#) in the early 1990s, especially the Internet, made it easy to produce data and disseminate it to the rest of the world. The strength of new [Database Management System \(DBMS\)](#) and the reduction of storage costs have led to an exponential increase of volume data within enterprise information system. The large number of correlations (*visible* or *hidden*) between data makes them more intertwined and complex. The data are also heterogeneous, as they can come from many sources and exist in many formats (*text, image, audio, video, etc.*) or at different levels of structuring (*structured, semi-structured, unstructured*). All companies now have to face with data sources that are more and more massive, complex and heterogeneous.

Recently, industrial projects (*product design*) require the participation and the collaboration of different actors from various fields and workplaces. To ensure the data quality, avoid redundancy and malfunctioning of data flows, all actors must work on a shared common data repository. In this multi-use of data environment, each user introduces their own point of view when adding new data and technical information. The data may either have different denominations or may not have verifiable provenances. Consequently, these data are difficult to interpret and accessible by other actors. They remain unexploited or not maximally exploited for the purpose of sharing and reuse.

Data access (or *data querying*), by definition, is the process of extracting information from a database using queries to answer a specific question. Extracting information is an indispensable function for any information system. However, the latter is never easy but it always represents a major bottleneck for all organizations ([Soylu et al. 2013](#)). In the environment of multiuse of complex and heterogeneous, providing all users with easy and simple access to data becomes more difficult for two reasons :

- **Lack of technical skills** : In order to correctly formulate a query a user must know the structure of data, ie how the data is organized and stored in the database. When data is large and complex, it is not easy to have a thorough understanding of all the dependencies and interrelationships between data, even for information system technicians. Moreover, this understanding is not necessarily linked to the domain competences and it is therefore very rare that end users have sufficient theses such skills.
- **Different user perspectives** : In the multi-use environment, each user introduces their own point of view when adding new data and technical information. Data can be named in very different ways and data provenances are not sufficiently recorded. Consequently, they become difficultly interpretable and accessible by other actors since they do not have sufficient understanding of data semantics.

The thesis work presented in this manuscript aims to improve the multi-use of complex and heterogeneous data by expert usiness actors by providing them with a semantic and visual access to the data.

We find that, although the initial design of the databases has taken into account the logic of the domain (using the *entity-association model* for example), it is common practice to modify

this design in order to adapt specific techniques needs. As a result, the final design is often a form that diverges from the original conceptual structure and there is a clear distinction between the technical knowledge needed to extract data and the knowledge that the expert actors have to interpret, process and produce data (Soylu et al. 2013).

Based on bibliographical studies about data management tools, knowledge representation, visualization techniques and Semantic Web technologies (Berners-Lee et al. 2001), etc., in order to provide an easy data access to different expert actors, we propose to use a comprehensive and declarative representation of the data that is semantic, conceptual and integrates domain knowledge closed to expert actors. This demand could be ensured by using **ontology** (Studer et al. 1998, Borst 1997, Grüninger et al. 1995), the pivot of the Semantic Web and Knowledge Engineering. The user queries formulated on the semantic representation are thus the ontological queries. In our approach, we also profit the advantages of visualization (Keller et Tergan 2005) and interaction tools to enable the visual and graphical formulation of ontological queries. The proposed data query methodology consists of four main functions :

1. **Data semantics representation,**
2. **Graphical and visual formulation of user queries,**
3. **Semantic queries to formal queries transformation,**
4. **Results to visual instances transformation.**

We present also necessary steps to implement this methodology that focus on the construction of application ontologies and mappings set between the ontologies and the data sources.

Neuroimaging is a multidisciplinary field that studies the functioning of the brain using techniques derived from medical imaging (Allanic 2015). Thanks to the new imaging techniques, the available data to be processed are more and more complex and diversified. Derived data can easily reach five times the volume of raw data. Initially developed by the manufacturing industry, Product Lifecycle Management (**PLM**) is an integrator of tools and technologies that streamlines information flow through different stages of the product lifecycle (John Stark 2015, Terzi et al. 2010). PLM solutions effectively manage all product-related data by providing data access rights and review control mechanisms, changes, and changes to the data. They also contain additional methods for a collaborative engineering environment. With these strengths, PLM solutions can be used to manage the complexity and heterogeneity of Biomedical Imaging data (Allanic 2015).

The PLM for Neuroimaging is used as the application domain. To test the feasibility of the proposed methodology on real examples, we develop a query system called VAQUERO (Pham et al. 2016a;b) (VisuAlisation and QUERy based Ontology) enabling the semantic access to Biomedical Imaging (BMI) data of different scientists at GIN (*Imaging Group Neurofunctional*, CEA CNRS University of Bordeaux) . The project was financed under the ANR BIOMIST project¹ involving UTC, UTT, the University of Bordeaux, and Fealinx (project leader) (Allanic et al. 2017).

1. <http://biomist.fr/jahia/cms/sites/biomist/home.html>

TABLE DES MATIÈRES

TABLE DES MATIÈRES	7
LISTE DES FIGURES	11
1 CONTEXTE ET PROBLÉMATIQUE	18
1.1 CONTEXTE DE L'ÉTUDE	19
1.1.1 Données, base de données et système d'information	19
1.1.1.1 Données	19
1.1.1.2 Base de données	19
1.1.1.3 Système d'information (SI)	19
1.1.2 Augmentation de données complexes et hétérogènes	21
1.1.3 Multi-utilisation de données	24
1.2 PROBLÉMATIQUE	24
1.3 QUESTIONS DE RECHERCHE	26
1.4 HYPOTHÈSE DE RECHERCHE	27
1.5 DOMAINE D'APPLICATION	29
1.5.1 Les solutions PLM	29
1.5.1.1 Historique	29
1.5.1.2 Définitions du PLM	30
1.5.1.3 Fonctionnalités du PLM	32
1.5.1.4 Adaptation du PLM dans les domaines de recherche	33
1.5.2 La Neuroimagerie	33
1.5.3 PLM pour la Neuroimagerie	33
1.6 CONCLUSION	34
2 ÉTAT DE L'ART	36
2.1 DONNÉES ET BASES DE DONNÉES	37
2.1.1 Définitions	38
2.1.2 Typologie des bases de données des SI	38
2.1.3 Construction des bases de données	39
2.1.4 Les utilisateurs de BDDs	40
2.1.5 Limitations d'accès aux BDDs	42
2.2 INTERROGATION DES BASES DE DONNÉES	42
2.2.1 Les langages de requêtes	42
2.2.2 Simplification de l'interrogation les BDDs	42

2.2.2.1	Recherche par mots-clés/ langage naturel	43
2.2.2.2	Les requêtes à facette	44
2.2.2.3	Les systèmes d'interrogation visuels	45
2.3	REPRÉSENTATION DES CONNAISSANCES	46
2.3.1	Donnée, information et connaissance	46
2.3.2	Pourquoi la représentation des connaissances	47
2.3.3	Formes de représentation des connaissances	47
2.3.3.1	Réseau Sémantique (Semantic network)	48
2.3.3.2	Cadre (Frame)	49
2.3.3.3	Description Logique (Logic Description)	49
2.3.3.4	Graphe Conceptuel (Conceptual Graph)	49
2.3.3.5	Logique Floue (Fuzzy Logic)	50
2.3.4	Ontologie	51
2.3.4.1	Définition	51
2.3.4.2	Usage de l'ontologie	55
2.3.4.3	Composants d'une ontologie	57
2.3.4.4	Typologies des ontologies	58
2.3.4.4.1	Selon formes d'apparence	58
2.3.4.4.2	Selon scopes	58
2.3.4.4.3	Selon degrés de formalité	59
2.3.4.5	Langages de représentation des ontologies	61
2.3.4.5.1	Classification par syntaxe	61
2.3.4.5.2	Classification par structure	61
2.3.4.6	Méthodologies de construction des ontologies	63
2.3.4.7	Conclusion	65
2.4	VISUALISATION	65
2.4.1	Visualisation des données	66
2.4.2	Visualisation des informations	67
2.4.3	Visualisation des connaissances	67
2.4.4	Visualisation des ontologies	68
2.5	LA CONSTRUCTION DE REQUÊTE SÉMANTIQUE VISUELLE	74
2.6	WEB SÉMANTIQUE ET INGÉNIERIE DES CONNAISSANCES	77
2.6.1	L'ontologie et le web sémantique	79
2.6.2	La recherche sémantique	79
2.6.3	L'intégration sémantique d'informations	80
2.6.4	Mapping entre BDDs et ontologies	82
2.7	CONCLUSION	87
3	MÉTHODOLOGIE	89
3.1	MÉTHODOLOGIE PROPOSÉE	92
3.1.1	Représentation visuelle de la sémantique du domaine	92
3.1.1.1	Représentation sémantique du domaine par l'Ontologie	92
3.1.1.2	Visualisation des ontologies	93

3.1.1.3	Construction de la représentation sémantique	95
3.1.2	Accès Visuel et sémantique aux données	96
3.1.2.1	Formulation de requête graphique et sémantique par utilisateur	96
3.1.2.1.1	Formulation de requêtes en graphe	98
3.1.2.1.2	Mécanismes d'inférences	99
3.1.2.2	Les conventions de visualisation	100
3.1.2.3	Transformation de requête	102
3.1.2.4	Transformation de résultats	103
3.1.2.5	Mapping entre l'ontologie et les sources de données	103
3.2	LA DÉMARCHE DE CONSTRUCTION DES ONTOLOGIES	106
3.2.1	Le cas d'étude	106
3.2.2	Acquisition de connaissances	107
3.2.2.1	Entretien des experts du domaine	107
3.2.2.1.1	Démarche de l'entretien	108
3.2.2.1.2	Entretien des chercheurs du GIN	108
3.2.2.2	Analyse des modèles conceptuels de données	111
3.2.2.3	Réutilisation des ontologies existantes	113
3.2.3	Modélisation de connaissances	113
3.2.3.1	Démarche de modélisation	113
3.2.3.2	Modélisation de l'ontologie d'application GIN	114
3.2.4	Edition de l'ontologie	116
3.2.5	Validation de l'ontologie	117
3.3	LA DÉMARCHE DE CONSTRUCTION DES MAPPINGS	118
3.3.1	Typologies du mapping	118
3.3.1.1	Création de l'ontologie à partir de la BDD	118
3.3.1.2	Mapping d'une base de données à une ontologie existante	119
3.3.2	Langages du mapping	120
3.3.2.1	Classification des langages	120
3.3.2.2	Liste de langages du mapping	121
3.3.2.3	Évaluation des langages	126
3.3.3	La construction des mappings	127
3.3.3.1	Choix d'un langage pour exprimer les mappings	128
3.3.3.2	Définition des mappings	128
3.3.3.3	Génération, teste et maintien des mappings	129
3.4	CONCLUSION	130
4	APPLICATION	132
4.1	PLM POUR L'IMAGERIE BIOMÉDICALE	133
4.1.1	Adaptation des solutions pour la gestion des données BMI	133
4.1.2	Interrogation de données dans Teamcenter	134
4.1.2.1	Query Builder	135
4.1.2.2	Limites de Query Builder	136
4.2	SYSTÈME DE REQUÊTE VAQUERO	137

4.2.1	Infrastructure	137
4.2.2	Interface de requête	140
4.2.2.1	Mise en page "MDD"	141
4.2.2.2	Mise en page "Classification"	141
4.2.2.3	Mise en page "Ontologie"	143
4.2.3	Deux ontologies dédiées aux utilisateurs	144
4.3	EXEMPLES	145
4.3.1	Exemple 01	145
4.3.2	Exemple 02	147
4.4	DISCUSSION	148
5	CONCLUSION	150
5.1	CONTRIBUTIONS	152
5.2	LIMITATIONS ET PERSPECTIVES	153
5.2.1	Limitations	153
5.2.2	Perspectives	154
5.3	LES CONTRIBUTIONS SCIENTIFIQUES	156
	ACRONYMES	157
	BIBLIOGRAPHIE	158

LISTE DES FIGURES

1.1	Les composants d'un système d'information	20
1.2	Hétérogénéité, complexité et interconnexions entre objets PLM	21
1.3	Les différentes parties prenantes couvertes par les solutions PLM ² .	22
1.4	Croissance de l'univers numérique entre 2010 à la prévision de 2020	22
1.5	Les facteurs de complexité	23
1.6	L'accès aux données avec les requêtes prédéfinies	25
1.7	L'accès aux données avec les requêtes complexes	26
1.8	La démarche idéale de la méthodologie proposée	28
1.9	La première version du SGDT	29
1.10	Evolution du SGDT dans les années 90s	29
1.11	La gestion des informations techniques de l'entreprise étendue	30
1.12	Processus de développement d'un nouveau document	31
1.13	Fonctionnalités du PLM dans chaque phase du cycle vie du produit (John Stark 2015).	32
1.14	Exemple d'une étude dansen Neuroimagerie	34
2.1	L'état de l'art sur les BDDs et les méthodologies de l'interrogation des BDDs	37
2.2	Différents types du MDD	39
2.3	Trois couches de modélisation des données	40
2.4	Les interactions de chaque type d'utilisateurs avec le SGBD	41
2.5	Processus de transformation de la requête en langage naturel à la requête SQL	43
2.6	L'interface de la recherche à facettes	44
2.7	La classification des VQSs	45
2.8	L'état de l'art sur les techniques de visualisation des connaissances	46
2.9	Les transitions de donnée à sagesse supportées par la compréhension	47
2.10	Évolution des techniques de représentation des connaissances	48
2.11	Exemple du réseau sémantique.	49
2.12	Base de connaissance représentée par les cadres.	50
2.13	Le graphe conceptuel utilisé pour exprimer "John va à Boston en bus"	50
2.14	L'ontologie MKS des opérations de fabrication	52
2.15	La construction des ontologies à partir des bases de connaissances (Davis 1986).	54
2.16	Les caractéristiques d'une ontologie	55
2.17	Résumé des avantages de l'ontologie.	57
2.18	Trois façons de représenter d'une ontologie	59
2.19	Ontologie supérieure et Ontologie du domaine.	60

2.20	Classification des ontologies par le degré de formalité.	60
2.21	Les langages d'ontologie traditionnels.	61
2.22	Langage d'ontologie de balisage	61
2.23	Classification des langages d'ontologie par structure.	62
2.24	Classification des langages d'ontologie par le degré de formalité.	62
2.25	Tâches de construction de l'ontologie de la méthodologie METHONTOLOGY.	63
2.26	Le processus de construction des ontologies proposé par Ismail	65
2.27	Liste indentée des concepts.	69
2.28	Noeud-lien visualisation d'une ontologie dans l'OntoViz	70
2.29	WebVOWL - une application web de visualisation d'ontologies	71
2.30	Visualisation agrandissable d'ontologie en Jambalaya.	72
2.31	EyeTree et RadialTree visualisation d'ontologies.	72
2.32	L'état de l'art sur la formulation visuelle d'une requête ontologique	73
2.33	Exemple d'une requête SPARQL et sa représentation en RDF-GL	74
2.34	Outil graphique supportant la création des requêtes visuelles.	75
2.35	Exemple d'une requête sémantique à base de l'ontologie.	76
2.36	L'état de l'art sur le Web Sémantique et le mapping entre BDDs et ontologies	77
2.37	Différents types de ressources et de liens dans le Web Sémantique	78
2.38	Avantages de base (essentiels) et fonctionnalités génériques de l'ontologie	80
2.39	Les fonctionnalités spécifiques de l'ontologie dans le Web Sémantique	81
2.40	Le processus de la recherche sémantique	82
2.41	Intégration des données (top-down approche)	83
2.42	Top-down intégration des données basée sur l'ontologie	84
2.44	Trois approches de l'intégration d'informations basées sur l'ontologie	84
2.43	Annotation d'un web dynamique	85
2.45	Accès aux données basé sur l'ontologie	86
2.46	Exemple de Construction de l'ontologie à partir du schéma de la BDD.	86
2.47	Positionnement de la bibliographie par rapport au processus idéal.	88
3.1	L'approche proposée pour la multi-utilisation et l'accès sémantique aux données	91
3.2	Arbre indenté des concepts ontologiques	93
3.3	Représentation graphique des relations sémantiques	94
3.4	Liste indentée et graphe des relations dans une seule vue	95
3.5	Positionnement de la représentation sémantique dans l'approche générale	96
3.6	La formulation graphique des requêtes utilisateur	99
3.7	La représentation graphique d'une requête avec les éléments graphiques	101
3.8	La requête utilisateur en graphe est transformée en requête SPARQL.	101
3.9	La réécriture sur les mappings	102
3.10	La transformation de requête : Un exemple illustré sur les mappings.	102
3.11	Les résultats des requêtes.	103
3.13	Exemple des mappings entre ontologies et BDDs	103
3.12	La transformation de requête et des résultats	104
3.14	Exemple d'un mapping entre deux requêtes	105

3.15	Le modèle de données BMI-LM utilisé pour gérer les données au GIN	106
3.16	Le processus de construction des ontologies d'applications en quatre étapes.	107
3.17	Acquisition de connaissances pour la construction des ontologies d'application.	107
3.18	Processus de recueil des données au GIN	109
3.19	Processus d'exploitation des données au GIN	110
3.20	Processus de mise à jour des données au GIN	110
3.21	Le modèle de données BMI-LM implémenté dans le système PLM Teamcenter	111
3.22	Top concepts ontologiques de l'ontovIP - Project Credible	113
3.23	La modélisation de connaissances	114
3.24	L'arbre des concepts ontologiques proposés pour les chercheurs au GIN.	115
3.25	Graphe des relations entre concepts de l'ontologie proposée	116
3.26	Edition de l'ontologie en utilisant des logiciels d'éditeurs.	117
3.27	La validation de l'ontologie construite	117
3.28	La classification des approches utilisées pour mapper d'une BDD à une ontologie	119
3.29	Niveaux d'intersection entre les domaines couverts par le DB et l'ontologie	119
3.30	Classification de RDB2RDF langages - Michel	120
3.31	Classification de RDB2RDF langages	121
3.32	Un exemple du D2R	121
3.33	La transformation de requête SPARQL en SQL avec le D2RQ	122
3.34	Architecture de mapping avec R2O	122
3.35	Exemple du Relational.OWL	124
3.36	Exemple du mapping R3M	125
3.37	Exemple du mapping R2RML	126
3.38	Exemple d'un outil web utilisé pour définir manuellement des mappings.	129
3.39	La définition des mappings avec Freemind.	129
4.1	Les phases du cycle de vie du produit	133
4.2	Le cycle de vie d'une étude de recherche en Neuroimagerie	133
4.3	Recherche des items par mots-clés	135
4.4	Prédéfinie requête avancé.	135
4.5	Query Builder : L'outil de construction de requêtes dans Teamcenter	135
4.6	Sélection des attributs de la requête.	136
4.7	Espace de travail actif Teamcenter	136
4.8	Le contenu PLM Teamcenter sont vus comme une base de documents XML	137
4.9	L'infrastructure du système de requête VAQUERO.	138
4.10	Accès sémantique aux données, adapté au domaine du PLM pour BMI.	139
4.11	Interface de requête implémentée dans VAQUERO.	141
4.12	La mise en page "MDD" de l'interface de requête.	141
4.13	La classification implémentée dans Teamcenter.	142
4.14	La mise en page "Classification" de l'interface de requête.	143
4.15	La mise en page "Ontologie" de l'interface de requête.	143
4.16	Imaging ontologie d'application pour les chercheurs GIN	144
4.17	Psychologie ontologie d'application pour les chercheurs GIN	144

4.18 Représentation graphique de la requête 01	146
4.19 La requête utilisateur est transformée à la requête xQuery.	146
4.20 Représentation graphique des résultats	147
4.21 La requête 02 est formulée au fur et à mesure en trois étapes.	148

INTRODUCTION GÉNÉRALE

L'émergence des [Technologies de l'Information et de la Communication \(TIC\)](#) au début des années 1990, notamment internet, a permis de produire facilement des données et de les diffuser au reste du monde. Les données croisées sur internet ont largement augmenté ces vingt dernières années ([McAfee et al. 2012](#)). La taille de l'univers numérique est doublée tous les deux ans selon le rapport de l'IDC ([Turner et al. 2014](#)). La vigueur des bases de données, le développement des outils applicatifs et la réduction des coûts de stockage ont conduit à l'augmentation quasi exponentielle des quantités d'informations numériques au sein de l'entreprise. Les entreprises maintenant doivent et devront travailler avec de nombreux pétaoctets de données. Le grand nombre d'interrelations et de corrélations (*visibles* ou *cachées*) rend plus entrelacées les données. Les données peuvent aussi venir de plusieurs sources et exister dans de nombreux formats (*texte, image, audio, vidéo, etc.*) ou à différents degrés de structuration (*structurées, semi-structurées, non-structurées*). ***Les systèmes d'information des entreprises actuels contiennent des données qui sont plus massives, complexes et hétérogènes.***

Les données rendues disponibles dans les systèmes d'information actuels sont non seulement nombreuses, complexes et hétérogènes, mais aussi elles évoluent par l'utilisation quotidienne. Les activités des acteurs du système génèrent une quantité de nouvelles données. L'augmentation de la complexité, la globalisation et la tendance du travail collaboratif, un projet industriel ou de recherche demande souvent la participation et la collaboration d'acteurs qui viennent de plusieurs domaines et de lieux de travail. Afin d'assurer la qualité des données, d'éviter les redondances et les dysfonctionnements des flux de données, tous les acteurs doivent travailler sur un référentiel commun partagé. Dans cet environnement de ***multi-utilisation de données***, chaque utilisateur introduit son propre point de vue quand il ajoute de nouvelles données et informations techniques. Les données peuvent soit avoir des dénominations différentes, soit ne pas avoir des provenances vérifiables. Par conséquent, ces données sont difficilement interprétées et accessibles aux autres acteurs. Elles restent inexploitées ou non exploitées au maximum afin de pouvoir les partager et/ou les réutiliser.

La Neuroimagerie est un domaine pluridisciplinaire qui étudie le fonctionnement du cerveau à l'aide de techniques issues de l'imagerie médicale. Grâce aux nouvelles techniques d'imagerie, les données disponibles à traiter sont de plus en plus complexes et diversifiées. Les données dérivées peuvent facilement atteindre cinq fois le volume de données brutes. Initialement développé par l'industrie manufacturière, le Product Lifecycle Management (*PLM*) est un intégrateur d'outils et de technologies qui rationalise le flux d'informations des différentes étapes du cycle de vie du produit. Les solutions PLM permettent de gérer efficacement

toutes les données reliées au produit en fournissant des droits d'accès aux données et des mécanismes de contrôle des révisions, des modifications et des évolutions effectuées sur les données. Ils contiennent aussi des méthodes supplémentaires permettant d'avoir un environnement collaboratif d'ingénierie et une gestion de l'accès des utilisateurs. Grâce à ces points forts, les solutions PLM peuvent être utilisées pour gérer la complexité et l'hétérogénéité des données d'Imagerie Biomédicale. Cependant, les solutions PLM ne sont pas très flexibles pour une multi-utilisation par l'ensemble des acteurs métiers qui ne sont pas des experts en technologie de l'information (*Information Technology (IT)*).

Dans ce manuscrit de thèse, nous allons proposer une méthodologie permettant une efficace multi-utilisation de données complexes et hétérogènes. Basée sur l'ontologie, le pivot du Web Sémantique et de l'Ingénierie des Connaissances, cette méthodologie ajoute une représentation sémantique des sources de données qui les rendent plus facilement accessibles par l'ensemble des différents acteurs métiers. Le domaine du PLM pour l'Imagerie Biomédicale est le cadre d'application de la thèse. La méthodologie proposée nommée VAQUERO (**Visu**Alisation et **QUERy** based **Ontology**) permet au sein d'une organisation de pouvoir formaliser des requêtes selon différents profil d'utilisateur. La mise en oeuvre de VAQUERO est le coeur méthodologique de cette thèse. Un démonstrateur nommé VAQUERO system a été testé pour valider la faisabilité sur de véritables exemples. Les travaux ont été soutenu par le projet ANR [BIO Medical research Imaging SemanTic data management \(BIOMIST\)](#) impliquant l'UTC, l'UTT, l'Université de Bordeaux, et la société Fealinx (porteur du projet). Ce manuscrit est organisé comme suit :

Le **chapitre 01** présente tout d'abord le contexte d'étude de la recherche qui considère l'accès aux données complexes et hétérogènes dans les systèmes d'information actuels. La problématique que nous abordons est "*la multi-utilisation des données complexes et hétérogènes par différents types d'acteurs métiers*". Puis, nous présentons les hypothèses de recherche. Nous terminons ce chapitre par une introduction du domaine d'application restreint au Product Lifecycle Management pour l'Imagerie Biomédicale.

Le **chapitre 02** présente une étude de la bibliographie approfondie sur les [Systèmes de Gestion des Données Techniques \(SGDT\)](#) et les solutions PLM. Un état de l'art sur l'accès aux données sera présenté pour montrer la nécessité d'une représentation sémantique des sources de données dans le contexte de la multi-utilisation. Une revue de la littérature sur le Web Sémantique et l'approche [Ontology-Based Data Access \(OBDA\)](#) est ensuite amenée pour justifier le choix de l'ontologie comme représentation sémantique des données. Pour conclure, ce chapitre présente des notions principales et des méthodologies de construction des ontologies. Ces éléments seront utilisés comme base pour les chapitres suivants.

Le **chapitre 03** consiste à expliciter la méthodologie que nous avons défini pour répondre aux besoins présentés dans le chapitre **01**. Cette méthodologie est une adaptation de l'approche OBDA dans le contexte du PLM pour l'Imagerie Biomédicale. Nous présentons en détail la démarche nécessaire pour mettre en place cette méthodologie. Cette démarche propose un cadre de construction des ontologies et la mise en place de mappings entre ces dernières et le modèle de données.

Le **chapitre 04** s'intéresse à la validation expérimentale de l'approche proposée. Cette validation est passée par la réalisation d'un démonstrateur VAQUERO system basé sur la méthodologie présentée dans le chapitre **03**. Des exemples de requêtes sur les données BMI seront utilisés pour montrer la faisabilité de notre approche.

Le **chapitre 05** mène une discussion sur les contributions de la thèse, ses limites et les perspectives.

CONTEXTE ET PROBLÉMATIQUE DE THÈSE



SOMMAIRE

1.1	CONTEXTE DE L'ÉTUDE	19
1.1.1	Données, base de données et système d'information	19
1.1.2	Augmentation de données complexes et hétérogènes	21
1.1.3	Multi-utilisation de données	24
1.2	PROBLÉMATIQUE	24
1.3	QUESTIONS DE RECHERCHE	26
1.4	HYPOTHÈSE DE RECHERCHE	27
1.5	DOMAINE D'APPLICATION	29
1.5.1	Les solutions PLM	29
1.5.2	La Neuroimagerie	33
1.5.3	PLM pour la Neuroimagerie	33
1.6	CONCLUSION	34

Ce chapitre délimite tout d'abord la problématique scientifique et le périmètre des travaux de recherche. Pour la problématique scientifique, nous nous concentrons sur la multi-utilisation des données complexes et hétérogènes. Le périmètre de notre étude est restreint au domaine du PLM pour l'Imagerie Biomédicale. Dans la deuxième partie de ce chapitre, avant de proposer l'hypothèse de recherche, deux questions de recherche seront présentées. Ce chapitre sera organisé comme suit :

- **Contexte de l'étude de la thèse**
- **Problématique**
- **Questions et l'hypothèse de recherche**
- **Domaine d'application**

1.1 CONTEXTE DE L'ÉTUDE

Dans cette section, avant de présenter le contexte de l'étude portant sur les données complexes, hétérogènes et la multi-utilisation des données par différents types d'acteurs métiers, nous présentons quelques notions.

1.1.1 Données, base de données et système d'information

1.1.1.1 Données

Nous vivons dans un monde digital où tout ce qui nous entoure est capturé numériquement. L'homme, la société, les organisations sont tous reliés à une source de données. Par définition, une donnée est une unité brute qui peut être "*un symbole ou un fait isolé et non interprété*" (Keller et Tergan 2005). C'est-à-dire, une donnée représente un fait ou une déclaration d'événement sans aucune relation avec d'autres données. Une donnée est le résultat direct d'une mesure et elle représente les propriétés d'un objet ou d'un événement (Ackoff 1989). Les données représentent le format brut le plus bas de l'information et de la connaissance. Dans le monde informatique, nous nous référons aux données généralement en termes de lignes et colonnes de valeurs organisées qui représentent une ou plusieurs entités et leurs attributs.

1.1.1.2 Base de données

Les données sont organisées, stockées dans une base de données avec un support informatique selon un modèle prédéfini. Les bases de données sont organisées et structurées de manière à pouvoir facilement consulter et modifier leurs contenus. Elles sont créées pour être partagées par de nombreux utilisateurs (*humains* ou *informatiques*) et si nécessaire, pour être consultées de manière confidentielle. Un système de gestion de base de données (SGBD) constitue l'interface entre ces utilisateurs et une base de données. Sa tâche principale est de fournir à chaque utilisateur un accès efficace, simple et cohérent à des données partagées.

1.1.1.3 Système d'information (SI)

Toute personne ou organisation qui possède les compétences et les outils permettant de gérer efficacement des sources des données porte un avantage considérable en termes de compé-

titivité et de développement (Soylu et al. 2013). Mettre en place un système d'information efficace est capital pour toute organisation. Un système d'information, par définition, est "un ensemble organisé de ressources (matériels, logiciels, personnel, données et procédures) qui permet de collecter, regrouper, classifier, traiter et diffuser de l'information sur un environnement donné" (De Courcy 1992). Autrement dit, les SI utilisent des outils (**logiciels**) pour interpréter les données provenant d'une base de données pour les transformer en information. La Figure 1.1 illustre cinq ressources (*composants*) différentes d'un SI.

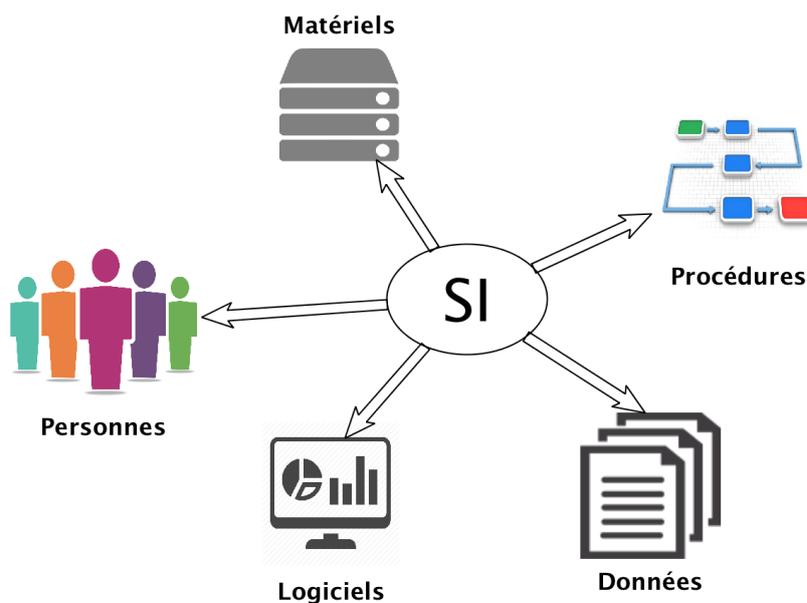


FIGURE 1.1 : Les composants d'un système d'information (Silver et al. 1995).

- **Personnes** ou **acteurs** qui sont soit la direction des systèmes d'information, soit les informaticiens (*analystes, programmeurs, etc.*), ou encore les utilisateurs finaux. Ces derniers peuvent être les employés qui pour la réalisation de leurs tâches, utilisent l'information produite par le système ou alimentent le système en données nouvelles. Le système d'information est conçu à leur attention.
- **Matériels** qui sont les technologies numériques de l'information (*ordinateurs et unité périphériques, réseaux, etc.*)
- **Logiciels** et procédures qui sont enregistrés dans les ordinateurs. Ces logiciels assurent le fonctionnement du SI.
- **Données** qui sont la ressource la plus importante. Ces ressources essentielles matérialisent l'information détenue par l'organisation.

Les systèmes d'information peuvent être divisés en deux par leur finalité principale : Les SI supports d'opérations et ceux qui supportent de gestion (Reix et al. 2016). Deux fonctions principale d'un SI sont la **production d'information** incluant la *collection*, le *traitement* et la *transmission* des informations ; et la **mise en oeuvre d'outils de gestion** incluant les fonctions : *technologique, économiques, et sociale*.

Dans le cadre cette thèse, le SI considéré est un système PLM. Dans l'industrie, depuis les années 90s, pour être toujours en compétition avec les autres concurrents, les entreprises

automobiles et aéronautiques ont commencé à chercher des moyens permettant d'accélérer le processus de développement de produits. Les produits sont devenus peu à peu plus complexes, les méthodes et outils proposés doivent être capables de gérer une énorme quantité de données complexes et hétérogènes. Ces données peuvent représenter une partie physique, un assemblage ou un document de conception. Elles peuvent aussi être des exigences clients ou fournisseurs ou des documents attachés qui contiennent des propriétés sur la conception du produit : numéro document et pièces, catégorie d'article, révision, titre, unité de mesure, les matériaux, coût, de masse, les pièces jointes, etc. À travers toutes les phases du cycle de vie du produit et dans un environnement multi-acteurs. Les relations entre données sont souvent nombreuses et complexes. La Figure 1.2 représente différents types de données utilisées et générées dans le processus de conception d'une voiture.

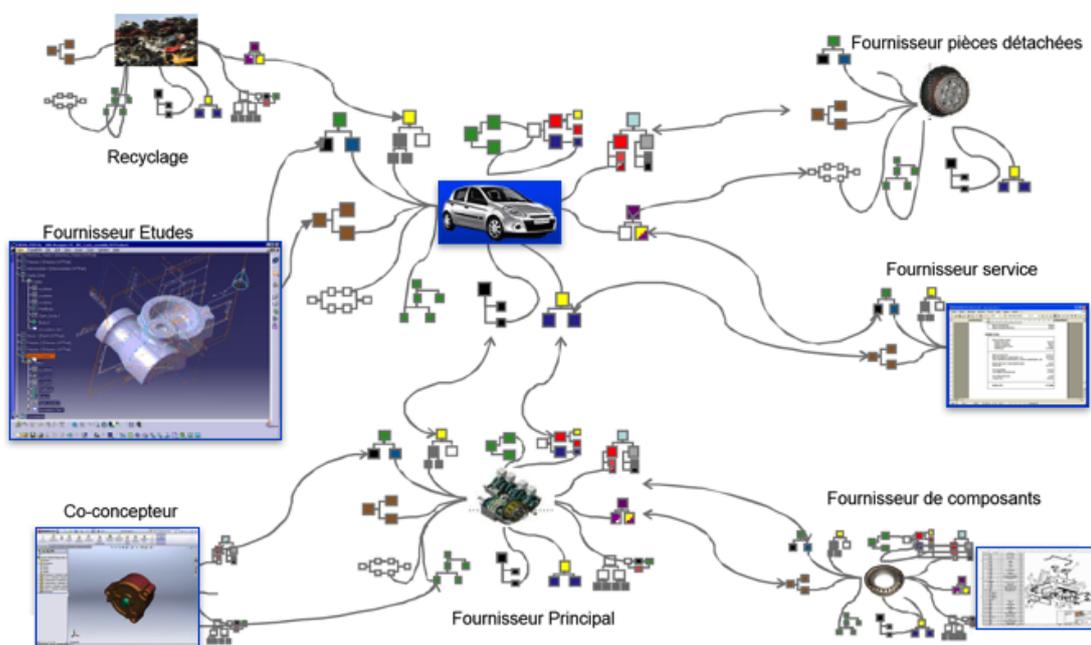


FIGURE 1.2 : Les objets PLM sont hétérogènes, complexes et interconnectés (Terzi et al. 2010).

La paradigme Product Lifecycle Managent (PLM) a émergé pour répondre à ce besoin. Évolué à partir de l'approche de gestion des données produit (*Product Data Management (PDM)*), les solutions PLM visent à fournir un référentiel commun de données entre différents acteurs de différentes équipes qui sont géographiquement distribuées (Figure 1.3). Elles permettent de gérer toutes les données et informations du produit d'une manière structurée et navigable, en assurant la traçabilité nécessaire pour une efficace réutilisation des données de produit (Assouroko et al. 2011). Aujourd'hui, les systèmes PLM vont devoir gérer des volumes de données de plus en plus massifs et hétérogènes.

1.1.2 Augmentation de données complexes et hétérogènes

Nous vivons dans un monde numérique où toutes nos activités sont liées aux données. Une énorme quantité de données sont produites et diffusées par l'humain, dans les organisations publiques ou privées, et à travers internet. Selon le rapport de l'International Data Corporation (IDC), "la quantité de données est doublée de taille tous les deux ans, et d'ici à 2020, les

données que nous créons et copions chaque année atteindront 44 zettabytes ou 44 milliards de giga-octets" (IDC-EMC 2014). Le volume de données croît à une vitesse exponentielle. Les données rendues disponibles dans les systèmes d'information actuels sont des plus en plus extrêmement volumineuses.

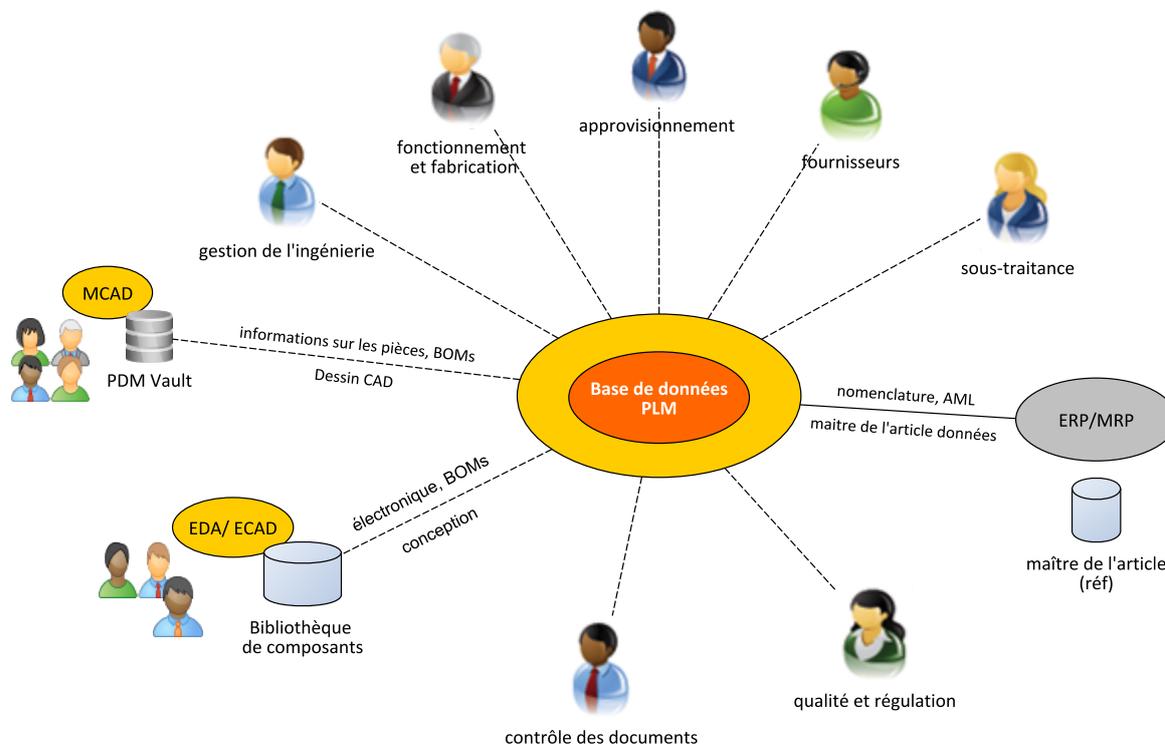


FIGURE 1.3 : Les différentes partie prenantes couverts par les solutions PLM³.

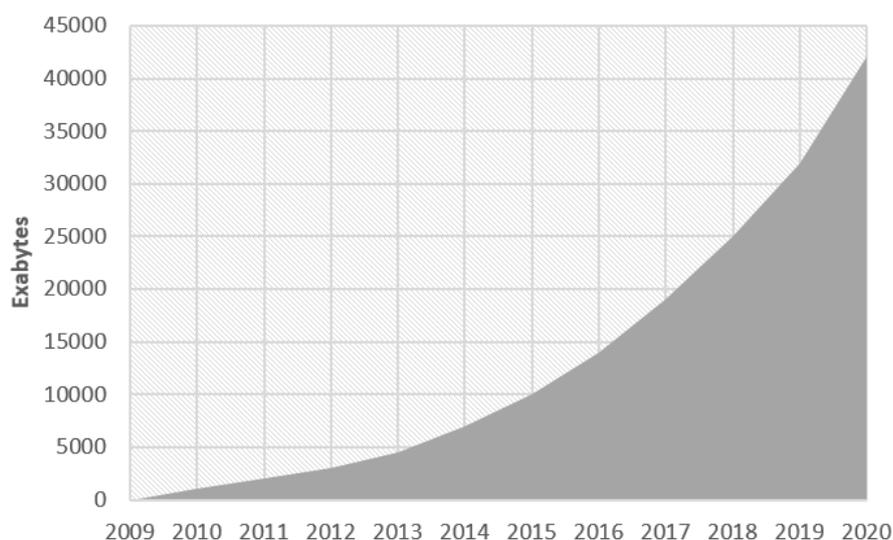


FIGURE 1.4 : Croissance de l'univers numérique entre 2010 à la prévision de 2020 (figure délivrée de (Gantz et Reinsel 2012)).

Les outils de gestion de données doivent faire face non seulement au grand volume de données mais aussi à la complexité et l'hétérogénéité de données. Les sources de données, notam-

3. <https://www.arenasolutions.com/products/plm/>

ment celles venant d'internet, sont bien dépendantes et interconnectées. Des grands nombres de relations complexes entre données doivent être gérées pour découvrir des informations pertinentes et utiles. Les données sont aussi hétérogènes car elles peuvent venir des différentes bases de données qui n'ont pas une structuration identique de données. Les données peuvent exister à des formats et des degrés de structuration variés.

Nous entrons dans l'ère des "**big data**". Cependant, on a tendance à qualifier les systèmes de complexes alors qu'en réalité, ils sont compliqués. Selon les définitions du Larousse, on peut différencier ces deux notions comme suit :

- **Compliqué** : "qui est difficile à comprendre ou à exécuter par suite du grand nombre et de la diversité des éléments composants ou des opérations nécessaires, etc." (Giraudin 2007).
- **Complexe** : "qui contient plusieurs parties ou plusieurs éléments combinés d'une manière qui n'est pas immédiatement claire pour l'esprit, etc." (Giraudin 2007).

D'après Giraudin (Giraudin 2007), "Ce n'est pas le volume en giga, téra, ou péta octets d'une base de données qui la rend complexe, mais sa complexité structurelle, les interrelations dynamiques entre ses objets, la gestion temporelle et la réactivité aux événements, le maintien de l'intégrité des données quelles que soient les transactions et les pannes matérielles ou logicielles naturellement toujours imprévisibles, mais dont les risques et les réactions doivent avoir été évalués". Simplement dit, les données peuvent être considérées comme complexes quand :

- Elles sont **représentées par différents types** (texte, image, sons, etc.)
- Elles sont **multi-structurées** (structurée, semi-structurée, structurée)
- Elles **proviennent de sources hétérogènes** (base de données, Web, etc.)
- L'information qu'elles représentent est décrit selon **plusieurs points de vue**
- Elles sont **évolutives** dans le temps

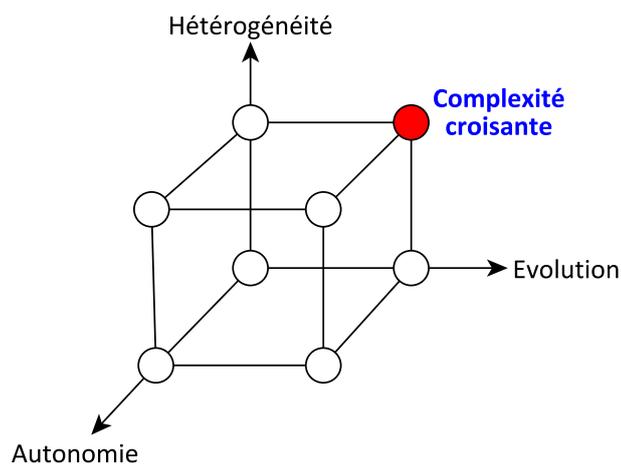


FIGURE 1.5 : Les facteurs de complexité (Giraudin 2007).

La Figure 1.5 illustre trois facteurs qui rend la complexité du système d'information : l'**hétérogénéité**, l'**autonomie**, l'**évolution**.

Dans ce manuscrit, par **hétérogénéité**, nous entendons les données qui sont représentées par différentes modèles selon des points de vue, niveaux d'abstraction (conceptuel, physique, etc.), catégories d'usage (gestionnaires, utilisateurs, etc.), domaine (étude, production, finance, etc.), types

de données (texte, images, sons, etc.) ou niveaux de structuration (structurée, semi-structurée, non-structurée), etc. Un système est autonome s'il est déconnecté des autres systèmes. Mais, un SI n'est jamais isolé, il s'insère dans son environnement avec un autre système avec lequel il a des échanges via des interfaces, des dispositifs techniques. Plus un SI est complexe, plus ses relations avec les autres sont fortes. Le SI n'est pas fixé dans le temps, mais, il évolue. L'évolution est un aspect important du SI. Le modèle du SI doit donc éventuellement évoluer, s'adapter. Dans notre travail, nous considérons uniquement les évolutions de la base de données par de nouvelles données générées par les activités quotidiennes des acteurs du SI.

1.1.3 Multi-utilisation de données

Face à la complexité croissante de données, avec une multiplication des volumes stockés sur des plates-formes disparates et la nécessité de partager des informations avec des rôles techniques comme non techniques, la collaboration devient capitale.

Dans le monde des entreprises industrielles, le développement des produits complexes demande la participation des acteurs venant de différents domaines. Bien que chaque acteur effectue ses propres activités, il partage des besoins communs d'utiliser des données. Par exemple, les ingénieurs, les fabricants et les fournisseurs travaillent et échangent autour de la nomenclature du produit (*Bill of Materials* (BOM) : liste de pièces et des sous-ensembles qui composent un produit). Dans le domaine académique, les nouveaux projets doivent impliquer de nombreux chercheurs avec des compétences variées pour assurer la qualité des résultats de recherche. Pour éviter les redondances et dysfonctionnements des flux de données, ces utilisateurs devraient travailler ensemble sur un référentiel commun partagé.

Définition de métier

Derrière le terme de multi-utilisation, nous faisons référence à la notion de métier. Selon Paviot (Paviot 2010), "le métier regroupe l'ensemble des savoirs et savoir-faire mobilisés par un opérateur pour mener à bien les activités relatives à un domaine identifié". Par exemple, le métier *conception/ fabrication* regroupe l'ensemble des savoirs et savoir-faire mobilisés lors d'activités liées à la *conception/ fabrication* des produits industriels. Suite à cette définition, nous utilisons le mot "**acteur métier**" pour appeler la personne qui s'occupe d'un métier spécifique comme les chercheurs en Neuroimagerie etc. Comme les données sont au coeur d'un nombre important d'initiatives au sein de l'entreprise, elles intéressent de plus en plus d'utilisateurs métiers. Dans le contexte des données complexes, évoluées et hétérogènes, s'assurer que toutes ces personnes, les informations et les processus soient dynamiquement alignés représente un défi. Le SI doit maintenant rationaliser les différents environnements techniques dans le but de créer une vue commune des principales ressources de données qui sont partagées et accessibles aux différents types d'acteurs métiers.

1.2 PROBLÉMATIQUE

L'accès aux données (ou la *recherche de données*), par définition est le processus d'extraction des informations à partir d'une base de données en utilisant des requêtes, pour répondre à une question spécifique. Une requête est souvent exprimée dans un format prédéfini par un

langage de requête, Structured Query Language (*SQL*) par exemple. L'accès aux données est une fonction indispensable pour tout système d'information. Cependant, cette fonction n'est jamais si facile, mais elle représente toujours un goulot majeur d'étranglement pour toutes les organisations (Soylu et al. 2013). Dans l'environnement de données complexes, hétérogènes et de multi-utilisation, fournir à tous les utilisateurs un accès facile et simple aux données devient plus difficile à cause de deux raisons :

- **La grande quantité d'interrelation entre données** : Pour formuler une requête complexe qui contient plusieurs jonctions (les *requêtes ad-hocs* ou *conjonctives*), l'utilisateur doit connaître l'organisation des données, c'est-à-dire la façon dont les données sont organisées et stockées dans la base de données. Par exemple, pour interroger une base de données relationnelle, l'utilisateur doit savoir quels sont les tableaux contenant les données qu'il souhaite et les relations entre ces tableaux. Quand les données sont volumineuses et complexes, ce n'est pas facile d'avoir une compréhension approfondie sur toutes les dépendances et interrelations entre données, même pour les techniciens du SI. Cette compréhension n'est pas forcément liée au savoir et savoir-faire du domaine et il est donc, très rare que les utilisateurs finaux possèdent les compétences suffisantes.
- **Différents points de vue des utilisateurs** : Dans l'environnement de multi-utilisation de données, chaque utilisateur introduit son propre point de vue quand il ajoute des nouvelles données et informations techniques. Les données peuvent être nommées de manières très différentes et les provenances de données ne sont pas suffisamment fournies. Par conséquent, ces données sont difficilement interprétées et accessibles par les autres acteurs et elles restent donc inexploitées ou non exploitées au maximum pour le partage et la réutilisation.

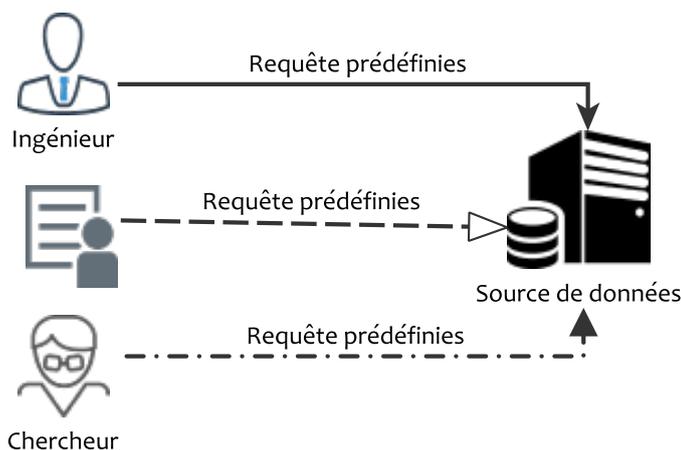


FIGURE 1.6 : L'accès aux données avec les requêtes prédéfinies (Giese et al. 2013)

Par conséquent, les acteurs métiers (*domaine-experts, utilisateurs finaux*) doivent :

- Soit utiliser des requêtes prédéfinies intégrées dans des applications.
- Soit communiquer leurs besoins d'information à un technicien (IT-expert). Ce dernier va ensuite transformer ces besoins (en langage naturel) à une (des) requête(s) exprimée(s) formellement en langage de requête.

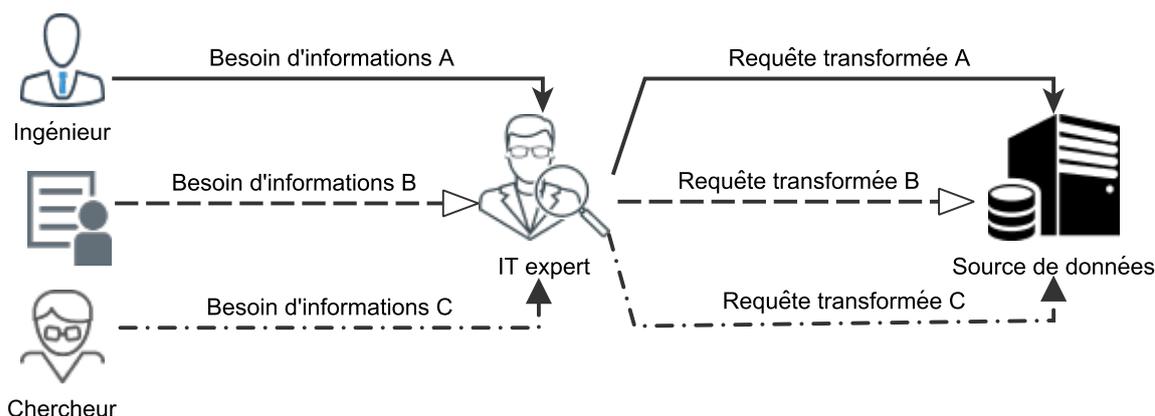


FIGURE 1.7 : L'accès aux données avec les requêtes complexes (Giese et al. 2013).

Le premier scénario (la Figure 1.6) est assez limitatif, car il est impossible d'énumérer tous les cas possibles au préalable, tandis que le deuxième scénario (la Figure 1.7) nécessite un délai important, car il dépend de la disponibilité des IT-experts. Ce délai peut être des jours ou des semaines à partir du moment où la requête utilisateur est créée jusqu'au moment où les utilisateurs reçoivent les réponses. De même, les utilisateurs ne sont pas informés des évolutions faites par leurs collègues, notamment, les ajouts de nouvelles données qui pourront les intéresser dans leurs traitements.

Dans le contexte de multi-utilisation, quand les points de vue des utilisateurs sont hétérogènes, c'est extrêmement difficile, voire même impossible pour les techniciens à traduire les besoins d'information de tous les différents types d'acteurs métiers. En fait, les SI actuels ne permettent pas une utilisation aisée par l'ensemble des divers acteurs métiers. Ce constat nous amène à poser la problématique suivante :

Problématique :

Les données gérées par les systèmes d'information actuels sont devenues de plus en plus complexes et hétérogènes. Elles sont difficilement accessibles par les acteurs métiers, en particulier par les utilisateurs n'ayant pas suffisamment de compétences techniques. Comment fournir à ces différents types d'utilisateurs un accès et un partage simple de l'ensemble de données et visualiser les informations et les connaissances issues de données ?

1.3 QUESTIONS DE RECHERCHE

Pour améliorer l'utilisation des données (en considérant une exploitation maximale), les acteurs métiers doivent avoir la capacité d'accéder eux-mêmes aux données et aux informations pratiques. Pour ce faire, il est nécessaire de leur fournir des moyens d'interprétation de ces données. Une représentation commune facilement compréhensible de la sémantique des données doit être partagée et accessible aux différents types d'acteurs métiers. La première question de recherche est donc :

Question 01 :

Comment permettre aux différents acteurs métiers de comprendre la structuration des données afin de trouver eux-mêmes les informations pertinentes ?

Les données doivent être représentées sémantiquement et au niveau conceptuel pour que la représentation soit accessible aux acteurs métiers de plusieurs disciplines. Une fois que les utilisateurs ont eu une représentation, il est nécessaire de leur proposer une démarche qui facilite le processus de formulation des requêtes. La deuxième question de recherche est :

Question 02 :

Comment faciliter le processus de formulation de requêtes selon différents types d'acteurs métiers ?

Une interface de requête orientée utilisateur pourrait répondre à cette question. Cette interface pourrait implémenter plusieurs mises en page. Chaque mise en page est dédiée à un groupe d'utilisateurs en fonction du niveau des compétences techniques. De plus, la formulation de requête peut être facilitée grâce aux avantages de la visualisation graphique, intuitive et des mécanismes d'interaction.

1.4 HYPOTHÈSE DE RECHERCHE

Les difficultés d'accès aux données des utilisateurs finaux viennent du fait que la manière dont les données sont structurées et stockées dans les bases de données ne tient pas en compte de l'hétérogénéité des compétences techniques et des domaines des utilisateurs. En fait, il existe une nette distinction entre les connaissances techniques nécessaires pour extraire des données et celles que les acteurs métiers ont pour interpréter, traiter et utiliser les données (Soylu et al. 2013).

Bien que la conception initiale de la base de données ait pris en compte la logique du domaine (en utilisant le *modèle entité-association* par exemple), il est de pratique courante de modifier cette conception afin de l'adapter à des besoins techniques spécifiques. Par conséquent, la conception finale est souvent une forme qui diverge de la structure conceptuelle originale.

Donc, afin de permettre à différents acteurs métiers d'accéder aux données, nous pouvons leur fournir une représentation plus compréhensive et plus déclarative de données. Cette représentation doit être au niveau sémantique, conceptuel et intégrer des notions du domaine qui sont proches de celles des acteurs métiers. Cette demande peut être assurée en utilisant l'ontologie, le pivot du Web Sémantique et l'Ingénierie des Connaissances. Le processus de formulation des requêtes peut également être facilité en profitant des avantages des outils de représentation graphique : **visuelle, dynamique et interactive**.

L'hypothèse de notre recherche est la suivante :

Nous proposons d'utiliser les concepts de l'ontologie, pivot de l'Ingénierie des Connaissances et du Web Sémantique, pour représenter les sources de données complexes d'une manière compréhensible, sémantique, déclarative et orientée acteurs métiers. Par les ontologies, il sera possible de mapper les points de vue métiers hétérogènes avec les données dans les bases de données pour

permettre une multi-utilisation aisée de données. Un système idéal de requête basé sur l'ontologie et les outils de visualisation dynamique peuvent aider les acteurs métiers à formuler eux-mêmes des requêtes complexes afin d'obtenir des données et informations techniques qu'ils souhaitent. Ce type d'environnement peut également fournir un moyen de partage sémantique des données à travers une visualisation dynamique des relations entre les données.

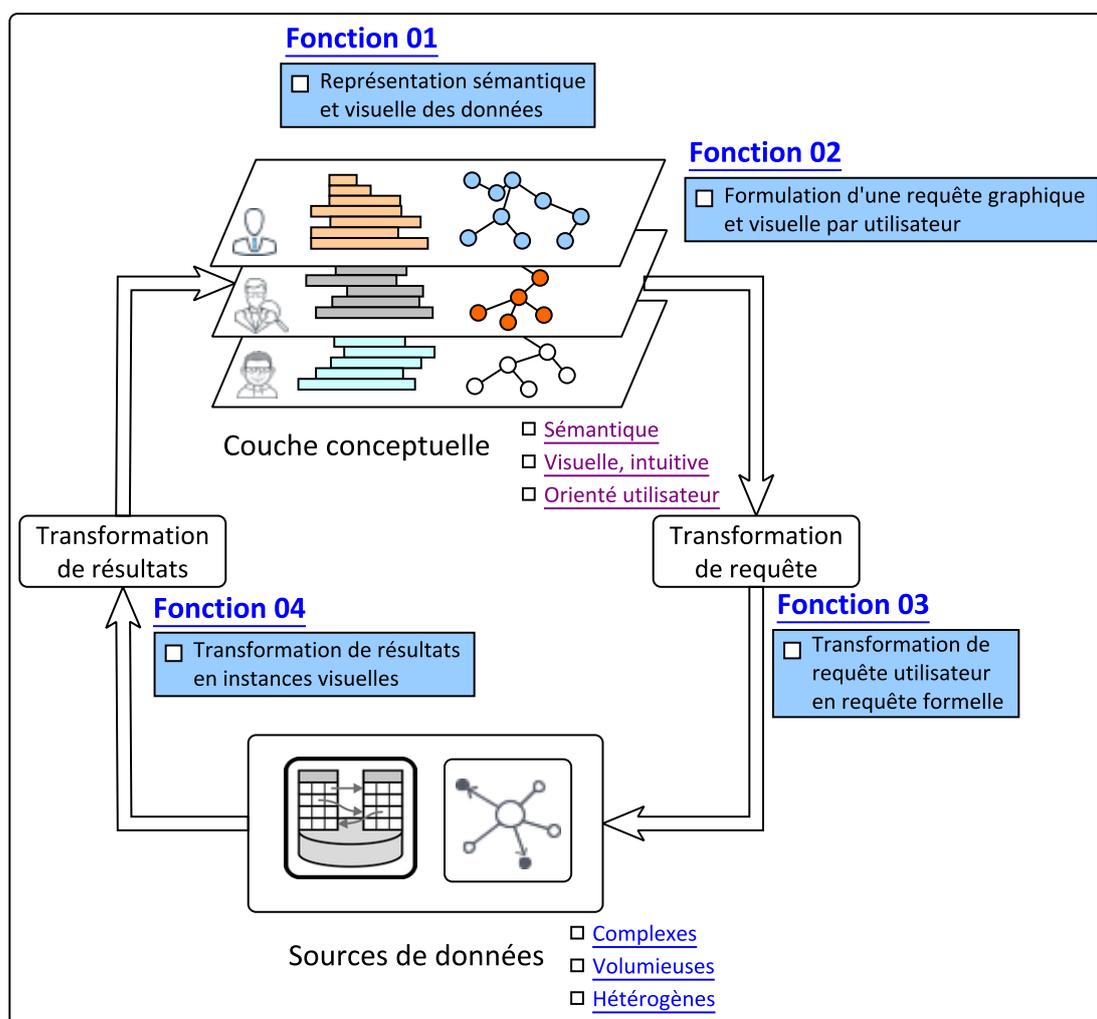


FIGURE 1.8 : La démarche idéale de la méthodologie proposée (hypothèse de recherche - accès sémantique aux données).

L'infrastructure du système de requête idéal basé sur la représentation sémantique des données est illustrée à la Figure 1.8. Cette démarche d'interrogation de données se constitue de quatre fonctions principales :

- **Représentation sémantique des données.**
- **Formulation d'une requête graphique et visuelle par utilisateur.**
- **Transformation de requête utilisateur en requête formelle.**
- **Transformation de résultats en instances visuelles.**

Tous les éléments bibliographiques ainsi que les propositions de recherche que nous allons

présenter dans les chapitres suivants seront présentés et structurés au tour de ces quatre fonctions.

1.5 DOMAINE D'APPLICATION, PLM POUR LA NEUROIMAGERIE

1.5.1 Les solutions PLM

1.5.1.1 Historique

Les données d'ingénierie ("*Engineering Data*") sont les données relatives à la conception, à l'approvisionnement, à la fabrication, à l'essai et à l'inspection d'un article ou d'une structure. Les premiers logiciels de gestion des données d'ingénierie (*Engineering Data Management Software* (EDMS)) ont géré des données qui étaient assez simples : les documents de conceptions, les listes de pièces et les spécifications. L'objectif principal de ces logiciels était de permettre aux utilisateurs d'organiser leurs données de produits, d'appliquer des règles pour identifier des items (*articles*) et de contrôler des révisions.

La première génération du SGDT dans les années 80s a permis seulement de collaborer entre utilisateurs dans un seul bureau d'études (Figure 1.9). Dans les années 90, afin d'améliorer la productivité, les logiciels de SGDT ont évolué par fournir des outils permettant à collaborer entre différents secteurs de l'entreprise dans différentes phases du cycle de vie du produit (Figure 1.10).

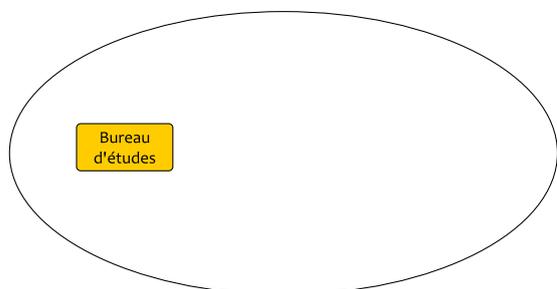


FIGURE 1.9 : La première version du SGDT (délivré de (Debaecker 2004)).

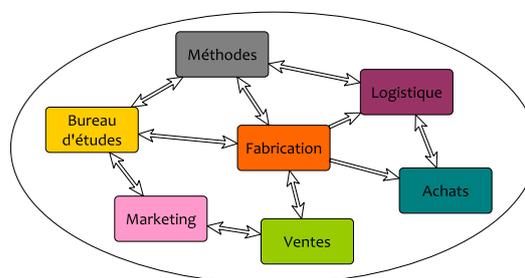


FIGURE 1.10 : Evolution du SGDT dans les années 90s (délivré de (Debaecker 2004)).

Dans les années 2000s, la quantité, la variété et la complexité des produits industriels ont augmenté rapidement. La conception du produit demande une multiplication et une spécialisation des acteurs qui sont répartis dans de nombreuses structures géographiquement dispersées. Comme conséquence, les informations concernant le produit sont aussi dispersées. Dans cet environnement hétérogène et distribué, ce n'est pas facile d'assurer que toutes les personnes, les informations et les processus soient dynamiquement alignés (Bricogne-Cuignières 2015, Terzi et al. 2010).

Pour résoudre ce défi, un nouveau paradigme est apparu comme une extension des SGDT en cours : La gestion des informations techniques de l'entreprise étendue PLM (*Production Lifecycle Management*). Cette approche est plus tôt centrée sur le cycle vie du produit avec l'objectif de fournir un environnement de travail collaboratif et un référentiel commun entre

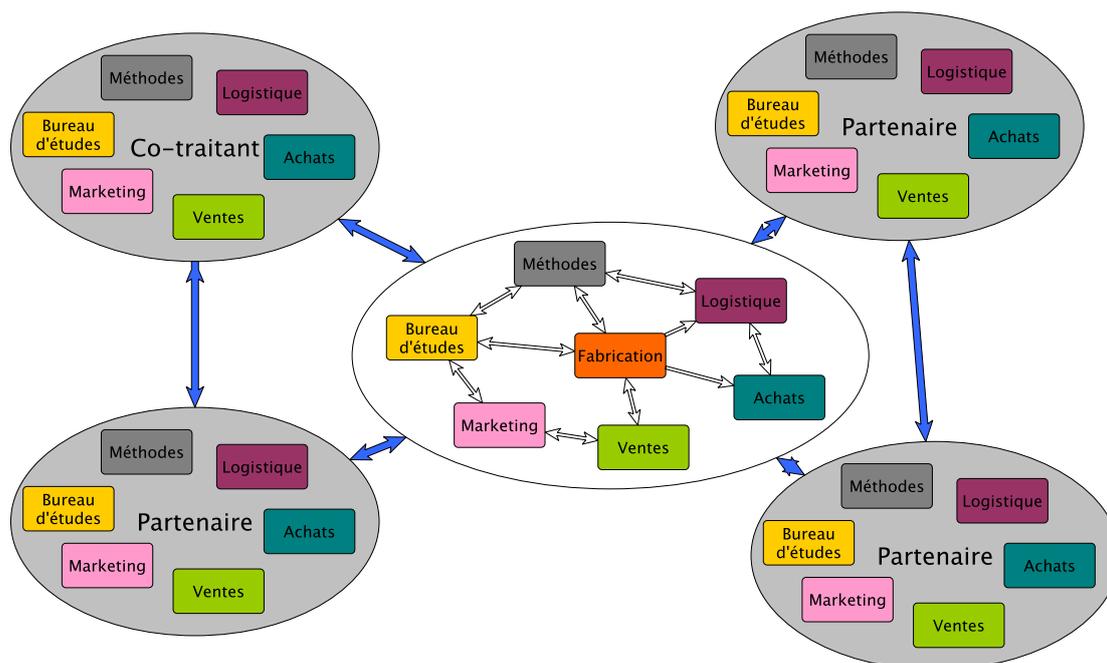


FIGURE 1.11 : La gestion des informations techniques de l'entreprise étendue (délivré de (Debaecker 2004)).

différents acteurs non seulement au sein d'une entreprise, mais aussi entre différentes entreprises qui sont géographiquement distribuées et dans un variété de domaine (Figure 1.11).

1.5.1.2 Définitions du PLM

De nombreuses définitions du PLM sont proposés dans la littérature. Paviot (Paviot 2010) et Fortineau (Fortineau 2013) ont présenté une liste exhaustive des définitions fournies par les éditeurs du PLM, les entreprises industrielles et la communauté scientifique.

Les éditeurs du PLM

"Le PLM est une stratégie d'entreprise qui aide les entreprises à partager les données produit, à appliquer des procédés communs et à capitaliser les informations de l'entreprise pour le développement de produits, de la conception à la mise au rebut, et dans tous les segments de l'entreprise étendue. En incluant tous les acteurs (collaborateurs de l'entreprise, partenaires, fournisseurs, équipementiers et clients), la gestion du cycle de vie du produit permet à ce réseau de fonctionner en tant qu'entité unique de la conception à la maintenance, en passant par la fabrication" _Dassault Système⁴_

Dans cette définition, le PLM est considéré comme une stratégie à long terme permettant d'améliorer le partage de données produit entre tous les acteurs du début jusqu'à la fin du cycle vie du produit comprenant la conception, la fabrication et la maintenance.

"Le PLM permet à l'entreprise de prendre des décisions cohérentes et pilotées par les systèmes d'information, à chaque phase du cycle de vie du produit. Les solutions PLM mettent en place

4. <http://www.3ds.com/>

une plateforme intégrée pour : Optimiser les relations au travers de toutes les phases du cycle de vie et de toutes les organisations impliquées; Maximiser dans le temps la valeur du portefeuille de produits; Mettre en place un système unique pour l'enregistrement et la diffusion des données numériques". **Siemens PLM**⁵

"Les solutions PLM permettent aux entreprises d'accélérer l'innovation et de maximiser la rentabilité des produits par la gestion de l'information, des processus et des décisions concernant le produit, tout au long de leur cycle de vie". **PTC**⁶

Un avantage à base du PLM décrit dans les définitions proposées par Siemens PLM et PTC est de fournir des informations nécessaires permettant de prendre des bonnes décisions dans toutes les phases du cycle vie du produit. Ces décisions réduisent le temps de développement, maximisent la rentabilité des produits et accélèrent l'innovation de l'entreprise.

La communauté scientifique

"Le PLM est l'activité qui consiste à gérer les produits de l'entreprise, tout au long de leur cycle de vie, de la manière la plus efficiente. Le PLM est holistique : son champ couvre de nombreux éléments comme les processus, les données, les hommes et les applications. Cela le distingue d'activités atomiques qui s'attachent à un point particulier, par exemple le **Product Data Management (PDM)**. Selon l'auteur, le PLM peut être vu comme un paradigme de rassemblement : il fédère de nombreux processus, disciplines, fonctions et applications jusqu'alors hétérogènes et indépendantes, qui adressent un même produit avec différents vocabulaires, règles métiers, cultures ou langues". **(Stark 2005)**

La définition de Stark (Stark 2005) et de PTC⁶ ci-dessus mettent sur la valeur de la gestion des processus tout au long du cycle vie de produit. Les systèmes PDM ou PLM prennent en charge la gestion des processus en définissant les étapes du processus liées au développement, à la distribution et à l'utilisation des données produit. Le processus est défini sous la forme d'étapes de processus et de niveaux de diffusion ou de promotion spécifiés que les données doivent atteindre (Figure 1.12).

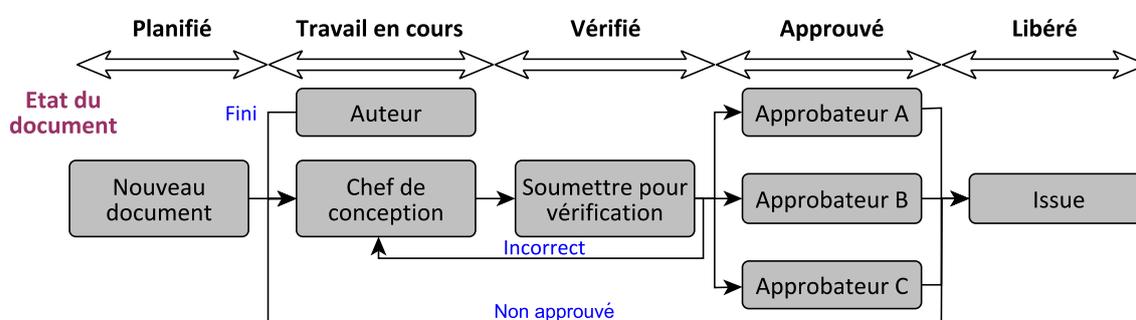


FIGURE 1.12 : Processus de développement d'un nouveau document⁷.

5. http://www.plm.automation.siemens.com/en_us/

6. <http://www.ptc.com/>

7. http://leachristiansen.dk/u2_plm.html

"Product Lifecycle Management est défini comme une approche stratégique pour la gestion et l'utilisation du capital intellectuel d'entreprise. Le PLM assure la gestion de toutes les informations sur les produits tout au long de leur cycle de vie : de la conceptualisation à des opérations/ élimination". _IJPLM⁸_

"Une approche stratégique qui met oeuvre un ensemble cohérent de pratiques permettant de supporter la création collaborative ainsi que l'organisation, la diffusion et l'utilisation des informations relatives à la définition du produit au travers de l'entreprise étendue, de la conception à la fin de vie, et d'intégrer les hommes, les processus, les systèmes d'organisation et d'information". _CIMData⁹_

Comme dans la définition de Dassault Système, IJPLM et CIMData définissent le PLM comme une approche stratégique permettant de gérer efficacement (organiser, diffuser, utiliser) des informations relatives à la définition du produit (*capital intellectuel d'entreprise*). Dans sa définition, CIMData a mentionné la notion "**Entreprise étendue**", c'est-à-dire, les solutions PLM permettent d'intégrer les **hommes**, les **processus**, les **systèmes d'information** de non seulement une seule entreprise, mais aussi l'ensemble d'entreprises et des acteurs économiques associées.

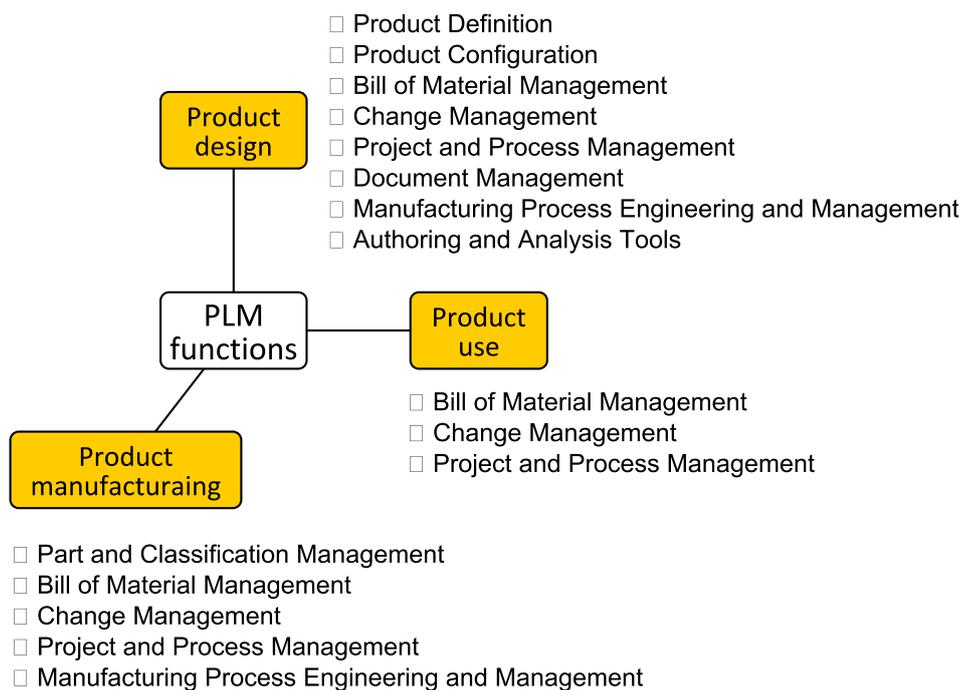


FIGURE 1.13 : Fonctionnalités du PLM dans chaque phase du cycle vie du produit (John Stark 2015).

1.5.1.3 Fonctionnalités du PLM

Selon les définitions présentées ci-dessus, les solutions PLM fournissent les fonctions suivantes dans chaque phase du cycle vie du produit (John Stark 2015) : Conception du pro-

8. <http://www.inderscience.com/jhome.php?jcode=ijplm>

9. <http://www.cimdata.com/en/>

duit (Product design), Fabrication du produit (Product manufacturing), Utilisation du produit (Product usage) (Figure 1.13).

1.5.1.4 Adaptation du PLM dans les domaines de recherche

Les solutions PLM peuvent être adaptées aux différents domaines de recherche qui partagent des problématiques de gestion de données comme les entreprises industrielles. Cependant, les systèmes PLM actuels ne sont pas assez souples pour les chercheurs non-technicien. L'interface des systèmes de PLM existants n'a pas été conçue d'une manière conviviale et ne permet pas une navigation facile des données interconnectées (Allanic 2015). L'efficace exploitation de données nécessite une compréhension assez solide de la structure des sources de données.

Les projets de recherche demandent une collaboration forte de chercheurs à l'intérieur ou à l'extérieur d'une unité de recherche. Venant de disciplines variées, chaque chercheur a son propre niveau de compétences techniques. Assurer que les données sont accessibles par tous ces acteurs est donc difficile. De plus, chaque chercheur introduit un point de vue spécifique quand il ajoute des nouvelles données, bien que les solutions PLM permettent de "*fournir des bonnes informations au bon moment à la bonne personne*" (Terzi et al. 2010), il n'est pas facile pour les autres chercheurs de trouver des bonnes informations pour leurs besoins spécifiques, quand ils ne comprennent pas la sémantique de ces nouvelles données.

1.5.2 La Neuroimagerie

Le domaine de la Neuroimagerie étudie le fonctionnement du cerveau à l'aide de techniques issues de l'imagerie médicale. En particulier, l'usage de la technique IRM (*Imagerie par Résonance Magnétique*) est répandu pour l'étude fonctionnelle du cerveau. La Neuroimagerie est un domaine pluridisciplinaires, multi-méthodes et multiplateformes où les données à collecter, à gérer et à traiter sont très nombreuses, complexes et hétérogènes. Les récents projets de recherche demandent une collaboration de chercheurs multidisciplinaires, d'une part pour assurer la qualité des résultats, et d'autre part pour diminuer le coût d'acquisition, de traitement et de gestion de données. En ce qui concerne l'aspect volumétrique des données dans le domaine du BMI, il convient de mentionner qu'il y a cinq ans des études de recherche ont été limitées à quelques dizaines de sujets, tandis que les études en cours traitent actuellement une cohorte de milliers de sujets. La quantité de données peut facilement atteindre cinq fois le volume de données brutes.

1.5.3 PLM pour la Neuroimagerie

La communauté Neuroimagerie partage les mêmes problèmes que les entreprises industrielles il y a des décennies : les données à gérer sont à très grande multi-utilisation et à de multi-expertises, il faut monter vite en compétence sur la gestion de données qui sont progressivement plus complexes et plus hétérogènes. Les solutions PLM peuvent être utilisées pour assurer une gestion idéale de données BMI (Allanic 2015) à trois aspects :

- **Provenance** : Toutes les données reliées à une étude (de la première phrase de l'ac-

quisition jusqu'à la publication de résultats) sont capturées et gérées efficacement pour l'objectif de reproduction et réutilisation.

- **Hétérogénéité** : Les données à tous les formats et degrés de structuration sont gérées.
- **Flexibilité** : Le système est suffisamment flexible pour faire face aux évolutions de données et même le modèle de données utilisées pour organiser des données.

Initialement utilisées pour les entreprises industrielles, cependant, les solutions PLM ne sont pas suffisamment flexibles pour une multi-utilisation de données des acteurs métiers non-techniciens ou les personnes occasionnelles. En fait, dans les systèmes PLM actuels, les outils de navigation entre données sont encore limités (pas suffisamment intuitifs) (comme illustré dans la Figure 1.14) et les moteurs de recherche intégrés demandent toujours des compétences informatiques pré-acquises.

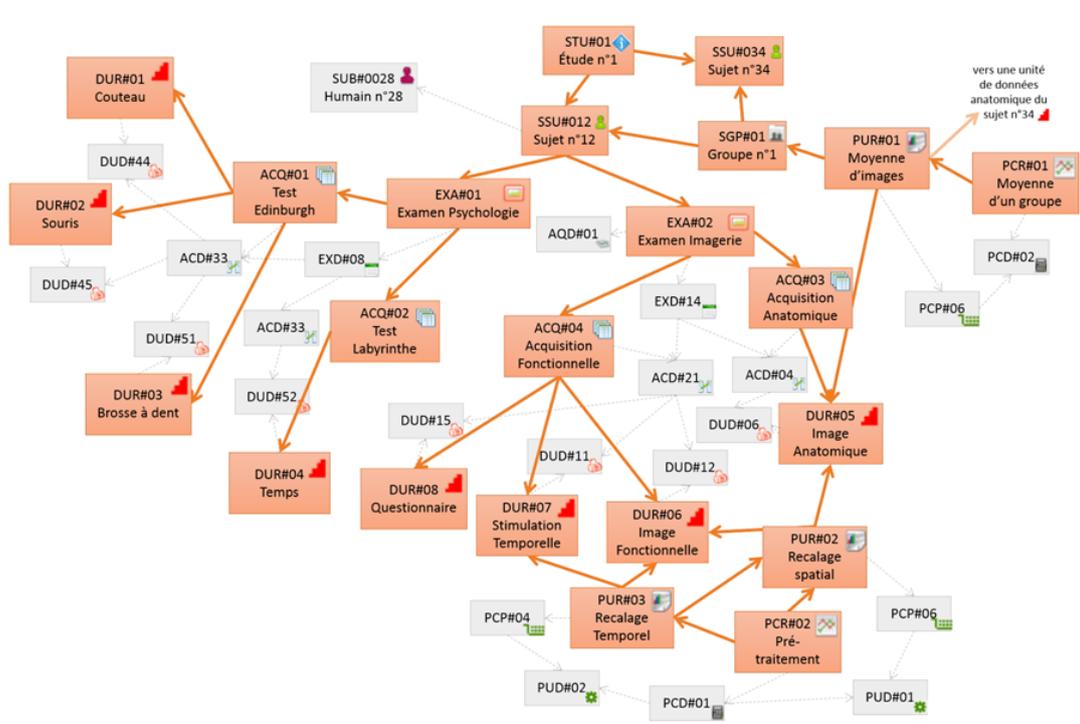


FIGURE 1.14 : Schéma exemple présentant une partie des données de deux sujets associés à l'étude dans le domaine Neuroimagerie (Allanic 2015).

La méthodologie proposée dans le cadre de ce manuscrit sera appliquée dans le domaine du PLM pour l'Imagerie Biomédicale présenté ci-dessus. Pour montrer la faisabilité, basé sur cette méthodologie, nous avons développé un système de requête nommé VAQUERO (VisuAlization and QUery based Ontology) qui permet aux différents chercheurs multi-métiers au laboratoire GIN (*Imaging Group Neurofunctional*, CEA CNRS Université de Bordeaux) de trouver des données et des informations pertinentes correspondantes à leurs besoins spécifiques.

1.6 CONCLUSION

Dans ce chapitre, nous avons présenté tout d'abord le contexte de l'étude et la problématique de recherche. Deux questions de recherche sont ensuite proposées :

1. *Comment permettre aux différents acteurs métiers de comprendre la structuration des données afin de trouver eux-mêmes les informations pertinentes ?*
2. *Comment faciliter le processus de formulation de requêtes des différents acteurs métiers ?*

Afin de répondre à ces questions, nous avons d'abord étudié les systèmes de données pour mettre en avant leurs limites en termes de multi-accessibilité et de gestion de l'évolution des données. Nous explorons par la suite, des voies de solution avec le web sémantique et les systèmes basées sur les ontologies comme le OBDA, etc. Enfin, nous détaillons le cadre de l'application de notre approche dans un PLM en Neuroimagerie.

ÉTAT DE L'ART

2

SOMMAIRE

2.1	DONNÉES ET BASES DE DONNÉES	37
2.1.1	Définitions	38
2.1.2	Typologie des bases de données des SI	38
2.1.3	Construction des bases de données	39
2.1.4	Les utilisateurs de BDDs	40
2.1.5	Limitations d'accès aux BDDs	42
2.2	INTERROGATION DES BASES DE DONNÉES	42
2.2.1	Les langages de requêtes	42
2.2.2	Simplification de l'interrogation les BDDs	42
2.3	REPRÉSENTATION DES CONNAISSANCES	46
2.3.1	Donnée, information et connaissance	46
2.3.2	Pourquoi la représentation des connaissances	47
2.3.3	Formes de représentation des connaissances	47
2.3.4	Ontologie	51
2.4	VISUALISATION	65
2.4.1	Visualisation des données	66
2.4.2	Visualisation des informations	67
2.4.3	Visualisation des connaissances	67
2.4.4	Visualisation des ontologies	68
2.5	LA CONSTRUCTION DE REQUÊTE SÉMANTIQUE VISUELLE	74
2.6	WEB SÉMANTIQUE ET INGÉNIERIE DES CONNAISSANCES	77
2.6.1	L'ontologie et le web sémantique	79
2.6.2	La recherche sémantique	79
2.6.3	L'intégration sémantique d'informations	80
2.6.4	Mapping entre BDDs et ontologies	82
2.7	CONCLUSION	87

Notre approche propose un lien entre les bases de données et sa représentation sémantique et visuelle. Ce chapitre est divisé en trois parties :

- Les notions fondamentales de bases de données, les méthodes et les technologies facilitant la recherche de données.
- Les techniques de représentation de données, l'ontologie, les technologies du Web Sémantique et les techniques de visualisation.
- Le mapping entre l'ontologie et les bases de données, la recherche de données à base d'ontologie (*Ontology-Based Data Access (OBDA)*).

2.1 DONNÉES ET BASES DE DONNÉES

Cette section présente des notions de base concernant les données, les bases de données et les techniques d'interrogation des BDDs (Figure 2.1).

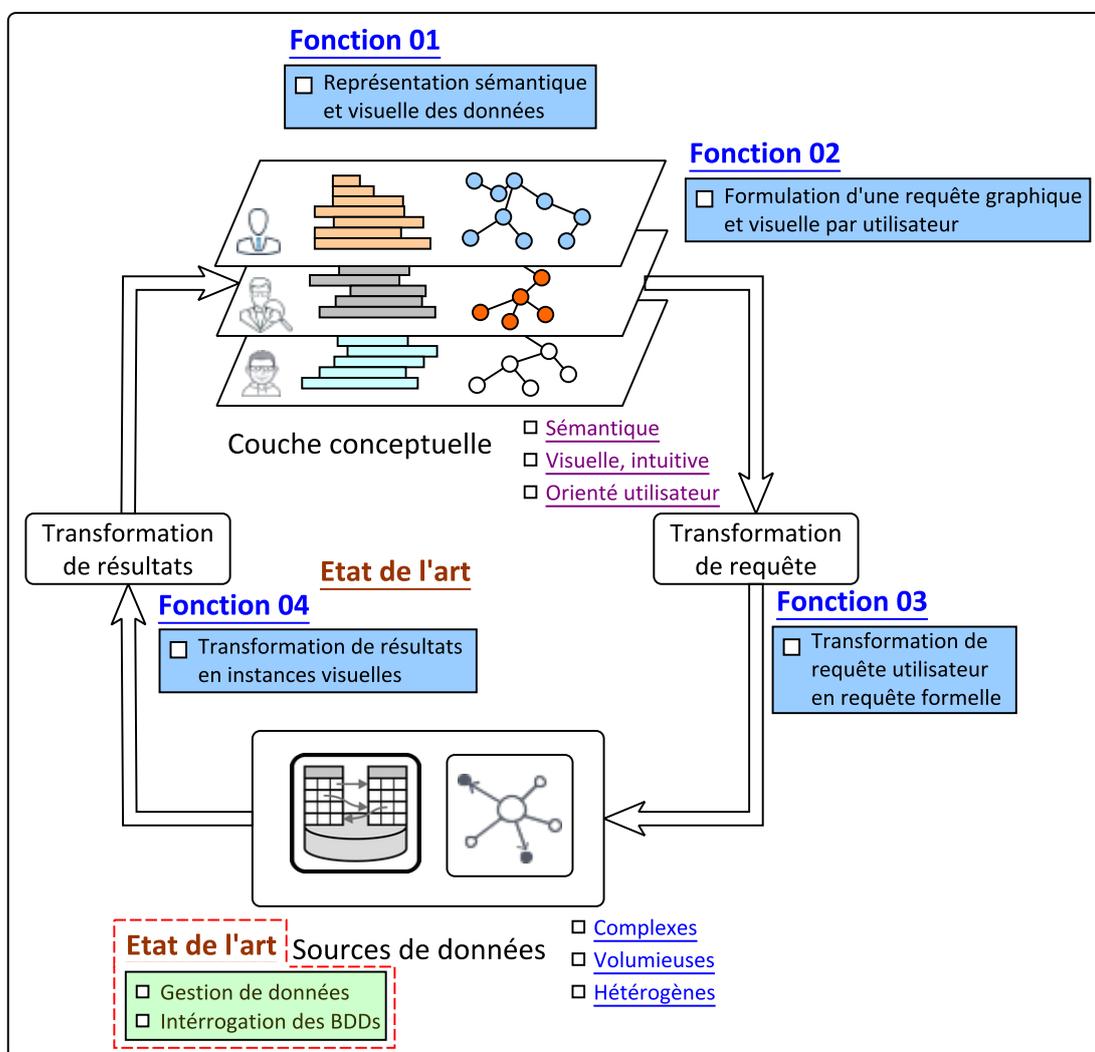


FIGURE 2.1 : L'état de l'art sur les bases de données et les méthodologies de l'interrogation des BDDs.

2.1.1 Définitions

Qu'est-ce qu'une donnée? Selon (Keller et Tergan 2005), on appelle donnée "une unité brute qui peut être un symbole ou un fait isolé et pas encore interprété, mis en contexte". Etant le résultat direct d'une mesure, "*une donnée peut être collectée par un outil de supervision, par une personne ou être déjà présente dans une base de données*"¹⁰. Au fil du temps, la **Base de Données** (BDD) et les systèmes de gestion de base de données (SGBD) sont apparus pour gérer plus efficacement des données, afin d'exploiter et de réutiliser les mêmes données dans différents contextes (Domingue et al. 2011).

Une base de données est une collection organisée de données permettant de stocker, d'organiser et de retrouver des données d'une manière efficace. Les données d'une BDD sont organisées et structurées selon des modèles de données. "*Un modèle de données est un ensemble de concepts de différents types représentant l'organisation de la réalité*" (Catarci et al. 1997). Une base de données est gérée par un système de gestion de base de données (SGBD).

2.1.2 Typologie des bases de données des SI

Les bases de données sont classifiées en fonction du nombre d'utilisateurs, du type de contenus, de leur objectif et surtout de la manière dont les données sont structurées. Selon le modèle de données (Figure 2.2) utilisé, les bases de données (ou SGBD) sont classifiées en deux catégories :

- **Les bases de données relationnelles (SQL)** : Les BDD relationnelles (BDDRs) utilisent le **Modèle De Données Relationnelles (MDDR)** introduit dans les années 1970 (Codd 1970) pour organiser des données. Le modèle relationnel offre un moyen très mathématiquement adapté à la structuration, à la conservation, et à l'utilisation des données. Basé sur ce modèle, la base de données relationnelle utilise strictement les relations (tables) pour stocker des données, regroupées selon leurs caractéristiques. Les BDDRs sont généralement utilisées par les entreprises et souvent pour les transactions qui nécessitent une grande précision. Elles sont généralement soumises à des contraintes relevant de ce qu'on appelle ACID (**A**tomicité, **C**ohérence, **I**solation, **D**urabilité) (Haerder et Reuter 1983). Cependant, ces contraintes rendent les bases de données plus lentes dans le cas où les données sont volumineuses. Ce type de modèle (MDDR) n'est pas adapté au contexte des données massives et distribuées.
- **Les bases de données non-relationnelles (Non-SQL)** : De nombreuses entreprises et organisations ont développé des systèmes spécifiques de stockage qui sont maintenant appelés base de données NoSQL. L'objectif principal de ces bases de données est la gestion de grands volumes de données, mais aussi des données qui sont géographiquement distribuées. Base de données NoSQL gèrent les données non structurées telles que les fichiers de traitement de texte, e-mail, le multimédia et les médias sociaux de manière efficace. Les bases de données NoSQL sont fondées sur la théorie de la CAP qui signifie la cohérence (*Consistency*), la disponibilité (*Availability*) et la tolérance au partitionnement (*Partition Tolerance*) (Gilbert et Lynch 2002). "Ces principes permettent d'écailler

10. <http://www.itilfrance.com/index.php?pc=pages/docs/itilv3-03/117-02.inc>

horizontalement des opérations à travers de nombreux serveurs, de reproduire, de distribuer, et d'ajouter dynamiquement de nouveaux attributs de données" (Sark 2013). Certains types populaires de base de données NoSQL sont organisées sous forme de paires valeur-clé, orientées colonnes ou document ou structurées selon des objets ou des graphes. Cependant, il n'y a pas de type de bases de données qui couvre toutes les exigences de l'entreprise, chaque entreprise doit choisir un modèle approprié pour un certain cas d'utilisation (Hecht et Jablonski 2011).

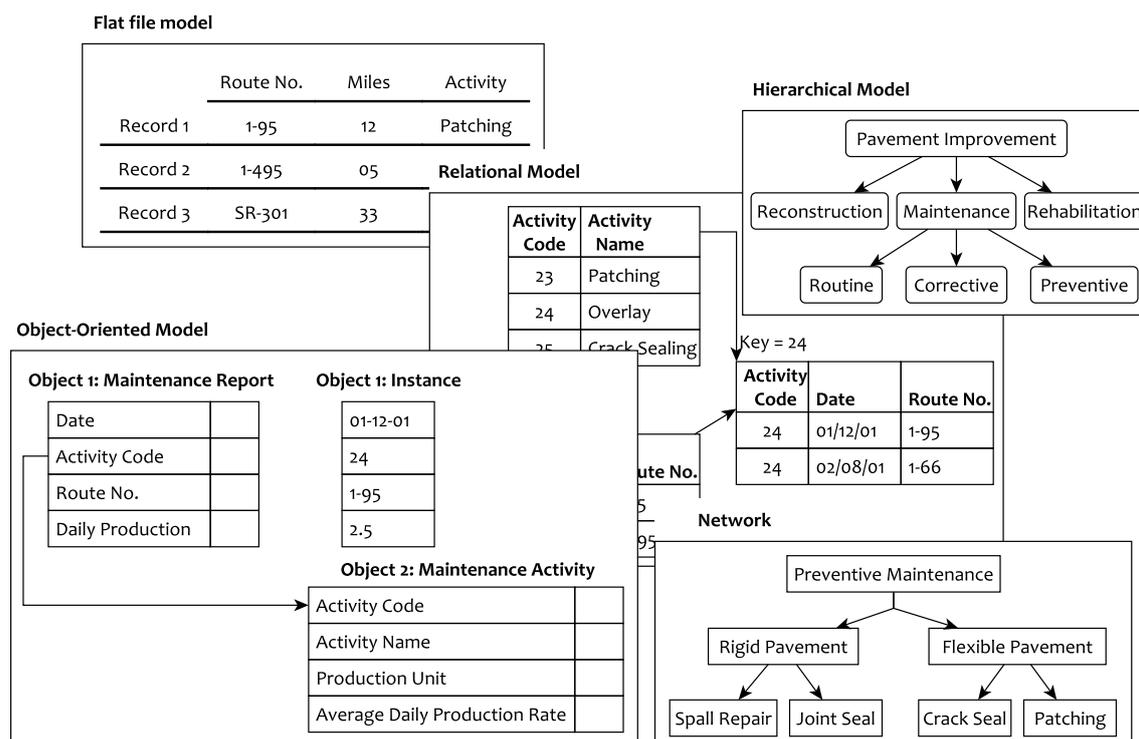


FIGURE 2.2 : Différents types du MDD ¹¹.

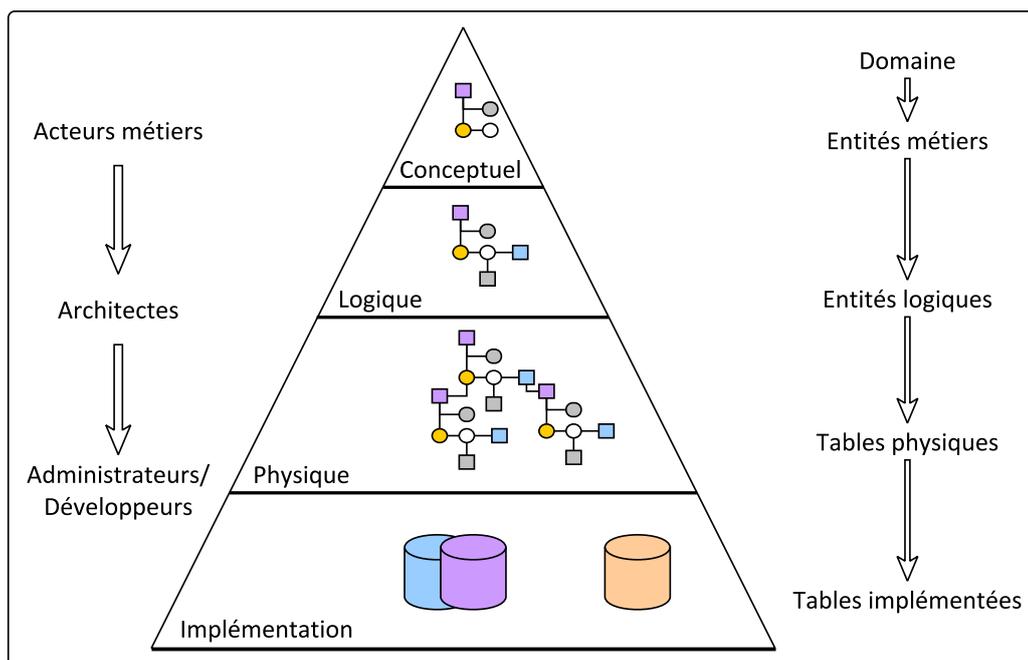
2.1.3 Construction des bases de données

Principalement, le processus standard de conception de BDD se compose en trois étapes (Figure 2.3) : **Construction du modèle conceptuel**, **Construction du modèle logique** et **Construction du modèle physique** (Spanos et al. 2012, Juan et Jones 2007).

- **Construction du Modèle Conceptuel (MC)** : La construction d'une BDD commence par l'identification des potentielles exigences fonctionnelles exprimées par les utilisateurs ciblés. Cette phase préliminaire s'ensuit par la construction du schéma conceptuel qui décrit, à un plus haut niveau, des données stockées dans la BDD ainsi que les contraintes sur ces données (Spanos et al. 2012). Afin de modéliser la sémantique des informations de façon compréhensible, le modèle conceptuel est exprimé sous forme d'Entité-Association (*Entity-Relationship* (ER)) (Chen 1976; 2002), Entité-Association-

11. <https://en.wikipedia.org/wiki/Database>

12. http://www.theregister.co.uk/2007/06/14/data_modelling_layers/

FIGURE 2.3 : Trois couches de modélisation des données ¹².

Entendue (EAE) (*Enhance Entity-Relationship (EER)*) ou en Langage de Modélisation Unifié (*Unified Modeling Language (UML)*).

- **Construction du Modèle Logique (ML)** : Ce type de modèle représente la structure abstraite d'un domaine de l'information. Il présente une représentation détaillée des données d'une entreprise, indépendamment de toute technologie particulière. Il est exprimé dans la langue des "affaires". Un modèle logique de données standardise les représentations des personnes, des lieux, des objets et des règles, des relations et des événements entre eux. Un ML vise à traduire le MC en tables utilisables par un SGBD. Dans la deuxième étape, le modèle conceptuel est transformé en modèle logique qui implémente la structure de données dans la BDD. Si le modèle conceptuel est indépendant des technologies utilisées par le SGBD, le modèle logique par contre, est exprimé en termes de modèles supportés par le SGBD tels que : *modèle relationnel (Codd 1970)*, *modèle hiérarchie*, *modèle réseau*, etc.
- **Construction du Modèle Physique (MP)** : Le modèle physique est utilisé directement pour implémenter la BDD. La sélection du modèle physique est dépendante notamment des exigences de la performance du système : vitesse, capacité, évolutivité, récupération, sécurité, etc. La phase de conception du modèle physique est invisible par les utilisateurs du système.

2.1.4 Les utilisateurs de BDDs

Les utilisateurs de BDD sont ceux qui ont accès aux bases de données. Nous distinguons différents types d'utilisateurs en fonction de leurs besoins et de la manière d'accéder à la base de données (Elmasri et Navathe 2010) (Figure 2.4) :

- **Les Concepteurs de la BDD** : leur objectif est d'identifier les données à stocker et de

choisir la structure la plus appropriée pour les représenter et les stocker. Ils doivent concevoir et implémenter les modules ainsi que les interfaces du SGBD.

• **Les Utilisateurs finaux :**

- * **Utilisateurs occasionnels :** Ils ont besoin d'accéder aux données et ils peuvent avoir divers besoins d'information.
- * **Utilisateurs paramétriques :** leur tâche consiste à mettre à jour la BDD grâce aux outils dédiés et ne demandent pas des compétences techniques pour les utiliser.
- * **Utilisateurs sophistiqués :** leur objectif est de comprendre les fonctionnalités du SGBD pour implémenter eux-mêmes des applications répondant aux différents problèmes.
- * **Utilisateurs autonomes :** Ils ont besoin d'éventuellement créer et de maintenir des BDD personnelles grâce aux "menu-based" ou "graph-based" qui sont des outils préprogrammés fournis par le SGBD.

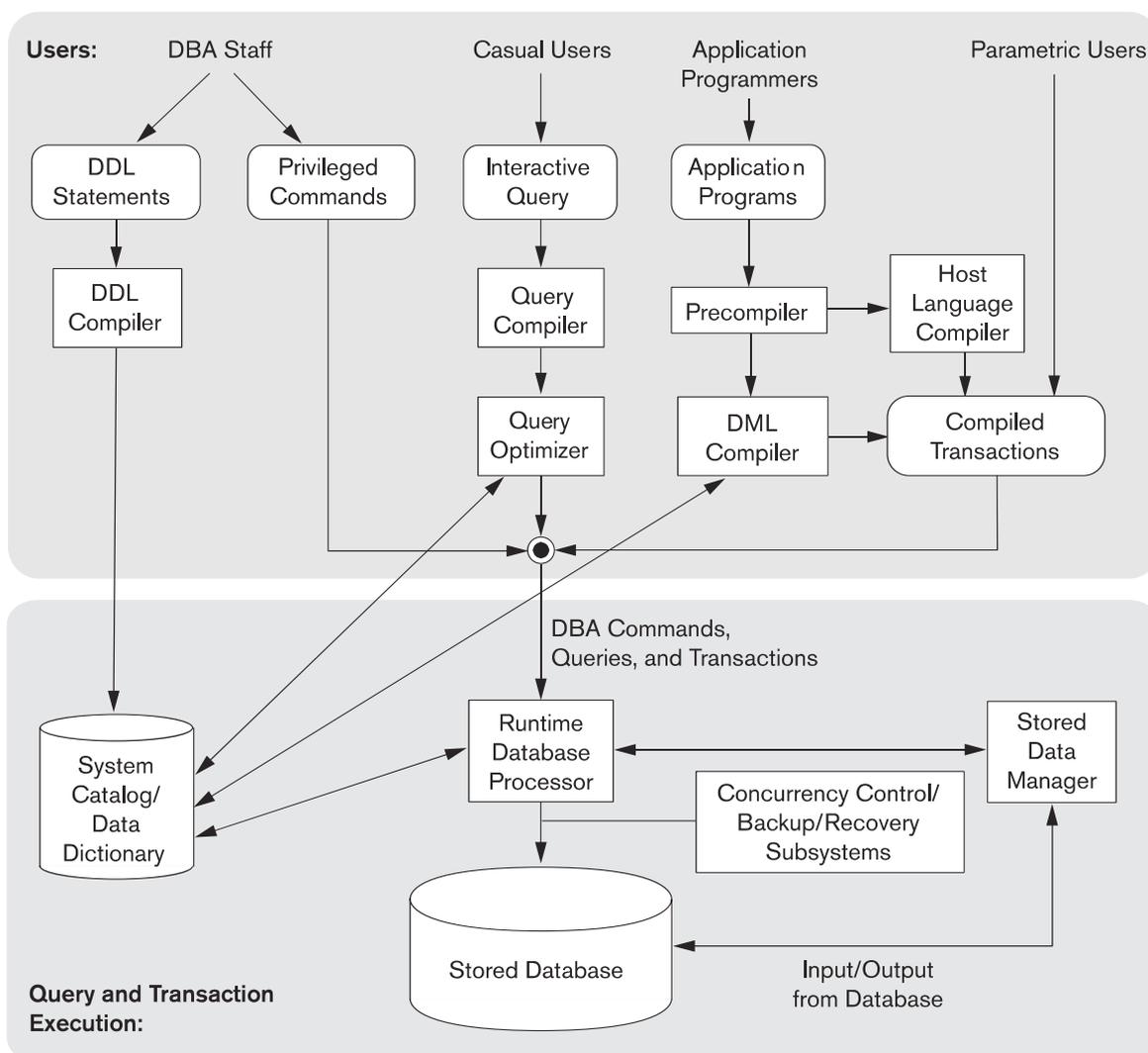


FIGURE 2.4 : Les interactions de chaque type d'utilisateurs avec le SGBD (Elmasri et Navathe 2010).

2.1.5 Limitations d'accès aux BDDs

La définition des BDDs, comme présenté ci-dessus, commence par la conception d'un modèle conceptuel, qui est ensuite transformé, dans une phase de conception logique, en un modèle logique souhaité. Cependant, le modèle conceptuel initial n'est souvent pas conservé parallèlement au modèle logique de la BDD implémentée, et les modifications ultérieures apportées à ce dernier ne sont pas propagées au premier, alors que la plupart du temps ces modifications ne sont même pas documentées (Juan et Jones 2007). Par conséquent, la BDD développée perd l'intention originale de son concepteur, et il devient alors difficile d'étendre ou de combiner cette base avec un autre modèle logique (Spanos et al. 2012).

La conception physique des BDDs doit répondre aux exigences spécifiques de l'application qui utilise les BDDs. La manière dont les données sont organisées dans les BDDs est donc différente de celle souhaitée par les utilisateurs finaux. Autrement dit, il existe un gap entre le point de vue informatique des concepteurs des BDDs et le point de vue du domaine des acteurs métiers. Cela rend les données difficilement accessibles par les utilisateurs non techniciens. Trouver des bonnes données pour des besoins d'informations spécifiques est donc une activité difficile, fastidieuse et sujette à des erreurs.

2.2 INTERROGATION DES BASES DE DONNÉES

Dans cette section, nous étudions des méthodologies et des techniques permettant de faciliter l'interrogation des bases de données. Avant cela, nous examinons les approches par l'utilisation traditionnelle des langages de requêtes.

2.2.1 Les langages de requêtes

Les langages de requêtes permettent aux différents utilisateurs d'accéder aux BDDs pour extraire des informations (Catarci et al. 1997, Rishe 2004) ou rajouter des valeurs. Chaque BDD implémente son propre langage de requête : SQL (Chamberlin et Boyce 1974) pour les BDD relationnelles (BDDR), *XQuery*¹³ pour les BDDs XML, OQL (Object Query Language) (Alashqur et al. 1989) pour les BDDs *Orientées Objets* etc. Les langages de requêtes les plus utilisés sont des langages de programmation (Date 1981; 2003) qui demandent un certain nombre de connaissances sur la syntaxe, de compétences et d'informations techniques à la fois du domaine d'application, du système et de ses mécanismes d'interaction (Catarci et al. 1997, Rishe 2004, Soylu et al. 2016) pour formuler des requêtes. Ces langages ne permettent pas de comprendre la sémantique des données, ni ne fournissent aucune indication pour les besoins d'information des utilisateurs. En général, ils sont difficilement utilisés par les utilisateurs inexpérimentés ou non-techniciens (Catarci et al. 1997).

2.2.2 Simplification de l'interrogation les BDDs

Dès les années 1990s, des méthodologies et principes facilitant les Interactions Homme-Machine (*Humain-Machine Interaction*) sont proposées par (Shneiderman 1996, Flickner

13. <https://www.w3.org/TR/xquery-30/>

et al. 1995, Catarci 2000, Helander et al. 1997, Calvanese 2012, Rodriguez-Muro et al. 2013). Ces travaux portent sur l'amélioration de la recherche d'information et peuvent être divisés en différentes pistes :

- **Recherche par mots-clés ; Utilisation d'un langage naturel restreint**^{14, 15}, ou un **système de dialogue** (dialog systems¹⁶) (Lamberti et al. 1994, Hendrix et al. 1978, Androutsopoulos et al. 1995, Kaufmann et Bernstein 2010, Safari et Patrick 2014, Waltinger et al. 2014).
- **Utilisation des langages de manipulation directe** (Shneiderman 1982) caractérisée par la visibilité des objets d'intérêt et le remplacement de la syntaxe du langage formel par la manipulation directe des objets (Helander et al. 1997) comme les systèmes d'interrogation visuels (*Visual Query System* (VQS)).
- **Accès sémantique aux données** : Cette approche est basée principalement sur l'usage des ontologies dans une couche conceptuelle où les utilisateurs peuvent effectuer des recherches sémantiques (Necib 2008, Calvanese 2012, Rodriguez-Muro et al. 2013, Giese et al. 2013).

2.2.2.1 Recherche par mots-clés/ langage naturel

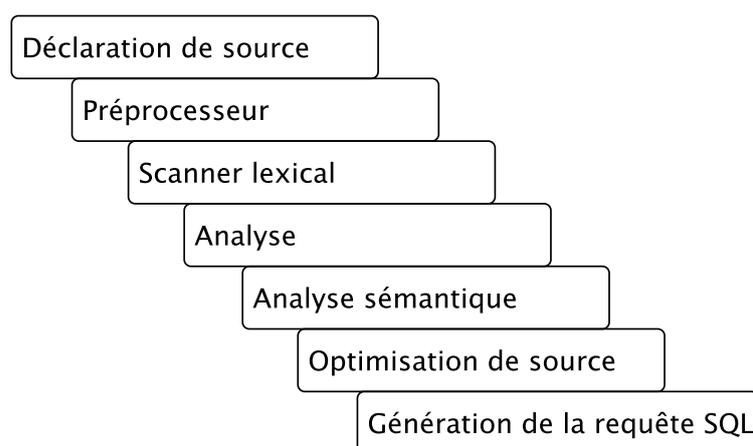


FIGURE 2.5 : *Processus de transformation de la requête en langage naturel à la requête SQL* (Chakrabarti et al. 2010)

Le succès des moteurs de recherche en ligne comme Google, Bing, Yahoo, etc. a popularisé la recherche de données basée sur des mots-clés. Ce type de recherche permet aux utilisateurs de requêter une énorme quantité de données en un temps très court, simplement avec un (des) mot(s)-clé(s). Bien que les SGBDs actuels peuvent traiter efficacement et simultanément un grand nombre de requêtes complexes sur des ensembles de données volumineuses et diverses, ils demandent des connaissances prérequisées sur les langages de requêtes formels et structurés. De nombreux travaux ont proposé d'utiliser le langage naturel pour traiter la recherche par mots-clés (Agrawal et al. 2002, Su et Widom 2005, Chen et al. 2011, Hristidis

14. <http://meta-guide.com/natural-language-sql>

15. <http://meta-guide.com/data-storage/nldb-natural-language-database-2014>

16. <http://meta-guide.com/dialog-systems/aiml/aiml-sql>

et Papakonstantinou 2002, Chen et al. 2009, Chakrabarti et al. 2010, Elbassuoni et al. 2010, Chaitali et al. 2016, Chen 2010).

Pour appliquer la recherche par mots-clés sur les données structurées, une approche possible est d'identifier la connexion (*association*) des données qui contiennent les mots clés saisis par l'utilisateur. Chakrabarti (Chakrabarti et al. 2010) a introduit un résumé des travaux basés sur cette approche. Les données générées par l'utilisateur peuvent être annotées et représentées en [Resource Description Framework \(RDF\)](#)¹⁷. Le langage naturel peut être utilisé pour requêter ce type de données (Elbassuoni et al. 2010). Les besoins d'information de l'utilisateur exprimés en langage naturel sont transférés en requêtes [SPARQL Protocol and RDF Query Language \(SPARQL\)](#)¹⁸ sur la collection des ressources RDF. Les besoins d'informations exprimés en langage naturel peuvent être également transformés en requêtes SQL sur les données relationnelles (Chaitali et al. 2016). La Figure 2.5 illustre un tel processus.

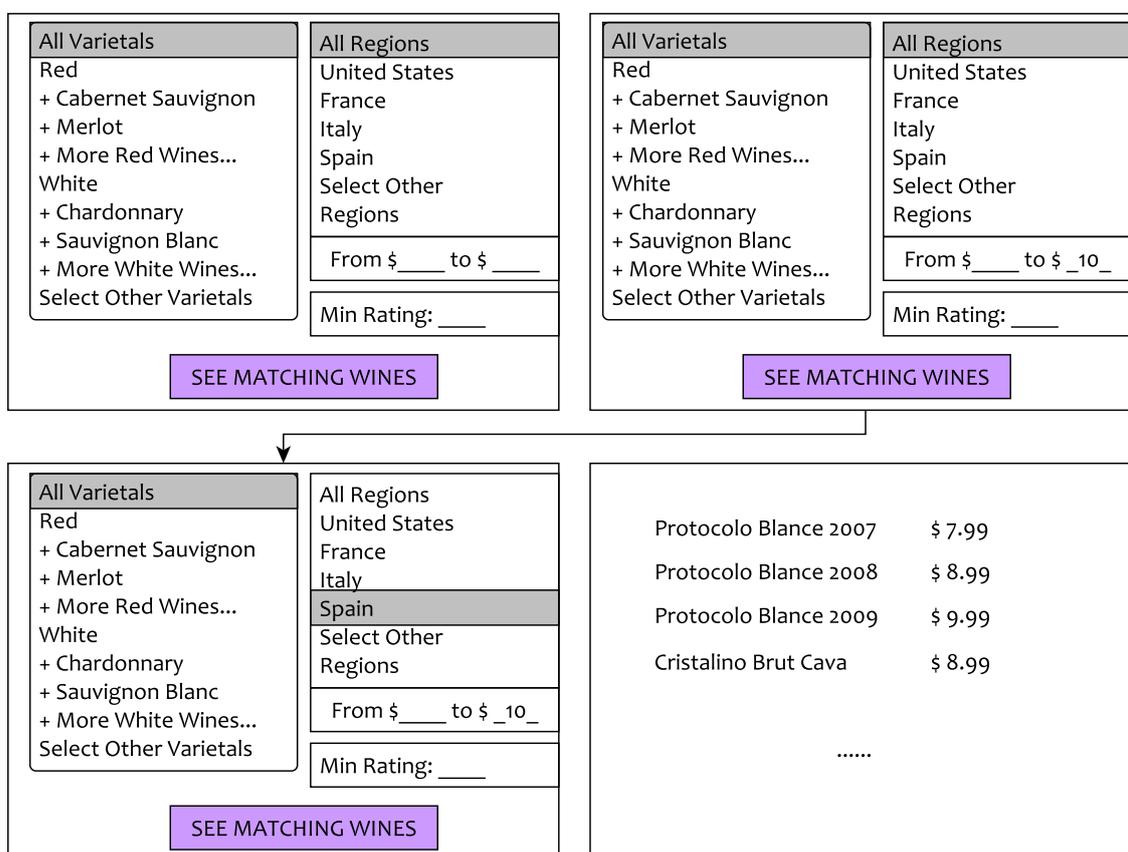


FIGURE 2.6 : L'interface de la recherche à facettes (Tunkelang 2009).

2.2.2.2 Les requêtes à facette

La recherche à facettes (*recherche facettée, navigation à facettes*) utilise une structure hiérarchique (*taxonomie*) pour permettre aux utilisateurs de parcourir les informations en choisissant parmi un ensemble prédéterminé de catégories. Cela permet à un utilisateur de saisir une requête simple, puis d'affiner les options de recherche en naviguant en profondeur. L'uti-

17. [ResourceDescriptionFramework](#)

18. <https://www.w3.org/TR/rdf-sparql-query/>

lisateur est donc assisté par une catégorisation de données pour la définition de sa requête. Ce type de requêtes est largement utilisé dans les sites de commerce électronique comme Amazon et eBay (Tunkelang 2009, Suominen et al. 2007, Ben-Yitzhak et al. 2008).

2.2.2.3 Les systèmes d'interrogation visuels

Le **Système d'Interrogation visuel (VQS)** est un système de requête qui utilise des représentations visuelles pour représenter le domaine d'intérêt et d'exprimer des demandes associées (Catarci et al. 1997). Les systèmes d'interrogation visuels peuvent être classifiés en fonction du type de représentation de données et des techniques d'interaction utilisées (Catarci et al. 1997).

Les systèmes d'interrogation visuels peuvent utiliser les langages de requête visuels pour interroger les sources de données. Ces langages sont basés sur une sémantique formelle bien définie avec une notion de syntaxe visuelle (Burnett 2001, Burnett et Baker 1994). Ils peuvent permettre une formulation visuelle de la requête. Les visualisations informent collectivement les utilisateurs sur les concepts de schéma, affordances du système et des données, à travers un certain nombre de variables sensorielles telles que la texture, la couleur, la taille et la forme, et permettent aux utilisateurs de communiquer leurs besoins d'information en interagissant avec eux (Catarci 2000, Catarci et al. 1997, Epstein 1991). Trois critères sont utilisés pour évaluer un VQS : l'Expressivité, la Convivialité et l'Interaction.

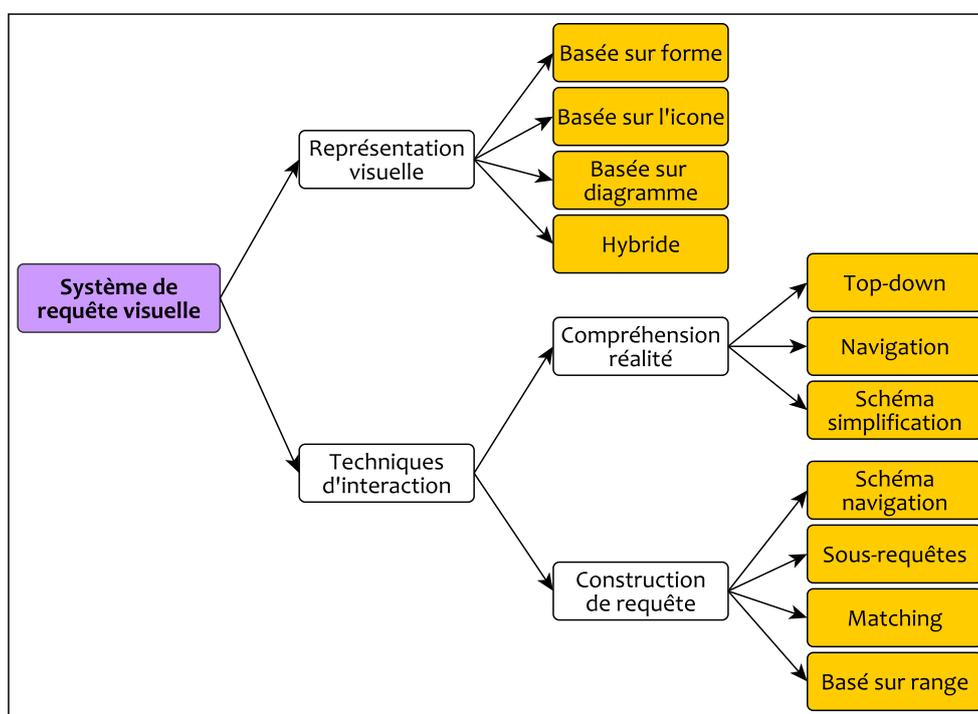


FIGURE 2.7 : La classification des VQSs selon le type de la représentation visuelle et des techniques d'interaction (Soylu et al. 2016).

Parmi les références citées ci-dessus, à ce stade, nous pouvons supposer que les démarches de formulation visuelle de requête permettent aux différents acteurs métiers de s'affranchir de la complexité de la structuration du modèle de données.

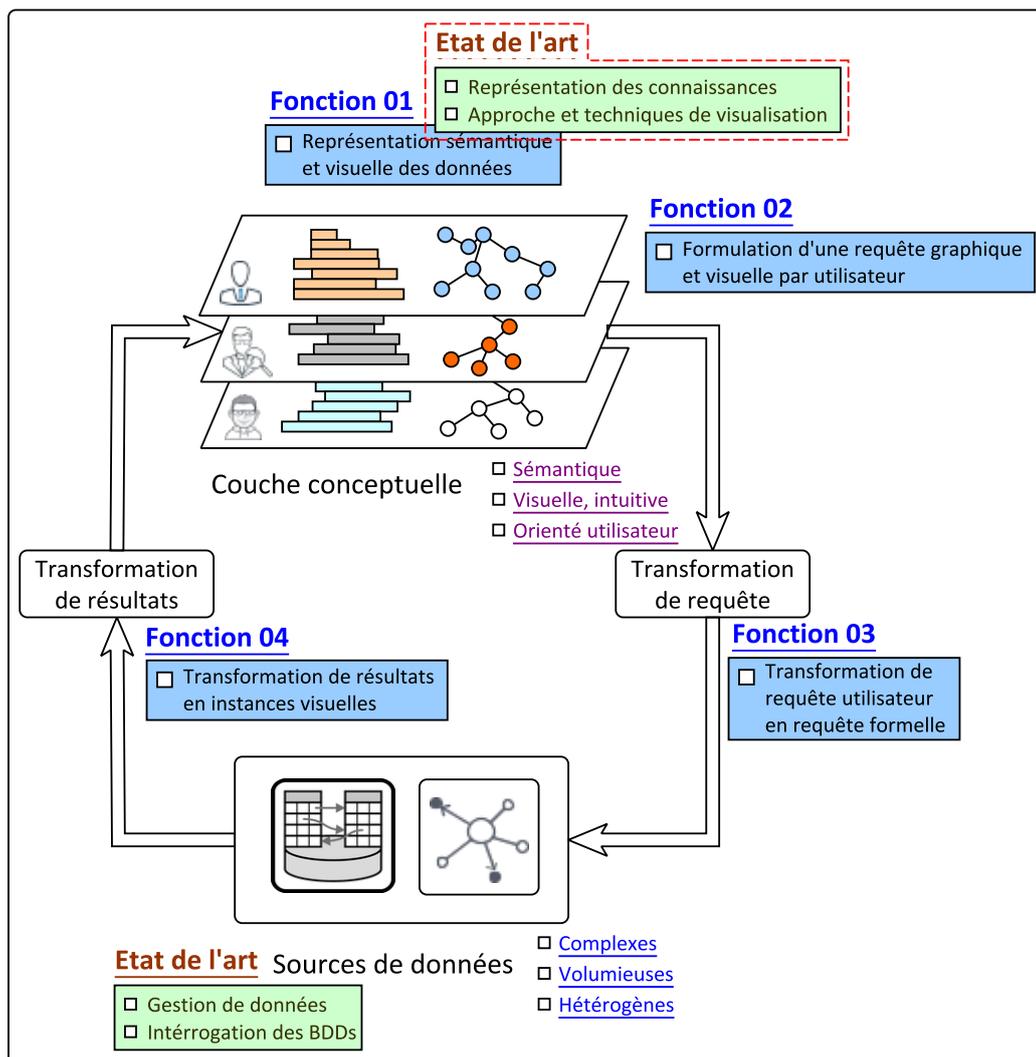


FIGURE 2.8 : L'état de l'art sur les techniques de visualisation et la représentation des connaissances.

Cependant, cela implique de rendre cette formulation à géométrie variable c'est-à-dire en fonction du profil de l'utilisateur. Pour la formulation, l'utilisateur doit voir et comprendre la structure du modèle de données. Il est donc impératif de pouvoir adapter la représentation du modèle de données en fonction du profil de l'utilisateur.

Par conséquent, nous allons dans les sections suivantes explorer les travaux de la littérature portant sur :

- La représentation des connaissances.
- Les approches et techniques de visualisation.

2.3 REPRÉSENTATION DES CONNAISSANCES

2.3.1 Donnée, information et connaissance

La connaissance est un des concepts les plus abstraits du vocabulaire humain. En informatique, la connaissance est souvent définie par rapport aux données et informations. L'infor-

mation est définie en termes de donnée et la connaissance est définie en termes d'information (Rowley 2007).

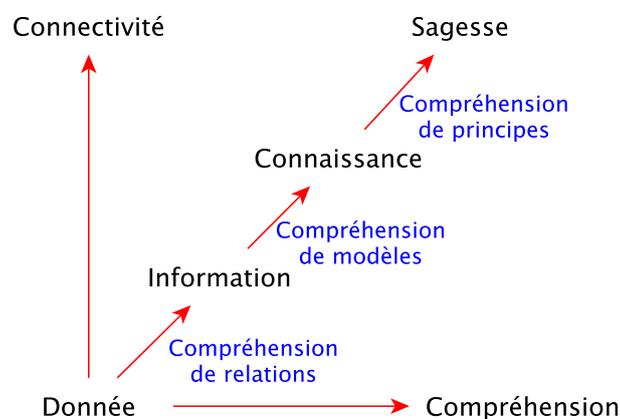


FIGURE 2.9 : Les transitions de donnée à sagesse supportées par la compréhension ¹⁹.

Selon Bellinger¹⁹ : "Les données représentent un fait ou une déclaration d'événement sans rapport avec d'autres choses. Les informations incarnent la compréhension d'une relation, éventuellement cause et effet, et les connaissances représentent un modèle qui se connecte et fournit généralement un haut niveau de prévisibilité quant à ce qui est décrit ou ce qui va se passer ensuite" (Figure 2.9). C'est la compréhension qui joue les transitions de chaque étape à la suivante.

2.3.2 Pourquoi la représentation des connaissances

Les connaissances sont souvent cachées derrière l'ensemble de données et d'informations et doivent être "dévoilées" pour devenir utilisables (Bertschi et al. 2013). La représentation des connaissances est un sous-domaine de l'Intelligence Artificielle qui étudie les modèles informatiques, les langages et les systèmes capables de représenter des connaissances de manière explicite ou de raisonner sur ces connaissances (Chein et al. 2013).

Selon l'Encyclopédie des sciences cognitives (Nadel 2003), la représentation de connaissances est chargée de concevoir et de mettre en oeuvre des moyens de représentation des informations dans les machines afin que les programmes puissent l'utiliser :

- Pour obtenir des informations qui en découlent,
- Pour converser avec des gens en langages naturels,
- Pour planifier les activités futures,
- Pour résoudre des problèmes dans des domaines qui exigent normalement une expertise humaine.

2.3.3 Formes de représentation des connaissances

Un rôle important de la représentation des connaissances est de fournir un support pour l'expression et la communication humaines (Davis et al. 1993), le langage naturel est donc le

19. <http://www.systems-thinking.org/dikw/dikw.htm>

meilleur moyen en raison de son expressivité. Cependant, le langage naturel est souvent ambigu, incohérent et difficile à modéliser, du coup, il n'est pas approprié pour la représentation des connaissances dans les systèmes informatiques (Jakus et al. 2013). Les formes suivantes sont plus utilisées : Réseau Sémantique, Cadre, Description Logique, Graphe conceptuel et Logique Floue. (la liste détaillée est illustrée dans la Figure 2.10).

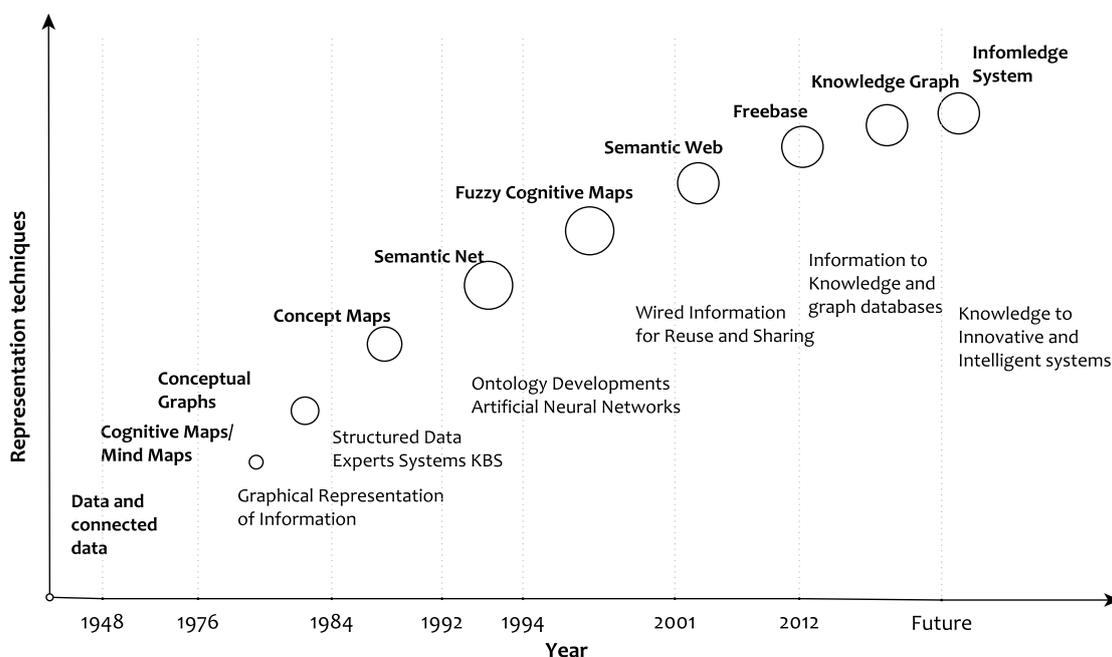


FIGURE 2.10 : Évolution des techniques de représentation des connaissances (Malhotra et Nair 2015).

2.3.3.1 Réseau Sémantique (Semantic network)

Le réseau sémantique, introduit par Quillian (Quillian 1968), utilise un graphe de noeuds étiquetés et d'arcs dirigés étiquetés pour encoder des connaissances (*noeuds* : *objets, concepts ou événements* ; *arcs* : *relations* entre deux noeuds) (Figure 2.11). Les relations souvent utilisées sont "*est-un*", "*un-type-de*", "*partie-de*", etc. Utilisées pour lier une classe à sa superclasse, les relations héritées (comme "*est-un*" ou "*partie-de*" par exemple) permettent d'effectuer des raisonnements dans les réseaux sémantiques. La représentation graphique est un grand avantage du réseau sémantique, qui rend des connaissances représentées plus facilement compréhensibles. Cependant, le réseau sémantique peut devenir difficile à interpréter quand le nombre de noeuds et arcs devient trop important.

Le Web Sémantique présenté dans la section 2.6 utilise les concepts du réseau sémantique pour rendre les contenus web accessibles pas l'humain et la machine. Deux langages sont utilisés pour cet objectif : RDF (*Resource Description Framework*) à bas niveau pour modéliser des informations implémentées dans la page Web, et *Ontology Web Language* (OWL) qui ajoute une sémantique aux ressources RDF.

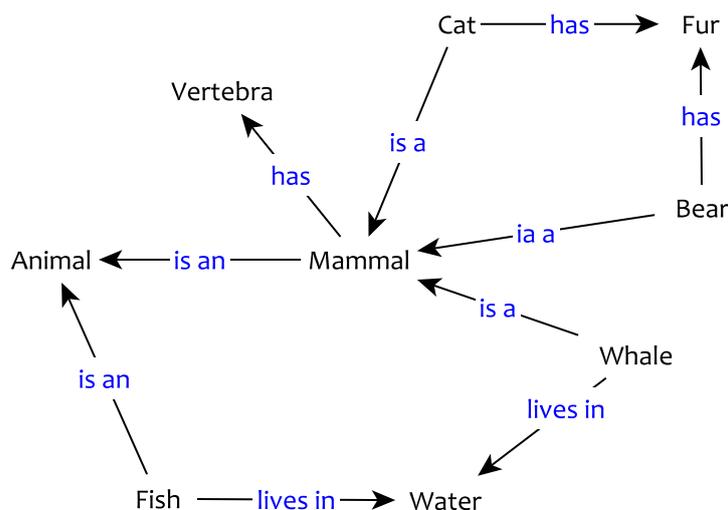


FIGURE 2.11 : Exemple du réseau sémantique.

2.3.3.2 Cadre (Frame)

Les réseaux sémantiques se sont transformés en cadres dans les années 70s et 80s (Minsky 1974). Un cadre s'apparente à un objet dans les langages orientés objet, mais a plus de méta-données. Il est utilisé pour décrire les connaissances reliées à un objet. Un cadre représente une entité comme un ensemble de slots (*attributs*) et des valeurs associées. Un slot représente une valeur ou une relation avec un autre cadre. Un slot a une ou plusieurs facettes. Une facette représente les caractéristiques de la relation. La représentation de connaissances en cadres est expressive et facilement compréhensive, elle est assez flexible pour ajouter/modifier des attributs et relations. Par contre, il n'y a pas de mécanismes d'inférence associés. Il y a des problèmes d'héritage, en particulier pour les héritages multiples.

2.3.3.3 Description Logique (Logic Description)

La description logique est un langage de représentation des connaissances utilisé pour représenter un domaine d'application à travers des sémantiques formelles à base de logique (Baader et al. 2003). Un point fort de la description logique est la capacité de raisonnement. Une sorte d'inférence (en plus de l'*abduction*, la *déduction* et l'*induction*) est la **classification automatique** : La nouvelle description peut être placée automatiquement à la bonne place dans la hiérarchie d'objets prédéfinis.

2.3.3.4 Graphe Conceptuel (Conceptual Graph)

Inventé par Sowa (Sowa 1984) le graphe conceptuel est un formalisme de représentation des connaissances qui est basé sur les graphes existentiels²⁰ et le réseau sémantique. L'objectif du graphe conceptuel est de représenter les connaissances "*précise logiquement, humainement lisible, et informatiquement traçable*"²¹. Les connaissances sont représentées par des graphes

20. <http://www.jfsowa.com/peirce/ms514.htm>

21. <https://conceptualgraphs.org/>

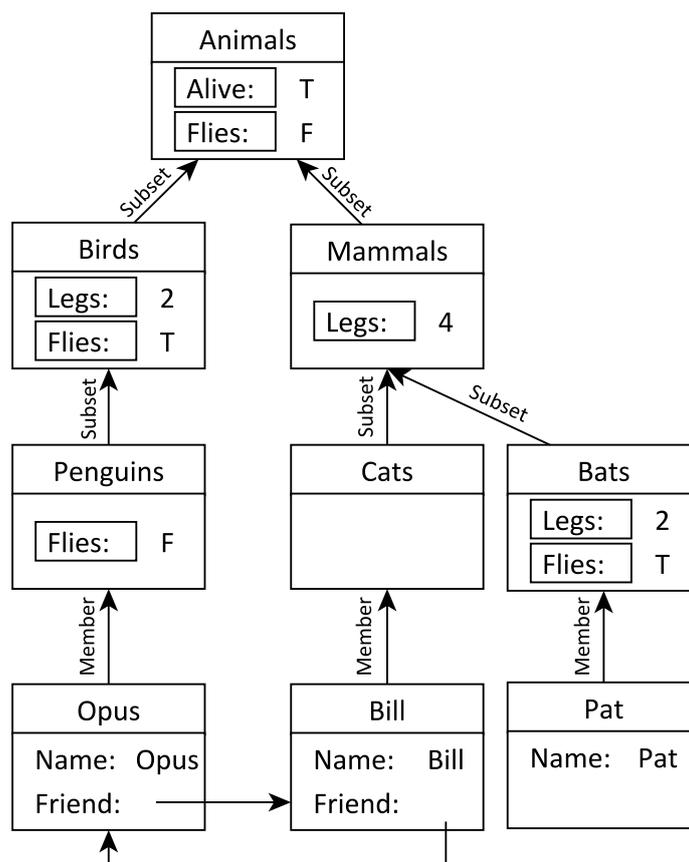


FIGURE 2.12 : Base de connaissance représentée par les cadres.

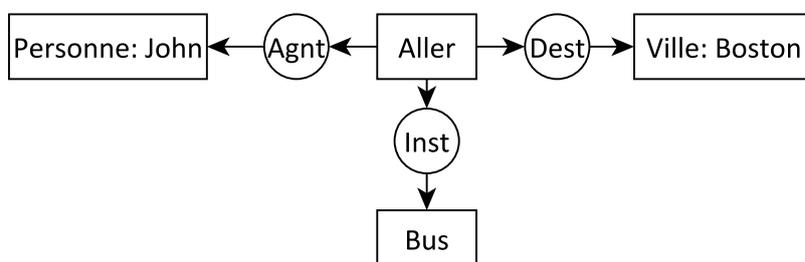


FIGURE 2.13 : Le graphe conceptuel utilisé pour exprimer "John va à Boston en bus" (Sowa 1984).

étiquetés et les mécanismes de raisonnement sont basés sur des opérations de graphes. La Figure 2.13 illustre un exemple du graphe conceptuel :

2.3.3.5 Logique Floue (Fuzzy Logic)

La logique floue est une extension des systèmes logiques traditionnels qui a pour objectif de fournir un formalisme efficace pour représenter des connaissances dans un environnement incertain et imprécis²². "Contrairement à d'autres approches typiques de la représentation des connaissances qui ne permettent qu'une représentation exacte des connaissances et tirent des conclusions à partir de celle-ci, la logique floue fonctionne d'une manière beaucoup plus similaire à celle de l'esprit humain" (Jakus et al. 2013). Du coup, la logique floue traite des raisonnements qui sont appropriés plutôt qu'exacts et fixés.

22. http://link.springer.com/chapter/10.1007%2F978-1-4615-3640-6_1

Discussion

En informatique, "la représentation des connaissances utilise des symboles formels pour représenter un ensemble de propositions de certains agents putatifs" (Brachman et Levesque 2004). Une instance de la représentation des connaissances peut contenir seulement un nombre limité de propositions, du coup il faut choisir les propositions les plus pertinentes en négligeant les autres (Jakus et al. 2013) quand on modélise la réalité. Ces propositions sont appelées les "engagements ontologiques" ("*ontological commitments*") (Davis et al. 1993). Les ontologies sont utilisées dans les systèmes d'intelligence artificielle comme des modèles conceptuels de ce qui "existe" et elles sont encodées de façon informatiquement traitable à travers des formalismes de représentation des connaissances présentés ci-dessus (Grimm et al. 2004). Dans la section suivante, nous présentons des éléments bibliographiques concernant l'ontologie afin de mettre en évidence son rôle important dans la couche conceptuelle de notre approche.

2.3.4 Ontologie

2.3.4.1 Définition

Le mot "Ontologie" est utilisé dans des disciplines très diverses avec différentes interprétations. Il n'existe donc pas de définition unique qui est acceptée et partagée par tous les experts. Nous présentons dans cette section, une collection exhaustive de l'ensemble de définitions que nous avons trouvées dans la littérature.

Ces définitions sont séparées en deux domaines différents : "**Philosophie**" et "**Informatique**". Nous distinguons, le terme innombrable "*Ontologie*" ("*O* en majuscule) pour le domaine de la philosophie et la terme dénombrable "*ontologie(s)*" ("*o* en minuscule) pour le domaine de l'informatique (Guarino and Giaretta 1995).

PHILOSOPHIE

Dans le domaine de la philosophie, "*Ontologie*" se réfère à la branche qui étudie la nature et la structure de l'existence et de l'être. Elle était née pour répondre aux questions fondamentales de la philosophie telles que : *Qu'est-ce qu'une chose? Quelles sont les significations de l'être? Qu'est-ce que l'existence? Quelles sortes de choses?* etc. (Craig 1998).

Selon Aristote²³, Ontologie offre des catégories primitives (*substance, quantité, qualité, etc.*) pour toutes les choses qui existent.

INFORMATIQUE

Dans le domaine de l'Informatique, "*ontologie*" est utilisée comme un terme technique dénotant un artefact ou un type spécifique de l'information qui représente la structure d'un système (Guarino et al. 2009). Selon Gruber (Liu et Zsu 2009), les ontologies sont désignées pour un but, qui est de permettre la modélisation des connaissances d'un domaine, réel ou imaginaire.

Fondé en 1956 à la Conférence Dartmouth²⁴, l'Intelligence Artificielle, une discipline scien-

23. <http://www.iep.utm.edu/aristot1/>

24. https://en.wikipedia.org/wiki/Dartmouth_Conferences

tifique de l'Informatique, est définie comme "la science et l'ingénierie de la fabrication de machines intelligentes, en particulier les programmes informatiques intelligents, qui sont liées à l'utilisation des ordinateurs pour comprendre/ émuler l'intelligence d'humaine" (McCarthy 2007). La construction d'un système intelligent puissant demande une capture efficace de connaissances du domaine (Neches et al. 1991). Depuis le milieu des années 1970, les chercheurs dans ce domaine ont utilisé les ontologies en tant que **modèle de modélisation des connaissances** et que modèle de calcul permettant certains types de **raisonnement automatisés**.

Dans divers contextes d'application, et au sein de différentes communautés, les ontologies ont été explorées à partir de différents points de vue, et il existe plusieurs définitions de ce qu'est une ontologie (Grimm et al. 2011).

(Neches et al. 1991)

"Les ontologies établissent les règles de base pour la modélisation d'un domaine en définissant les termes de base et les relations qui composent le vocabulaire d'une zone thématique".

(Neches et al. 1991)

La première définition de l'ontologie que nous avons trouvée est proposée par Neches (Neches et al. 1991). Dans son approche, les ontologies sont définies comme «les termes de base et les relations comprenant le vocabulaire d'une zone thématique ainsi que les règles de combinaison des termes et des relations pour définir l'extension du vocabulaire». Pour construire des ontologies, selon cette définition, il faut identifier un ensemble de termes pertinents, les vocabulaires significatifs pour les nommer et les relations qui les relient les uns aux autres. Utilisées comme une composante réutilisable du système de connaissances, les ontologies permettent non seulement de préserver, de capturer, de partager et de réutiliser des connaissances, mais aussi d'inférer des nouvelles connaissances grâce aux règles prédéfinies entre termes et relations.

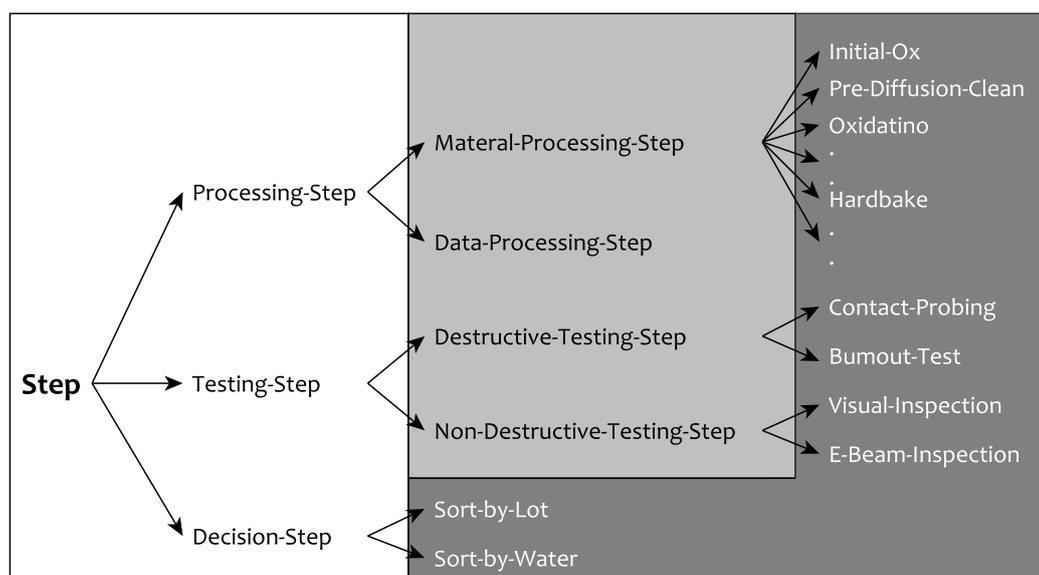


FIGURE 2.14 : L'ontologie MKS (Manufacturing Knowledge System) des opérations de fabrication proposée par Neches (Neches et al. 1991).

(Gruber 1993, Borst 1997)

Selon Gruber et Borst, (Gruber 1993, Borst 1997), une ontologie est une représentation explicite d'une conceptualisation d'un domaine.

"Une ontologie est une spécification explicite d'une conceptualisation" (Gruber 1993)

"Les ontologies sont définies comme une spécification formelle d'une conceptualisation partagée" (Borst 1997)

(Studer et al. 1998)

En 1998, Studer (Studer et al. 1998) a introduit une définition de l'ontologie qui est largement acceptée par la communauté. Cette définition est une combinaison de deux définitions proposées par Gruber (Gruber 1993) et Borst (Borst 1997).

"Une ontologie comme spécification explicite formelle d'une conceptualisation partagée. La conceptualisation se réfère à un modèle abstrait de quelque phénomène dans le monde en identifiant les concepts pertinents de ce phénomène" (Studer et al. 1998)

Définie comme une spécification des connaissances du domaine, des caractéristiques de l'ontologie doivent être prises en compte (Grimm et al. 2011) : **Formelle, Explicite, Consensuelle, Conceptuelle** et **Spécifique à un domaine**.

- **Formelle** : Le langage d'expression des ontologies est formel, c'est-à-dire, les spécifications décrites dans les ontologies sont interprétables par les machines.
- **Explicite** : Les types de concepts utilisés et les contraintes de leurs utilisations sont définis explicitement. Une ontologie doit indiquer explicitement les connaissances pour les rendre accessibles aux machines.
- **Consensuelle** : Ontologie capture les connaissances consensuelles qui sont acceptées par un groupe de personnes plutôt que des individus.
- **Conceptuelle** : Les ontologies représentent les connaissances d'une manière conceptuelle en termes de symboles conceptuels qui peuvent être compréhensives intuitivement par les humains, car ils correspondent aux éléments de leurs modèles mentaux. Les concepts ontologiques sont nommés en termes généraux fréquemment utilisés par les acteurs du domaine.
- **Spécifique à un domaine** : Les connaissances du domaine sont diverses. Les spécifications représentées dans une ontologie sont limitées à un domaine d'intérêt spécifique.

(Bernaras et al. 1996, Swartout et al. 1997)

Il existe un groupe de définitions qui sont basées sur le processus de construction des ontologies (Gómez-Pérez et al. 2004). Une définition dans ce groupe est fournie par Bernaras (Bernaras et al. 1996) :

"Une ontologie fournit les moyens de décrire explicitement la conceptualisation derrière le savoir représenté dans une base de connaissances" (Bernaras et al. 1996)

Selon ces définitions, les ontologies peuvent être extraites et construites à partir des applications du système de connaissances (Davis 1986) par une approche "bottom-up" (comme illustré dans la Figure 2.15). Selon (Grimm et al. 2011), une autre stratégie de construction des ontologies et de réutiliser des ontologies génériques telle que SENSUS (Swartout et al. 1997). Dans cette optique, une ontologie peut être définie comme suivante :

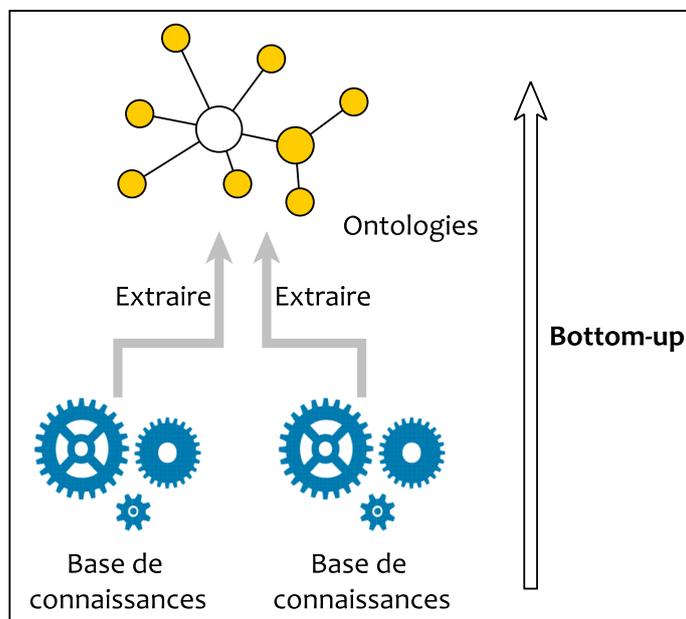


FIGURE 2.15 : La construction des ontologies à partir des bases de connaissances (Davis 1986).

"Une ontologie est un ensemble structuré hiérarchiquement de termes pour décrire un domaine qui peut être utilisé comme un fondement squelettique pour une base de connaissances"
 (Swartout et al. 1997)

Lightweight et heavyweight ontologies Une ontologie peut être considérée comme une taxonomie des concepts, des relations entre concepts et des propriétés qui décrivent les concepts. Dans ces cas, elle est appelée ontologie "lightweight". En revanche, les ontologies "heavyweight" ajoutent des axiomes et des contraintes aux ontologies "lightweight". (Grimm et al. 2011). Ce type d'ontologie représente le domaine d'une manière plus profonde.

Les ontologies sont utilisées dans des domaines variés. Elles sont souvent confondues avec le "modèle conceptuel" de la communauté de base de données qui représente aussi le domaine par l'ensemble de concepts, des relations et des propriétés (Entity-Relationship par exemple). Pour clarifier et positionner l'usage des ontologies dans leurs domaines d'application, Uschold (Uschold et Jasper 1999) a précisé :

"Une ontologie peut prendre une variété de formes, mais elle comprendra nécessairement un vocabulaire de termes et une certaine spécification de leur signification. Cela comprend des définitions et une indication de la façon dont les concepts sont interdépendants qui imposent collectivement une structure sur le domaine et limitent les interprétations possibles des termes"
 (Uschold et Jasper 1999)

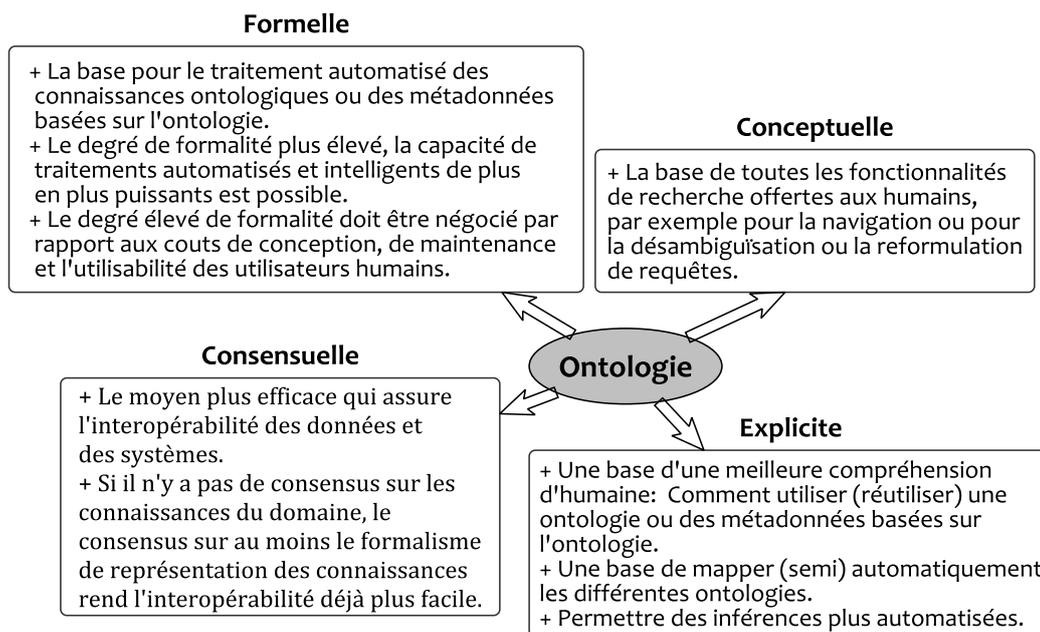


FIGURE 2.16 : Les caractéristiques d'une ontologie (Grimm et al. 2011).

Pour conclure cette section, en récapitulant toutes les définitions présentées au-dessus, nous listons les caractéristiques indispensables d'une ontologie (Dillon et al. 2008) :

- **Explicite, Partagée et Consensuelle** : Les connaissances représentées dans une ontologie doivent être explicites, utilisées, partagées et acceptées par un groupe de personnes dans la communauté. Les ontologies sont souvent construites coopérativement.
- **Formelle** : Les ontologies doivent être processable par la machine.
- **Raisnable** : En utilisant les règles prédéfinies de l'ontologie, des nouvelles spécifications (ou *connaissances*) peuvent être extraites. La fonction d'inférence est un point fort des ontologies.

2.3.4.2 Usage de l'ontologie

"Ontologies are widely used in Knowledge Engineering, Artificial Intelligence and Computer Science, in applications related to knowledge management, natural language processing, e-commerce, intelligent integration information, information retrieval, database design and integration, bio-informatics, education, and in new emerging fields like the Semantic Web"
 (Gómez-Pérez et al. 2004)

Les ontologies sont utilisées pour différents buts dans différents domaines. Au début des années 90, les chercheurs de la communauté de l'Acquisition de Connaissances ont utilisé les ontologies dans le but de capturer, préserver et représenter des connaissances (Staab et Studer 2003). De temps en temps, les ontologies ont été le sujet de recherche d'autres disciplines telles que : le traitement automatique du langage naturel, gestion de connaissances, e-commerce, etc. Plus récemment, la notion d'ontologie se généralise également dans des domaines tels que l'Intégration intelligente de l'information, les systèmes d'information coopératifs, la récupération de l'information, etc. (Tursi 2009).

Parmi les travaux remarquables sur l'ontologie, nous mettons en évidence la contribution de Jasper et Uschold ([Jasper et al. 1999](#)). Selon ces auteurs, les ontologies peuvent être utilisées pour : **Aider à la communication entre les humains** ; **Assurer l'interopérabilité entre différents systèmes** et **Améliorer la capacité des systèmes d'information**. Notre approche s'intègre dans cette démarche car nous souhaitons améliorer le système d'information et le partage de données entre les acteurs dans une organisation.

1. **Aider à la communication entre les humains** : Une des caractéristiques les plus importantes de l'ontologie est qu'elle est une représentation "explicite" de connaissances. Les spécifications (concepts, relations, axiomes, etc.) explicitement définies dans l'ontologie facilitent la communication entre différents individus venant d'un ou de domaines variés. Les spécifications explicites permettent aussi une interprétation commune et cohérente des ressources ontologiques.
2. **Permettre l'interopérabilité entre différents systèmes** : Les données rendues disponibles dans les systèmes d'information actuels sont hétérogènes et faiblement structurées (*textuelles, visuelles, images, vidéos, etc.*). Chaque système utilise son propre modèle pour la gestion des données. La communication entre différents systèmes devient de plus en plus, difficile à cause de l'hétérogénéité sémantique ou structurelle de ces systèmes. Les ontologies peuvent être utilisées comme un format d'échange qui assure l'interopérabilité entre les systèmes hétérogènes par la transformation entre différents langages et logiciels méthodes de modélisation.
3. **Améliorer la capacité des systèmes d'information** : Les ontologies jouent un rôle indispensable et irremplaçable dans les systèmes d'information. En fait, les données brutes en grandes quantités ne produisent pas de valeur ni d'avantages concurrentiels pour les organisations : les informations sont inutiles sans une compréhension de la façon de l'appliquer efficacement ([Fensel 2003](#)). Les données et les informations brutes doivent être transformées en connaissances facilement interprétables. L'utilisation de l'ontologie améliore de nombreux aspects de la gestion des connaissances : de la capture/ acquisition, la représentation, l'utilisation, la préservation jusqu'à l'entretien, le partage et la réutilisation des connaissances du domaine. Les ontologies sont considérées comme le pivot d'un système à base de connaissances.

Les ontologies dans le système d'information sont construites en utilisant les techniques de représentation des connaissances symboliques venant de la communauté de l'Intelligence Artificielle. Les spécifications (*axiomes, contraintes, fonctions, etc.*) définies dans une ontologie représentent une connaissance explicite et formelle du domaine d'intérêt. Les techniques de raisonnement peuvent traiter ces connaissances afin de construire des connaissances implicites qui sont cachées logiquement. Deux aspects importants du raisonnement logique à l'aide d'ontologies sont principalement examinés, à savoir la vérification de la spécification d'une ontologie et la déduction de nouveaux axiomes ([Grimm et al. 2011](#)).

- **Vérification** : Cette fonction nous permet de savoir si une ontologie est une bonne représentation du domaine d'intérêt en vérifiant tous les axiomes définis dans l'ontologie (*au moins la justesse technique*). Une ontologie contenant des informations contradic-

toires n'est pas considérée comme une bonne représentation du domaine (Grimm et al. 2011). La vérification d'une ontologie peut être réalisée automatiquement.

- **Déduction** : Basé sur des règles logiques, les nouveaux axiomes (*contraintes*) peuvent être générées à partir de celles prédéfinies dans l'ontologie. Les conclusions déductives (*dérivées*) sont diverses, des faits simples aux axiomes complexes et génériques.

Nous exploitons ces deux fonctions dans notre système afin de garantir une représentation cohérente de données entre plusieurs acteurs et d'inférer des requêtes pour répondre à leurs multiples besoins. Pour conclure cette section, la Figure 2.17 résume les avantages essentiels apportés par l'ontologie.

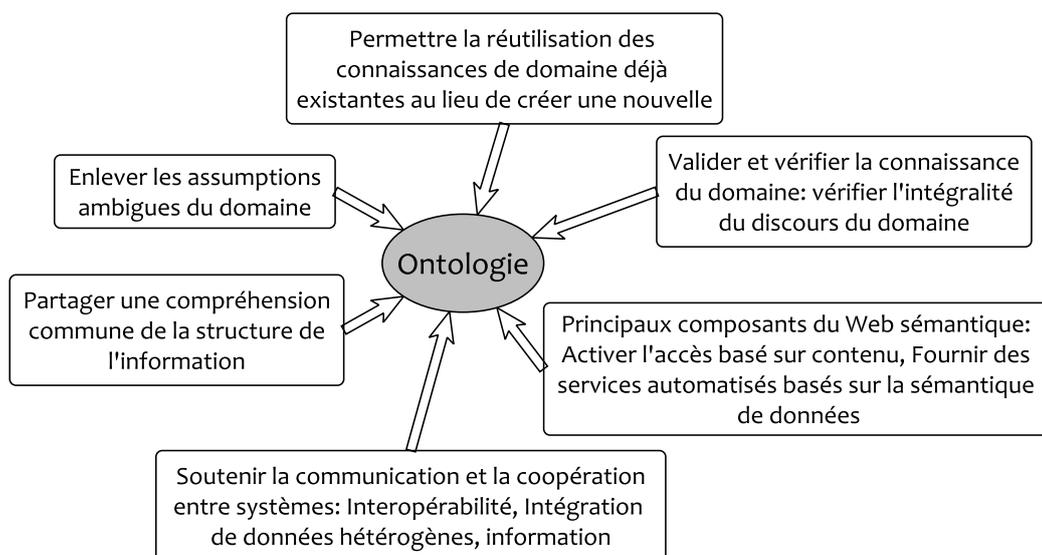


FIGURE 2.17 : Résumé des avantages de l'ontologie.

2.3.4.3 Composants d'une ontologie

Les composants d'une ontologie sont divers en fonction des langages et des techniques utilisées pour la représenter. Par exemple, le langage de description logique représente les ontologies avec trois composants : concepts, rôles et individus ; pourtant, le langage basé sur une carte ou logique de premier ordre représente les connaissances du domaine avec cinq composants : **Classes**, **Relations**, **Fonctions**, **Axiomes formels** et **Instances** (Gruber 1993, Moghaddam et Horton 2010).

- **Concept** (aussi connu comme une classe ou un terme) est un groupe, un ensemble ou une collection abstraits d'objets qui sont pris dans un sens large. C'est l'élément fondamental du domaine et il représente habituellement un groupe ou une classe dont les propriétés sont communes. Les classes dans l'ontologie sont classées en taxonomies par lesquelles des mécanismes d'héritage peuvent être appliqués. Le concept est représenté par une "*classe -supérieure*", représentant la classe supérieure, ou "**classe parent**" et une "*sous-classe*" qui représente la classe subordonnée ou "*classe enfant*". Par exemple, le concept "*ordinateur*" pourrait être représenté comme "**classe mère**" de "**ordinateur portable**" et de "*PC de bureau*".
- **Relation** (*slot or property*) est utilisée pour exprimer les relations entre deux concepts

d'un domaine donné (*ObjectProperty*) ou entre un concept et un littéral²⁵ (*Datatype-Property*). Les ontologies contiennent généralement des relations binaires.

- **Fonction** est un cas particulier de relations dans lesquelles l'élément n de la relation est unique pour les $n-1$ éléments précédents. Par exemple, la taille de l'écran peut être définie par la largeur et la hauteur de l'écran de l'ordinateur.
- **Axiome** est utilisé pour imposer des contraintes aux valeurs de concepts ou d'instances. Les axiomes sont toujours vrais et ainsi, ils sont généralement exprimés en utilisant des langages logiques comme la logique de premier ordre. Les axiomes sont utilisés pour vérifier la cohérence de l'ontologie.
- **Instance** (également appelée **individu**) est la composante d'une ontologie "au sol" qui représente un objet ou un élément spécifique d'un concept ou d'une classe. Un exemple peut représenter des objets concrets tels des personnes, des machines ou des protéines ; Ils peuvent également représenter des objets plus abstraits tels que, le travail d'une personne ou une fonction. Par exemple, "MacBook 2015" est une instance du concept "Ordinateur portable".

2.3.4.4 Typologies des ontologies

L'ontologie est utilisée avec différentes interprétations en Informatique. De nombreuses formes différentes de modèles conceptuels sont considérées également comme des ontologies (Grimm et al. 2011). Il existe certaines façons de classifier les ontologies : **Formes d'apparence**, **Scopes** et **Degrés de formalité**.

2.3.4.4.1 Selon formes d'apparence

Pour différents buts d'utilisation, une ontologie peut être représentée dans des formes variées. Un ingénieur de connaissances visualise une ontologie au moyen d'une visualisation graphique ou formelle, tandis que dans une finalité de transfert, l'ontologie est codée dans un langage de représentation qui est traitable par la machine. Pour raisonner, l'ontologie apparaît comme un ensemble d'axiomes qui constituent une théorie logique (Grimm et al. 2011).

2.3.4.4.2 Selon scopes

Les ontologies peuvent être classifiées en fonctions du type (le *scope* ou la *généralité*) de connaissances qu'elles capturent et modélisent (Grimm et al. 2011, Roussey et al. 2011).

- **Ontologies fondamentales/ supérieures** : Les ontologies génériques pouvant être appliquées à différents domaines/ applications et comparées au méta-modèle d'un schéma conceptuel (Fonseca et Martin 2007) . Ces ontologies contiennent des notions élémentaires telles que l'objet, les relations, les processus, etc. Le DOLCE (*Ontologie descriptive pour l'ingénierie linguistique et cognitive*²⁶) et le BFO²⁷ (Ontologie formelle de base sont deux des ontologies génériques les plus connues.

25. Un "littéral" représente la valeur d'une instance (des chaînes ou des entiers particuliers)

26. <http://www.loa.istc-cnr.it/old/DOLCE.html>

27. <http://ifomis.unisaarland.de/bfo/>

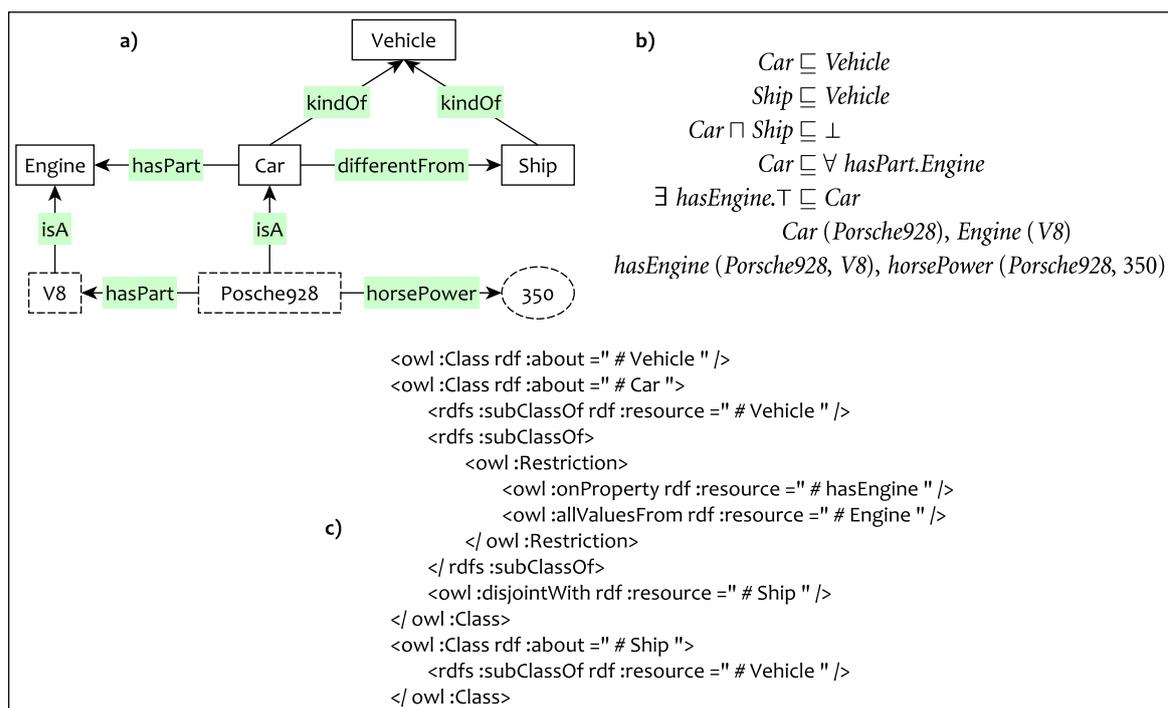


FIGURE 2.18 : 3 façons de représenter d'une ontologie : a) Visualisation graphique, b) Formules logiques, c) Langage formel (Grimm et al. 2011).

- **Ontologies générales** : Visent à exprimer une connaissance générale des domaines qui lui sont propres.
- **Ontologies de référence principales** : Ces ontologies sont reliées à un domaine, mais ils intègrent différents points de vue liés à un groupe spécifique d'utilisateurs. Elles sont le résultat de l'intégration de plusieurs ontologies de domaine. Une référence de base contient des concepts centraux et des relations du domaine.
- **Ontologie du domaine** : Contiennent des vocabulaires sur les concepts dans un domaine. Ce type d'ontologie ne s'applique qu'à un domaine avec un point de vue spécifique. Elle définit comment un groupe d'utilisateurs conceptualise et visualise une conceptualisation spécifique. Une ontologie de domaine pourrait être liée à une application spécifique (Figure 2.19).
- **Ontologie de tâche** : Contient des connaissances nécessaires pour réaliser des tâches reliées à un domaine spécifique.
- **Ontologie d'application (locale)** : Les ontologies "spéciales" qui représentent le modèle particulier d'un domaine selon un point de vue unique d'un utilisateur ou d'un développeur (*technicien de base de données* par exemple). Fonseca (Fonseca et Martin 2007) a présenté ce type d'ontologie comme une combinaison d'ontologie de domaine et l'ontologie des tâches afin de remplir le but spécifique d'une application.

2.3.4.4.3 Selon degrés de formalité

Selon Grimme (Grimm et al. 2011), les ontologies utilisées dans le contexte du Web Sémantique peuvent être distinguées en fonction de leurs degrés de formalité. Les ontologies "light-weight" ne contiennent pas ou contiennent très peu d'axiomes limitant l'utilisation de leurs

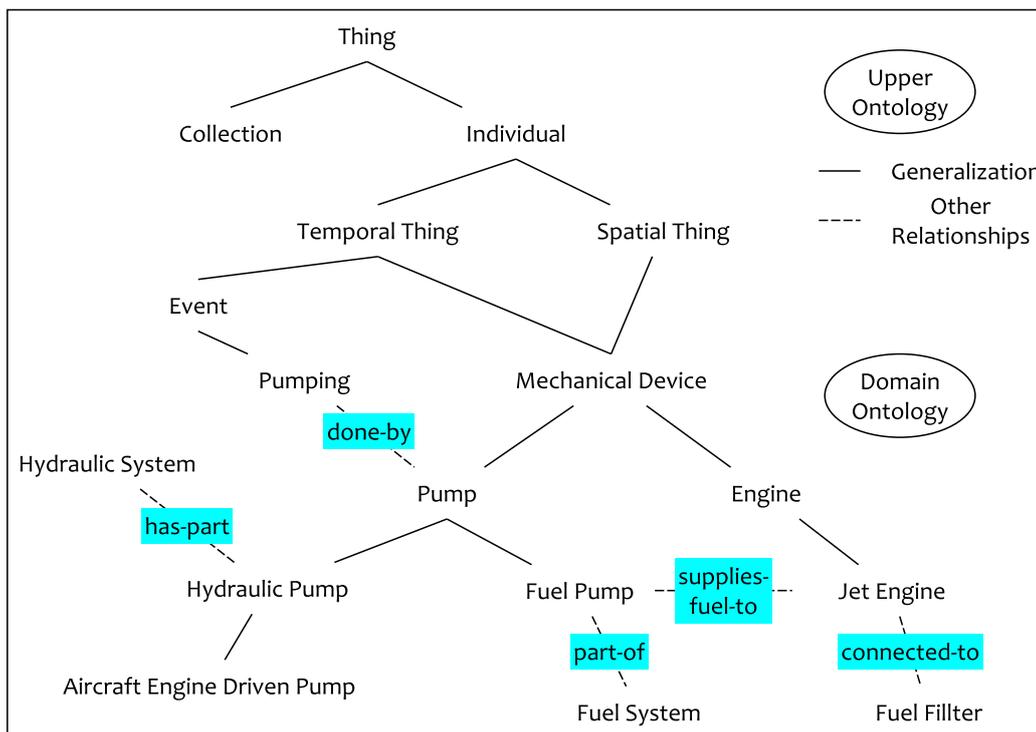


FIGURE 2.19 : Ontologie supérieure et Ontologie du domaine (Uschold et Gruninger 2004).

entités (concepts, relations). Par contre, pour les ontologies "heavyweight", presque toutes les entités sont accompagnées de nombreux axiomes qui contraignent leurs utilisations et soutiennent des raisonnements.

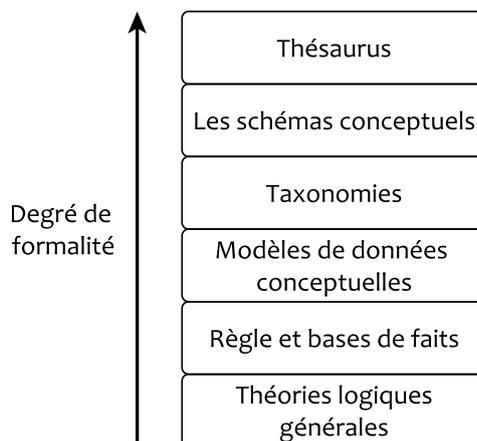


FIGURE 2.20 : Classification des ontologies par le degré de formalité.

Dans notre approche, nous proposons un environnement qui permet de relier des ontologies d'applications construites sur la base d'une ontologie de domaine et de recueil des connaissances. Ces ontologies sont des "lightweight" où seulement des concepts et des relations sont décrits.

2.3.4.5 Langages de représentation des ontologies

Les langages d'ontologie sont les langages utilisés pour exprimer les ontologies. Ils permettent d'encoder des connaissances du domaine spécifique et incluent souvent des règles de raisonnement qui soutiennent le traitement de ces connaissances. En général, les langages d'ontologie peuvent être classifiés par la syntaxe ou la structure.

2.3.4.5.1 Classification par syntaxe

Du point de vue syntaxique, les ontologies peuvent être représentées en utilisant de la logique de premier ordre, un langage de balisage ou un "langage à cadres".

- **Syntaxes traditionnelles** : Dans les années 90s, les langages d'ontologie ont été basés principalement sur les cartes et les logiques de premier ordre (CyC (Lenat et Guha 1989), Ontolingua (Farquhar et al. 1996), etc.).

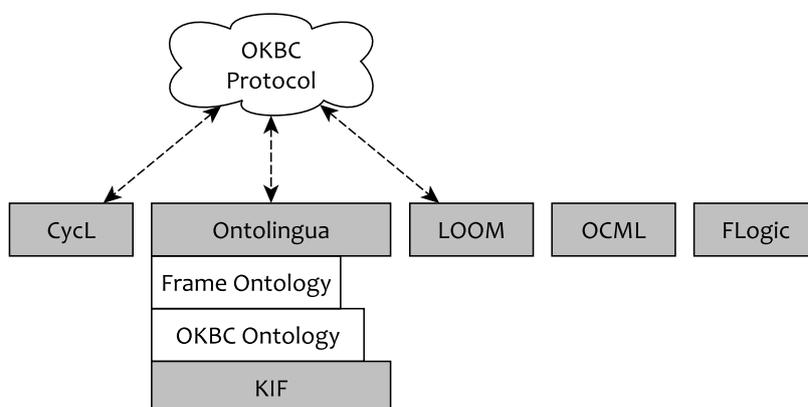


FIGURE 2.21 : Les langages d'ontologie traditionnels (Gómez-Pérez et al. 2004).

- **Langage de balisage** : Actuellement, d'autres techniques de représentation des connaissances basées sur la logique de description (Baader et al. 2003) ont été utilisés pour construire des ontologies et de nouveaux langages logiques de description comme OIL (Fensel et al. 2001), DAML + OIL²⁸, et OWL (Dean et al. 2004) sont apparus dans le contexte du Web sémantique (Gómez-Pérez et al. 2004).

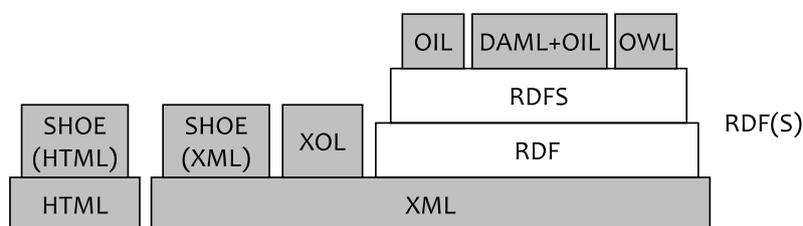


FIGURE 2.22 : Langage d'ontologie de balisage (Gómez-Pérez et al. 2004).

2.3.4.5.2 Classification par structure

Uschold (Uschold 1996) a classifié les langages d'ontologie entre quatre catégories en fonction du degré de formalité :

28. <https://www.w3.org/TR/daml+oil-reference>

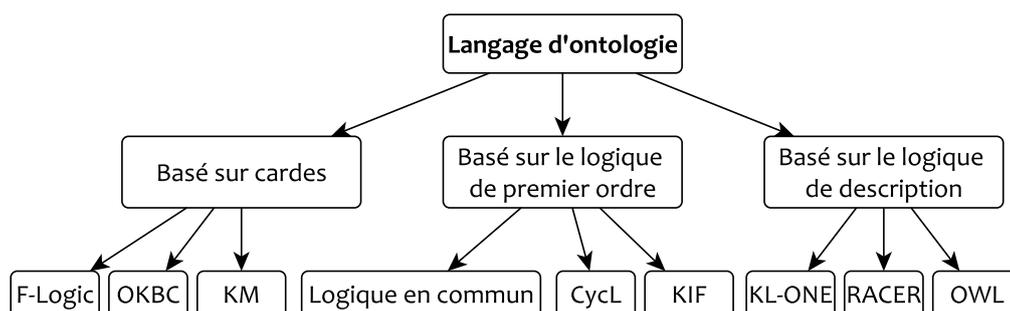


FIGURE 2.23 : Classification des langages d'ontologie par structure.

- **Fortement informel** : Ontologie exprimée en langage naturel. Cette méthode est appropriée pour la communication entre humains. Notons que selon la définition de Studer (Studer et al. 1998), une ontologie très informelle ne serait pas une ontologie puisqu'elle n'est pas "machine-traitable".
- **Semi-informel** : Ontologie exprimée en un langage naturel restreint et structuré.
- **Semi-formel** : Ontologie exprimée dans un langage artificiel et formellement défini (Ontolingua (Farquhar et al. 1996), OWL (Dean et al. 2004), etc.)
- **Rigoureusement formel** : Ontologie exprimée dans un langage qui fournit des termes méticuleusement définis avec une sémantique formelle, des théorèmes et des preuves de propriétés telles que la solidité (*justesse*) et l'exhaustivité.

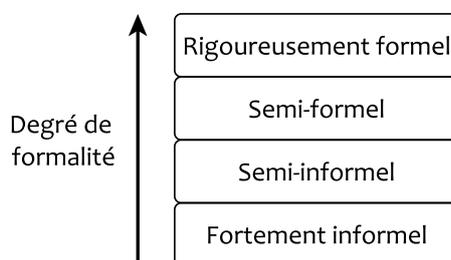


FIGURE 2.24 : Classification des langages d'ontologie par le degré de formalité.

En informatique et intelligence artificielle, les langages d'ontologie sont des langages formels utilisés pour construire des ontologies. Ils permettent l'encodage des connaissances sur des domaines spécifiques et incluent souvent des règles de raisonnement qui soutiennent le traitement de ces connaissances. Les langages d'ontologie sont généralement des langages déclaratifs, sont presque toujours des généralisations de langage orienté cadre et sont généralement basés soit sur une logique de premier ordre, soit sur une logique de description²⁹.

Nous exprimons dans notre environnement l'ontologie sous deux aspects : une représentation semi-informelle visuelle pour les acteurs et une description semi-formelle avec OWL exploitant la puissance de l'inférence pour générer des requêtes.

29. https://en.wikipedia.org/wiki/Ontology_language

2.3.4.6 Méthodologies de construction des ontologies

Plusieurs approches permettant de construire une ontologie sont proposées dans la littérature (Wache et al. 2001). En 1989, Lenat (Lenat et Guha 1989) ont proposé des étapes générales et des points intéressants sur le développement des ontologies dans le cadre du projet Cyc³⁰.

Quelques années plus tard, sur la base de l'expérience acquise dans le développement de l'Ontologie d'Entreprise (Uschold et King 1995) et de l'ontologie du projet TOVE (TOronto Virtual Enterprise) (Grüninger et al. 1995) dans le domaine de la modélisation des entreprises, lors de la 12ème Conférence Européenne pour l'Intelligence Artificielle (ECAI'96), Bernaras (Bernaras et al. 1996) a présenté une méthode pour construire une ontologie dans le domaine des réseaux électriques dans le cadre du projet Esprit KACTUS (KACTUS 1996).

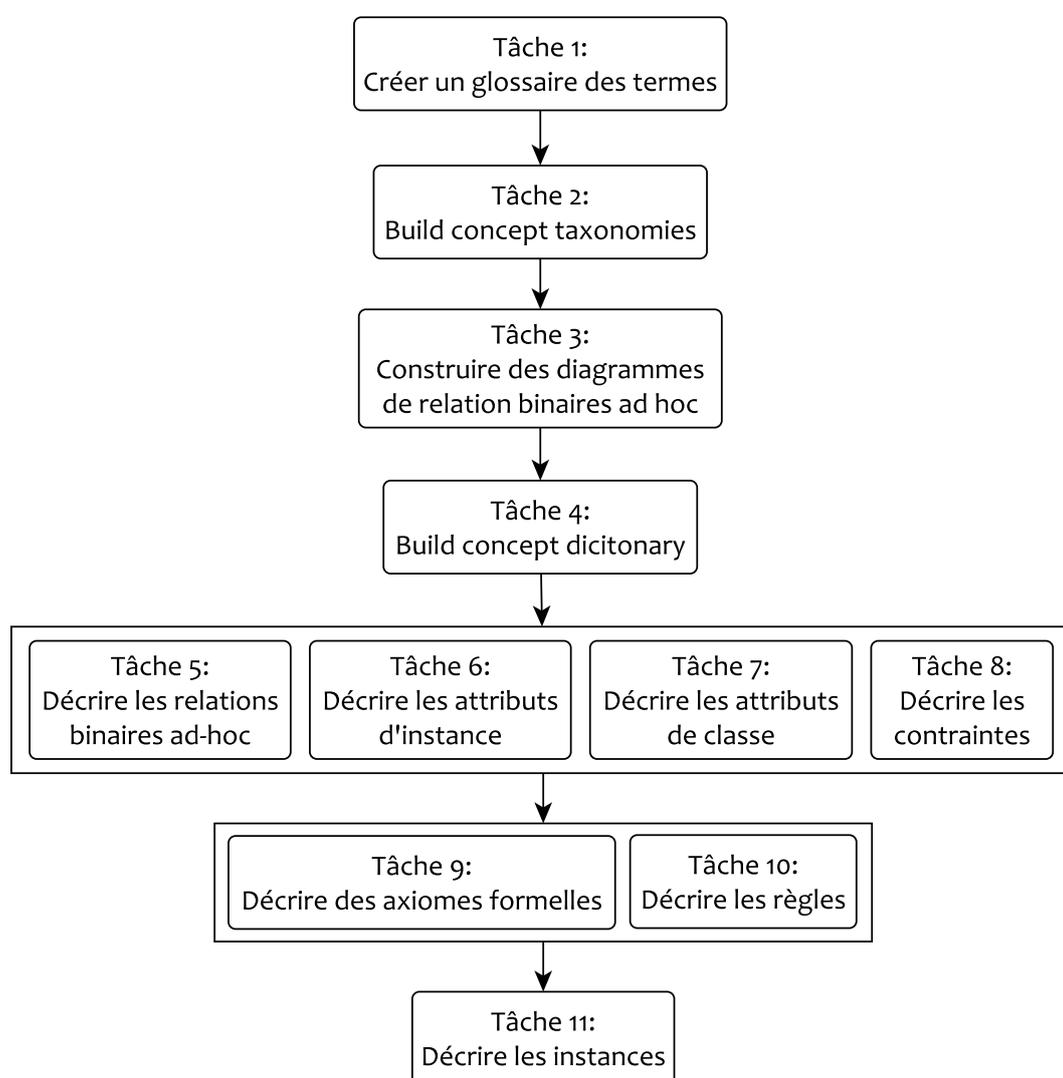


FIGURE 2.25 : Tâches de construction de l'ontologie de la méthodologie METHONTOLOGY (Gómez-Pérez et al. 2004)

La méthodologie METHONTOLOGY (Gómez-Pérez et al. 1996, Fernández-López et al. 1997)

30. <http://www.cyc.com/>

est apparue simultanément et a été étendue dans d'autres articles (Gómez-Pérez 1998, Lopez et al. 1999). Cette méthodologie permet de construire des ontologies au niveau du savoir. Il a des racines dans les principales activités identifiées par le processus de développement logiciel³¹ et dans les méthodologies de l'ingénierie du savoir (Waterman 1986).

La Figure 2.25 illustre l'ensemble de tâches de la structuration. Cette figure met l'accent sur les composantes de l'ontologie (*concepts, attributs, relations, constantes, axiomes formels, règles et instances*) attachées à chaque tâche. Une fois ces éléments définis, des axiomes et des règles formelles sont utilisés pour la vérification des contraintes et pour inférer des valeurs pour les attributs. Le Tableau 2.1 décrit en détail la mission et les résultats de chaque tâche.

Tâche 01	Construire le glossaire des termes qui identifie l'ensemble de termes à inclure dans l'ontologie, leur définition du langage naturel et leurs synonymes et acronymes.
Tâche 02	Construire des taxonomies conceptuelles pour classer les concepts. La sortie de cette tâche pourrait être une ou plusieurs taxinomies où les concepts sont classés.
Tâche 03	Construire des diagrammes de relation binaires ad hoc pour identifier les relations ad hoc entre les concepts de l'ontologie et les concepts d'autres ontologies.
Tâche 04	Construire le dictionnaire conceptuel, qui inclut principalement les instances de concept pour chaque concept, leurs attributs d'instance et de classe, et leurs relations ad hoc.
Tâche 05	Décrire en détail chaque relation binaire ad hoc qui apparaît sur le diagramme de relation binaires ad hoc et sur le dictionnaire de concepts. Le résultat de cette tâche est la table de relations binaires ad hoc.
Tâche 06	Décrire en détail chaque attribut d'instance qui apparaît sur le dictionnaire de concepts. Le résultat de cette tâche est la table où les attributs d'instance sont décrits.
Tâche 07	Décrire en détail chaque attribut de classe qui apparaît sur le dictionnaire de concepts. Le résultat de cette tâche est la table où les attributs de classe sont décrits.
Tâche 08	Décrire en détail chaque constante et produire une table constante. Les constantes spécifient des informations relatives au domaine de la connaissance, elles prennent toujours la même valeur et sont normalement utilisées dans les formules.
Tâche 09	Les axiomes formels sont définis.
Tâche 10	Les règles sont définies pour la vérification des contraintes et pour inférer des valeurs pour les attributs.
Tâche 11	Des informations supplémentaires sur les instances sont décrites.

TABLE 2.1 : Description détaillée de chaque tâche de méthodologie METHONTOLOGY (Gómez-Pérez et al. 2004)

En 1997, une nouvelle méthode a été proposée pour construire des ontologies basées sur l'ontologie SENSUS (Swartout et al. 1997). Quelques années plus tard, la méthodologie On-To-Knowledge est apparue dans le projet portant le même nom (Staab et al. 2001) . En 2006, Ismail (Ismail M. A 2006) a présenté la construction de l'ontologie comme un processus itératif et comprend cinq étapes : **Conception, Développement, Intégration, Validation, Feedback** et **Itération** (Figure 2.26).

Ces deux démarches peuvent être utilisées pour construire les ontologies soutenues dans

31. <http://ieeexplore.ieee.org/document/511093/>

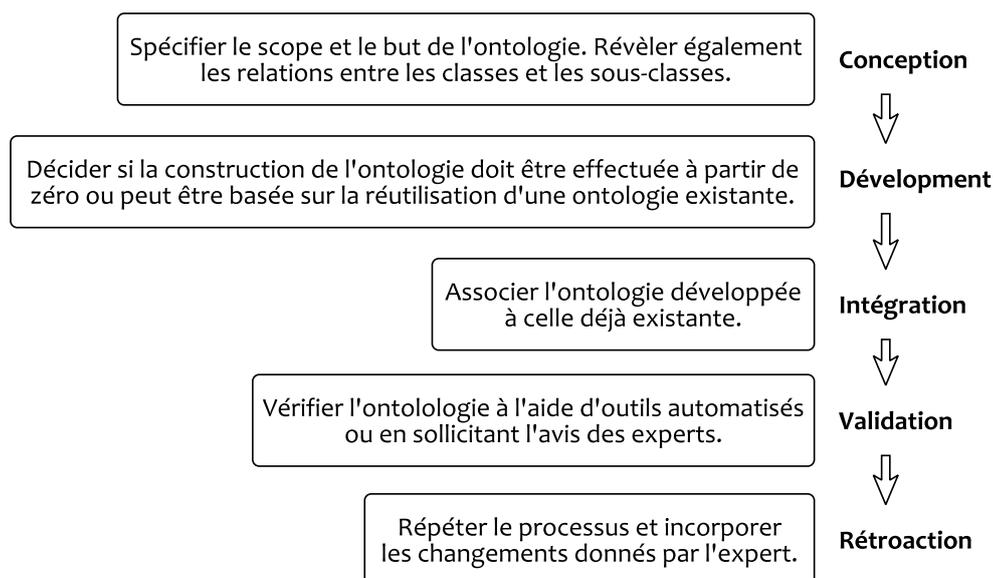


FIGURE 2.26 : Le processus de construction des ontologies proposé par (Ismail M. A 2006).

notre environnement. En revanche dans notre application dans le domaine de la bio-imagerie, nous avons suivi la démarche d'Ismail où une confrontation entre l'ontologie du domaine "ontoNeurolog" et les concepts identifiés lors du recueil des connaissances auprès des acteurs au laboratoire GIN a été effectuée.

2.3.4.7 Conclusion

Dans notre approche, nous choisissons l'ontologie pour capturer et représenter les connaissances du domaine. L'ontologie fournit une représentation formelle des concepts du domaine et toutes les relations possibles entre eux. Deux raisons qui justifient notre choix :

- L'ontologie peut être exprimée dans les langages formels (*RDF* ou *OWL*) qui permettent un fort raisonnement sur les connaissances décrites dans l'ontologie.
- L'ontologie peut être représentée sous forme d'un réseau sémantique qui est facilement compréhensible par les acteurs non-techniciens.

La section suivante présente les principes de la visualisation. Bien que les ontologies peuvent représenter les connaissances de manière explicite, nous trouvons qu'elles devraient être visualisées graphiquement pour avoir une meilleure utilisation. De plus, une représentation visuelle des ontologies est nécessaire pour améliorer la formulation des requêtes des acteurs métiers non-techniciens.

2.4 VISUALISATION

Cette section présente des notions de base et des techniques de la visualisation. Elle a pour objectif de montrer pourquoi la visualisation est nécessaire dans notre approche et de justifier les techniques de visualisation que nous avons choisies pour la représentation visuelle sémantique des données.

"Visualization is about how the human visual system can be inferentially amplified by rendering foundation data in appropriate visual spaces" [\(Marchese et Banissi 2013\)](#)

La visualisation est importante en raison de sa relation avec le système de perception (système cognitif) humain. "Pour la plupart des gens, nous pouvons affirmer que les dépendances et interactions complexes peuvent être plus facilement comprises lorsqu'elles sont illustrées" [\(Bertschi et al. 2013\)](#). Selon Ware [\(Ware 2004\)](#) "la visualisation est justifiée par les capacités visuelles naturelles de l'homme. Nous sommes capables d'assimiler quasi instantanément et sans effort un grand nombre d'informations représentées graphiquement".

Les capacités de la visualisation

- La visualisation est attirante. La mémoire à court terme de l'homme est limitée, mais les organisations visuelles permettent le stockage des informations sensorielles du cerveau et la capacité de décomposer les structures complexes en morceaux est plus facile à gérer [\(Croitoru 2006\)](#).
- Aider l'utilisateur à dépasser les problèmes qui sont dus aux limitations de la mémoire de travail : améliorer la capacité de mémorisation et de la durée pour mémoriser, durant un processus d'acquisition ou de résolution de problèmes (*raisonnement*) [\(Keller et Tergan 2005\)](#).
- Augmenter nos capacités de traitement en visualisant des relations abstraites entre les éléments visualisés [\(Scaife et Rogers 1996\)](#).

Les fonctionnalités supportées par la visualisation : Selon [\(Zhuhadar et al. 2010\)](#), l'analyse visuelle est le pont entre les yeux humains et la machine, elle facilite le processus de :

- i) **Représenter les données d'une manière que le système cognitif humain peut percevoir.**
- ii) **Permettre aux utilisateurs d'interagir avec d'énormes quantités de données facilement et efficacement.**
- iii) **Résumer les données.**
- iv) Aider les utilisateurs à trouver les informations nécessaires le plus rapidement possible.
- v) **Découvrir des connaissances cachées.**

2.4.1 Visualisation des données

"Data visualization is the process of using graphical presentation to represent complex data in a way that provides the viewer with a qualitative understanding of its information contents, turning complicated sets of data into visual insights".

Les données n'existent pas physiquement, étant les résultats directs d'une mesure, elles sont représentées souvent en chiffres. "Les solutions de visualisation de données apportent de la clarté aux données numériques grâce à la représentation visuelle, ce qui contribue à révéler des idées et des tendances qui pourraient autrement passer inaperçues. Le défi consiste à trouver une technique de visualisation adaptée pour mieux comprendre les données"³² [\(Tran 2010\)](#). Keim [\(Keim 1997\)](#) a classifié les techniques de visualisation de données en trois catégories :

32. http://www.infovis-wiki.net/index.php?title=Data_Visualization

Les techniques d'exploration visuelle de données

- Techniques géométriques
- Techniques basées sur les icônes
- Techniques orienté-pixel
- Techniques hiérarchiques
- Techniques graphiques
- Techniques hybrides

Les techniques de distorsion

- Mur en perspective
- Affichage bifocal
- TableLens
- Graphe des vues Fisheye
- Représentation hyperbolique
- Hyperbox

Les techniques dynamiques et interactives

- Cartographie des données à la visualisation
- Projections
- Filtrage (sélection, interrogation)
- Liaison et broissage
- Zoom
- Détails sur demande

2.4.2 Visualisation des informations

"Information visualization is the use of computer-supported, interactive, visual representations of abstract data to amplify cognition" _([Card et al. 1999](#))_

Les données deviennent des informations quand elles sont transformées, transférées et rendues disponibles à l'interprétation par les utilisateurs. L'objectif de la visualisation d'information est de traduire les informations abstraites en représentations visuelles qui sont facilement, précisément et significativement décodées. Deux capacités principales de la visualisation des informations sont donc :

- Favoriser la reconnaissance des structures dans les données abstraites et supporter la recherche d'information ([Card et al. 1999](#)), ([Keller et Tergan 2005](#)).
- Promouvoir un niveau de compréhension plus intuitif, plus profond et susciter de nouveaux points de vue sur les informations visualisées.

Pour visualiser efficacement les informations, les primitives graphiques telles que point, ligne, surface ou volume sont utilisées pour coder l'information par position dans l'espace, la taille, la forme, l'orientation, la couleur, la texture et d'autres indices visuels, connexions et enceintes, changements temporels et transformations de points de vue ([Card et al. 1999](#)).

2.4.3 Visualisation des connaissances

La visualisation de connaissances correspond à *"l'utilisation de représentations visuelles pour améliorer le transfert de connaissances entre au moins deux personnes ou groupes de personnes"* _([Burkhard 2004; 2005](#))_

Visualisation des connaissances : La visualisation de connaissances diffère de la visualisation des informations par la nature de l'objet à visualiser. La visualisation d'information représente tous types d'informations pourtant la visualisation de connaissances traite uniquement des concepts.

Selon (Bertschi et al. 2013), "il existe de nombreuses façons dont les connaissances peuvent être représentées. Normalement, nous représentons les connaissances par des modèles formels composés de concepts, de relations entre elles et de conditions logiques - les ontologies. La visualisation de connaissances utilise les représentations graphiques intuitives pour exprimer et représenter ces concepts et ces relations, etc. La représentation visuelle des connaissances peut être soit statique, comme des esquisses et des diagrammes présentés dans des présentations ou sur des affiches, soit interactive à l'aide de l'affichage par ordinateur et offrant des possibilités de navigation, d'exploration et de manipulation de connaissances".

Critère d'évaluation d'une visualisation de connaissances : Certains critères d'évaluation d'une visualisation de connaissances ont été proposés. Selon Eppler (Eppler 2013), une visualisation de connaissances doit :

- **Capter et dépeindre les connaissances** (idées, expériences, concepts, perspectives, opinions, arguments, etc.).
- Être **visuelle**.
- **Faciliter la communication** (*conversation*) d'un **groupe de personnes**.
- Être **flexible et révisable** pour **adapter** aux éventuels *changements/ modifications*.
- Être **compréhensible et transmissible**.
- Permettre de **découvrir de nouvelles connaissances**.

La représentation des connaissances³³ doit répondre aux besoins d'appréhension des informations et des connaissances en :

- **Navigant à travers l'ensemble de concepts** pour appréhender la sémantique du domaine : La visualisation doit représenter la sémantique du domaine. Les utilisateurs peuvent la comprendre, l'assimiler et l'exploiter en explorant la visualisation.
- **Proposant une vision à plusieurs échelles** : La visualisation devait offrir simultanément une vision globale et une vision particulière/ détaillée dans le même espace.
- **S'adaptant à l'utilisateur** : tous les collaborateurs n'ont pas la même activité et le même niveau d'expertise. Chaque carte doit donc être adaptée à ses utilisateurs.

Dans notre approche, nous privilégions une représentation dynamique permettant une navigation à plusieurs échelles. La visualisation guidée par des ontologies d'applications spécifiques pour chaque métier permet d'offrir des utilisations dédiées aux besoins des utilisateurs.

2.4.4 Visualisation des ontologies

Les ontologies peuvent être implémentées et représentées de différentes manières en fonction de l'utilisation (*communication, partage, raisonnement*) ou du consommateur (humain, machine). Les langages d'ontologie formels ou semi-formels sont traitables par la machine, mais difficilement interprétés par l'humain. Pour l'utilisateur humain, les ontologies devraient être représentées graphiquement et visuellement.

La visualisation des ontologies fournit des supports nécessaires permettant de créer des nouvelles ontologies ou d'exploiter des ontologies existantes . Choisir une méthode et une technique de visualisation appropriées est difficile. Généralement, une ontologie est une hiérar-

33. <http://www.cartographie-semantique.fr/resume/>

chie de concepts qui est enrichie avec des interrelations, des axiomes, des fonctions entre concepts. Chaque concept est attaché à une liste d'attributs et un ensemble d'instances qui évoluent. Les méthodologies de visualisation choisies doivent montrer toutes ces informations en temps réel et d'une manière facilement accessible par l'humain (Katifori et al. 2007).

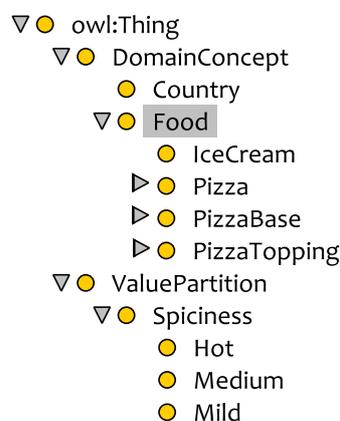


FIGURE 2.27 : Liste indentée des concepts. Les concepts sont organisés taxonomie avec les relations d'héritage "est-un".

Différentes techniques de visualisation des ontologies ont été étudiées. Une liste des outils de visualisation d'ontologies est introduite à [rdf-graph-visualization-tools](#) . Selon Katifori (Katifori et al. 2007), les techniques de visualisation des ontologies peuvent être classées en six catégories : **Liste indentée**, **Noeud - lien et arbre**, **Agrandissable**, **Remplissage d'espace**, **Focus + Contexte** ou **Distorsion 3D** et **Paysages**. Nous résumons ici les cinq premières catégories.

1. **Liste indentée** : La plupart des éditeurs d'ontologies fournissent une liste indentée des concepts ontologiques permettant d'illustrer la hiérarchie des concepts en utilisant les relations d'héritage "est-un". L'avantage de la liste indentée est qu'elle est navigable. La manière dont les concepts sont structurés peut-être facilement appropriée par l'humain. L'utilisateur peut *réduire/ développer* un concept pour avoir une vue générale ou détaillée de la structure des concepts. Cependant, la liste indentée n'est pas très appropriée pour la visualisation des ontologies larges qui contient des milliers de concepts.
2. **Noeud-lien et arbre** : En générale, une ontologie est définie comme un ensemble de triplets (sujet-prédicat-objet), donc elle peut être représentée dans un graphe où "prédicat" est le label de l'arc reliant deux noeuds "**sujet**" et "**objet**". Plusieurs outils ou plugins implémentés dans les éditeurs d'ontologies ont été développés pour fournir une visualisation graphique des ontologies. OntoViz³⁴, IsaViz³⁵, SpaceTree³⁶, OntoTrack³⁷, OntoSphere³⁸, etc. Ces outils utilisent des techniques de visualisation d'information (*couleurs, tailles, forme, interaction, navigation*, etc.) pour représenter clairement dif-

34. <http://protegewiki.stanford.edu/wiki/OntoViz>

35. <https://www.w3.org/2001/11/IsaViz/>

36. <http://www.cs.umd.edu/hcil/spacetree/>

37. <http://www.informatik.uni-ulm.de/ki/ontotrack/>

38. <http://ontosphere3d.sourceforge.net/index.html>

férents composants d'une ontologie : Concept, relation, instance, etc. Quelques outils permettent aux utilisateurs de filtrer les éléments affichés.

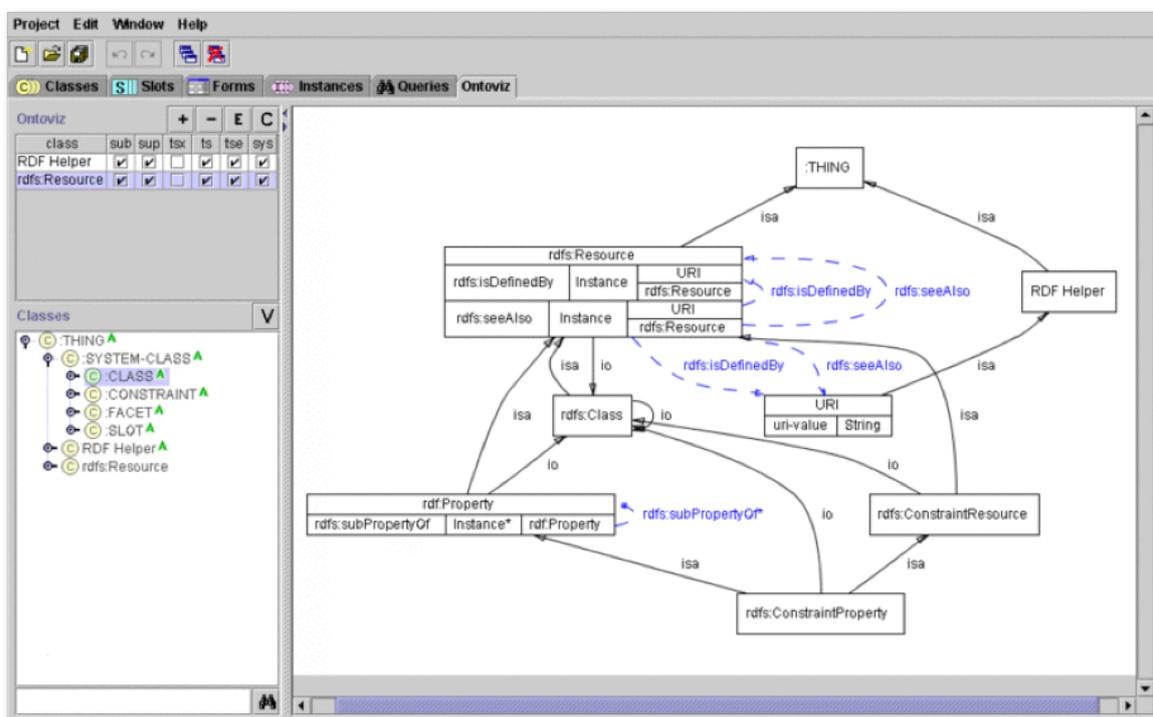


FIGURE 2.28 : Noeud-lien visualisation d'une ontologie dans l'OntoViz³⁹. L'utilisateur peut filtrer les éléments affichés grâce au panel des propriétés à gauche.

Des outils supportant la visualisation des ontologies (OWL et RDF) en ligne sont actuellement développés. La Figure 2.29 illustre une capture d'écran du WebVOWL⁴⁰, une application web pour la visualisation orientée utilisateurs des ontologies. Cette application utilise un graphe dirigé pour représenter l'ontologie. Elle intègre également des techniques d'interaction permettant d'explorer intuitivement des ontologies et de personnaliser la visualisation.

3. **Agrandissable** : Ce type de visualisation représente les concepts ontologiques de tailles différentes en fonction de la position dans la hiérarchie des concepts. Les noeuds représentant les concepts plus proches du concept racine sont plus grands que les feuilles. L'utilisateur peut zoomer sur les noeuds enfants afin de les agrandir et de modifier le niveau d'affichage. Les outils supportant ce mode de visualisation sont Jambalaya⁴¹, CropCircles (Wang et Parsia 2006) et Knoocks (Kriglstein et Wallner 2010). En profitant des points forts des visualisations en graphe et en conteneur, Jambalaya et Knoocks permettent d'exploiter facilement la structure de l'ontologie et d'accéder rapidement aux instances à travers d'une représentation hiérarchique des concepts et une visualisation en graphe des inter-connexions entre concepts.
4. **Remplissage l'espace** : Ce type de visualisation utilise l'espace entier de l'écran pour visualiser les concepts ontologiques. Chaque concept est représenté par un espace (une

39. <http://protegewiki.stanford.edu/wiki/OntoViz>

40. <http://vowl.visualdataweb.org/webvowl.html>

41. <http://protegewiki.stanford.edu/wiki/Jambalaya>

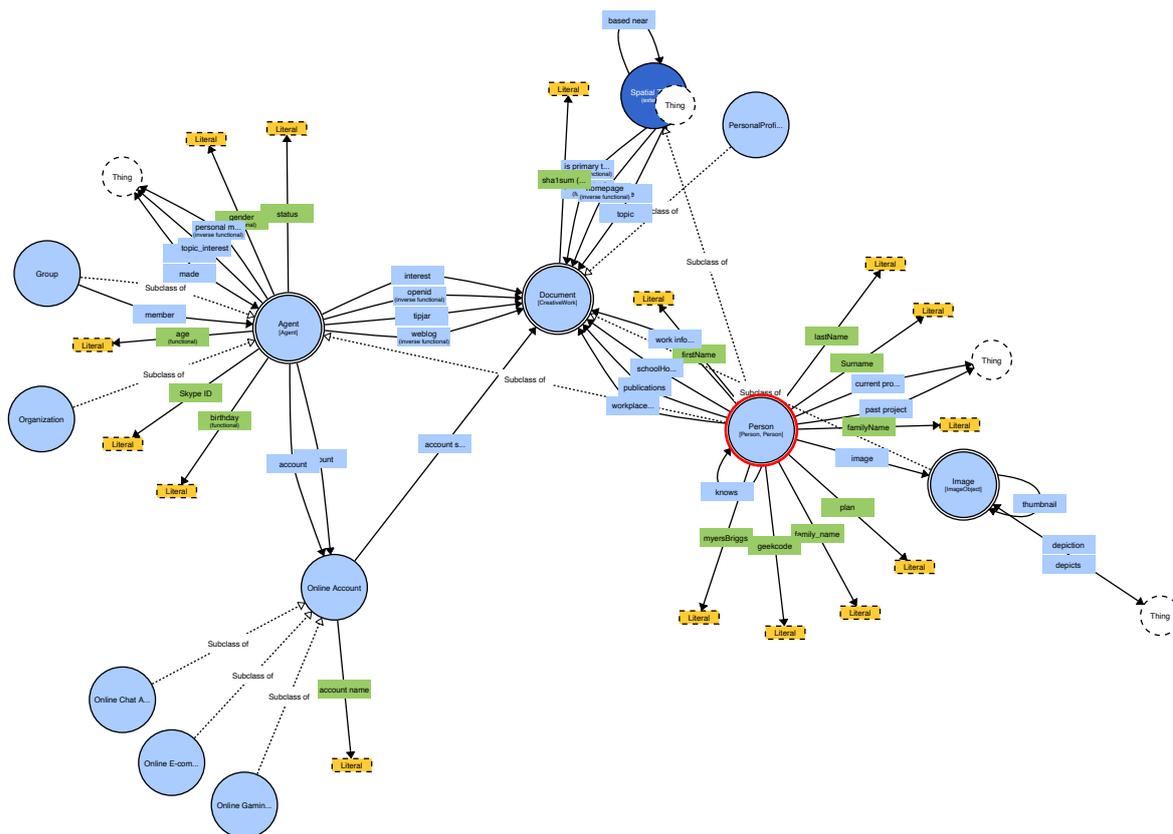


FIGURE 2.29 : WebVOWL ⁴⁰- une application web de visualisation d'ontologies.

division) dont la taille correspond avec le nombre de noeuds qu'il contient. Cette technique de visualisation n'est pas très appropriée pour les ontologies contenant un grand nombre des concepts.

5. **Focus + Contexte** ou **Distorsion 3D** : Ce type de visualisation est basé sur la notion de distorsion de la vue graphique afin de combiner le contexte et le focus (Katifori et al. 2007). Le noeud en focus est représenté au milieu de la visualisation et entouré par les autres noeuds dont la taille décroît jusqu'à ce que le noeud devienne un point. En général, cette visualisation utilise une équation hyperbolique et l'utilisateur doit se concentrer sur un noeud spécifique, afin de l'agrandir. Nous pouvons lister quelques exemples de ce type de visualisation : OntoRama (Eklund et al. 2002), MoireGraphs (Jankun-Kelly et Ma 2003), TGVizTab ⁴², OZONE ⁴³, etc.

42. <http://users.ecs.soton.ac.uk/ha/TGVizTab/>

43. <http://www.cs.umd.edu/hcil/ozone/>

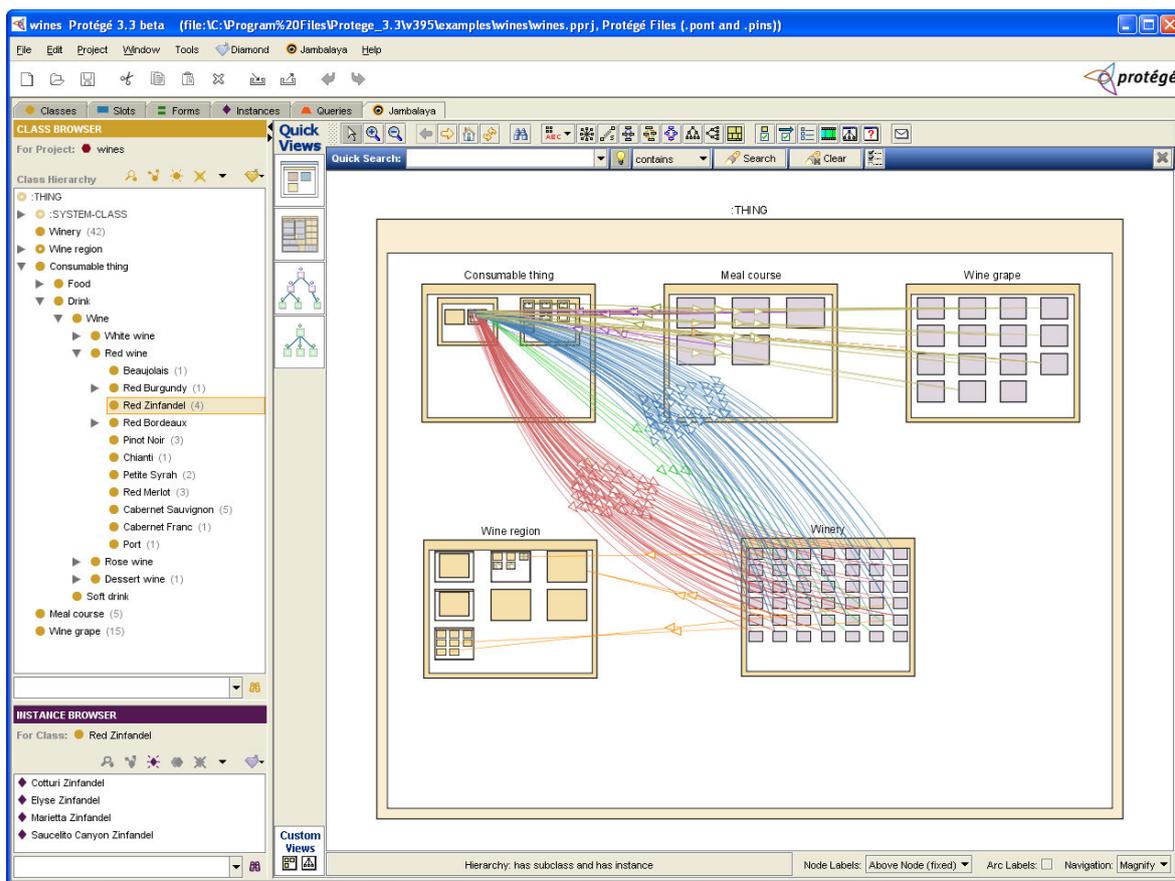


FIGURE 2.30 : Visualisation agrandissable d'ontologie en Jambalaya.

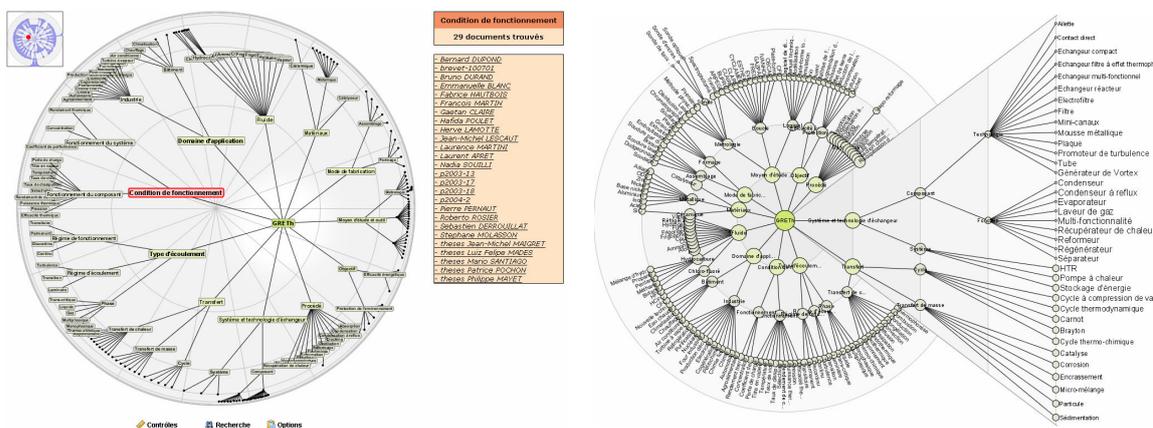


FIGURE 2.31 : EyeTree et RadialTree visualisation d'ontologies (Tricot et Roche 2006).

Ce type de visualisation fournit une vue globale de l'ensemble de concepts ontologiques, mais il ne donne pas une visualisation évidente de la structure des concepts, et devient difficilement compréhensible dans le cas des grandes ontologies.

À la base d'une étude approfondie sur les techniques de visualisations des ontologies, Dudáž et al. (Dudáž et al. 2014) ont introduit 17 fonctionnalités pour une visualisation des ontologies. Chaque technique de visualisation listée ci-dessus a ses propres avantages et inconvénients. Aucune technique satisfait toutes les exigences d'une visualisation des ontologies ou les 17 fonctionnalités proposées par Dudáž et al (Dudáž et al. 2014). Une combinaison

des différentes techniques peut être nécessaire pour avoir un meilleur résultat. Par exemple, la "liste indentée" peut être combinée avec le graphe "noeud-lien" pour la visualisation des ontologies complexes.

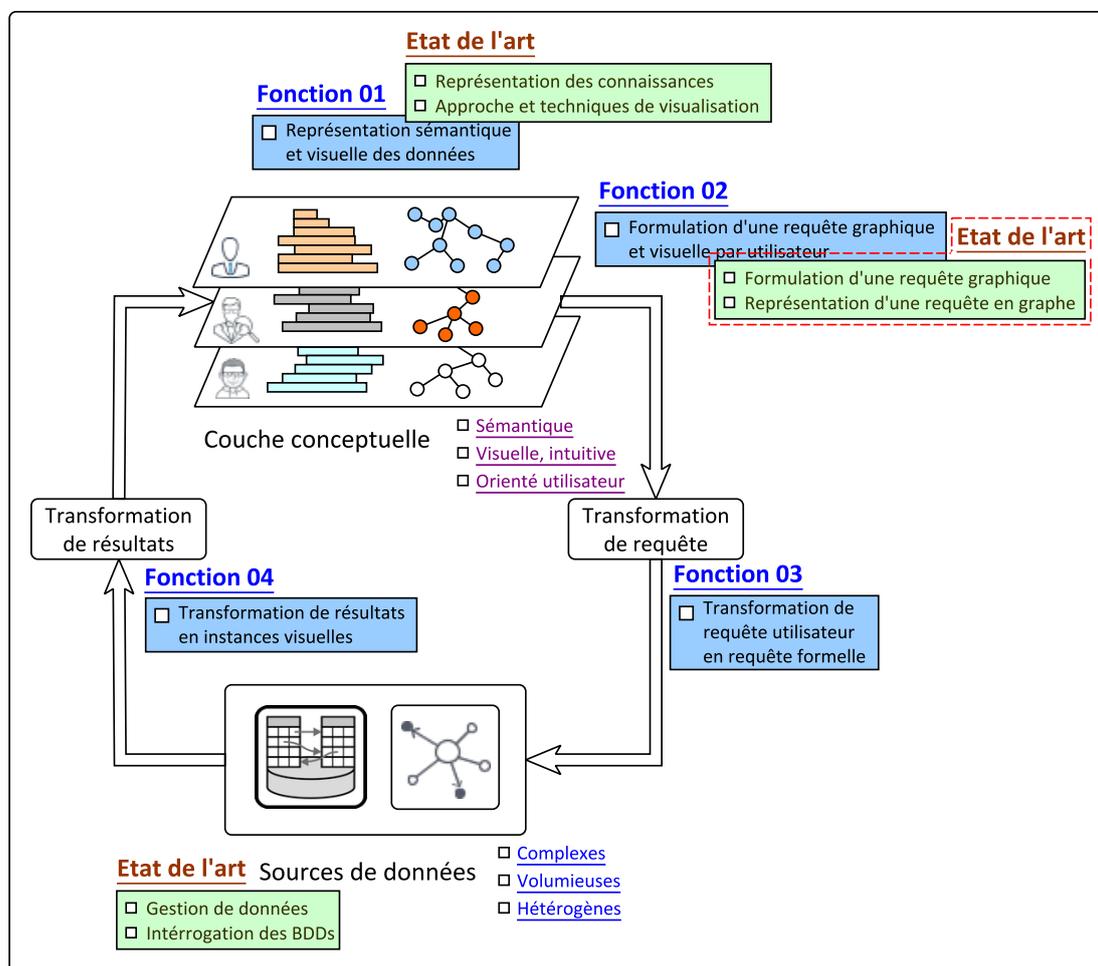


FIGURE 2.32 : L'état de l'art sur les techniques de formulation visuelle d'une requête ontologique.

Discussion

Exploitation de l'ontologie dans notre approche

Après avoir étudié les travaux bibliographiques portant sur la représentation des connaissances et les techniques de visualisation, dans notre approche, nous exploitons les notions suivantes liées à l'ontologie :

- Une construction exploitant une ontologie de domaine et un recueil de connaissances.
- Notre environnement est un support pour des **ontologies d'applications** liées à un domaine précis.
- L'**expression de l'ontologie** est définie selon **deux modes** : semi-informelle accessible aux utilisateurs et semi-formelle exploitant un mécanisme d'inférence.
- La **visualisation de l'ontologie semi-informelle** combine une **liste indentée** avec une représentation sous forme de **graphe** et de **possibilité de focus**.

La construction d'une requête sémantique (requête sur les ontologies) en SPARQL n'est pas

facile si la requête est complexe (longue ou conjonctive). Les avantages des représentations graphiques peuvent faciliter la construction de requêtes conjonctives et longues. La section suivante présente des travaux bibliographiques sur la construction des requête sémantique visuelles (Figure 2.32).

2.5 LA CONSTRUCTION DE REQUÊTE SÉMANTIQUE VISUELLE

Nous supposons que dans cette section, une requête sémantique est une requête ontologique. Elle est formulée par SPARQL. Cependant, formuler proprement une requête ontologique complexe et conjonctive n'est pas facile. Nous citons ici les travaux de Smart (Smart et al. 2008), et Hogenboom (Hogenboom et al. 2010), Balis (Balis et al. 2012) qui ont été centrés sur des méthodes et techniques permettant de créer et modifier une requête sémantique de manière visuelle et graphique.

Hogenboom (Hogenboom et al. 2010) a proposé RDF-GL, un langage de requête graphique à base de SPARQL, utilisé pour requêter des données RDF (Resource Description Framework). Les éléments du RDF-GL sont divisés en trois groupes : Boîtes, ronds et flèches et peuvent être de différentes couleurs (voir Tableau 2.2 en détail). La Figure 2.33 illustre une requête SPARQL et sa représentation en RDF-GL selon les règles de visualisation décrites dans Tableau 2.2. Le RDF-GL est limité à l'expressivité en comparaison avec le SPARQL. Il supporte un sous-ensemble du SPARQL (les requêtes "SELECT" parmi de "SELECT", "AST", "CONSTRUCT" et "DESCRIBE"). Des options de la requête "SELECT" telles que "FROM", "REDUCED", etc. ne sont pas non plus supportées.

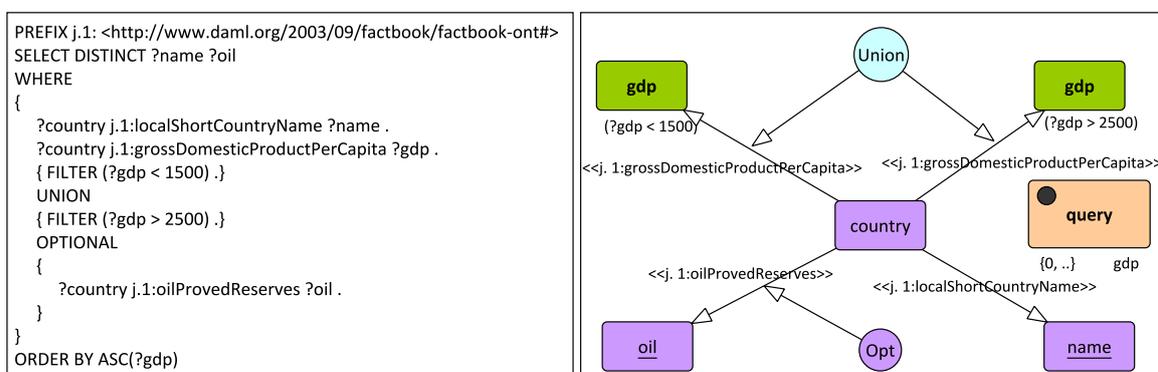


FIGURE 2.33 : Exemple d'une requête SPARQL (à gauche) et sa représentation en RDF-GL (à droite) (Hogenboom et al. 2010).

Subset	Name	Acronym	Color
Box (BOX)	Result box	BR	Orange
	Subject/ object box	BSO	Pink
	Filtered subject/ object box	BFSO	Green
Circle (CRC)	Union circle	CU	Blue
	Optional circle	Co	Purple
Arrow (ARR)	Property arrow	AP	Black
	Optional arrow	AO	Grey
	Union arrow 1	AU1	Yellow
	Union arrow 2	AU2	Red

TABLE 2.2 : Constructions de RDF-GL : formes, propriétés et noms (Hogenboom et al. 2010).

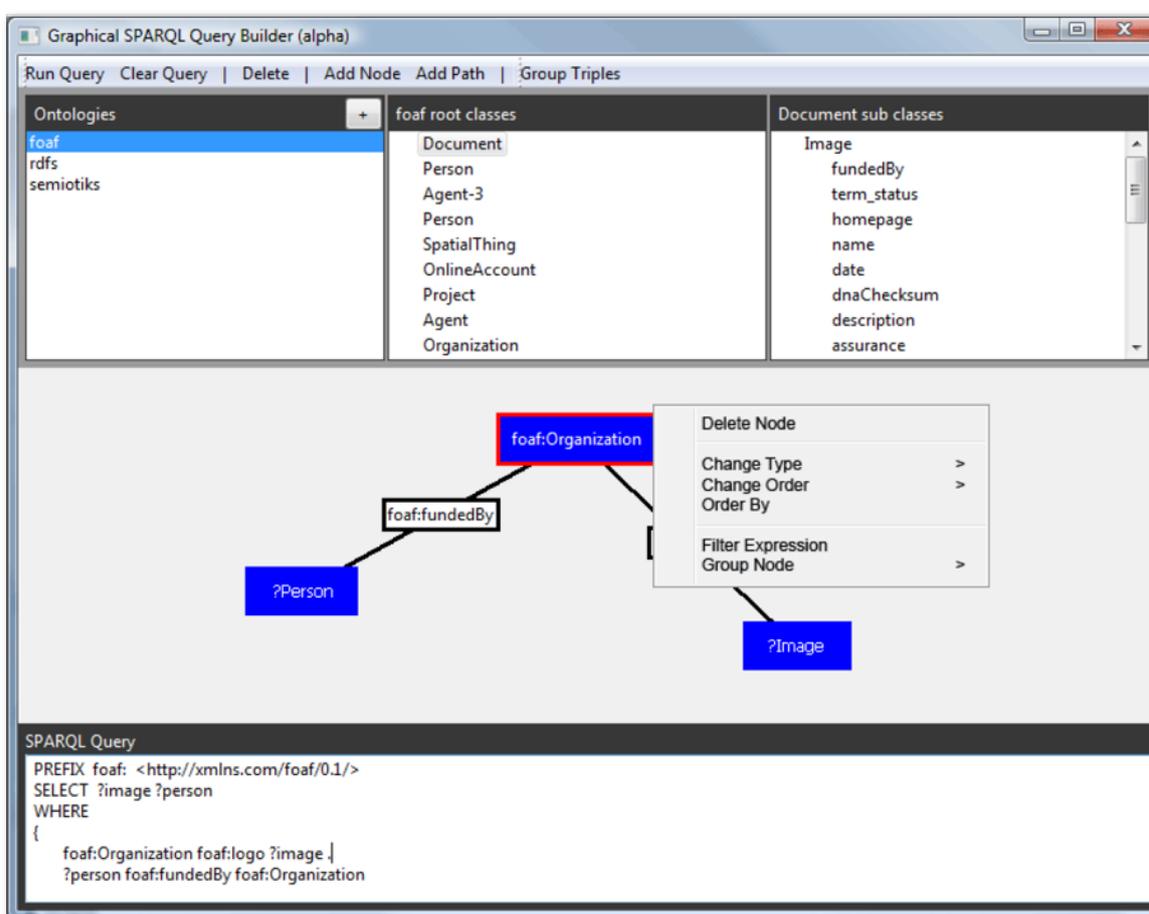


FIGURE 2.34 : Outil graphique supportant la création des requêtes visuelles (Russell et al. 2008).

Smart (Smart et al. 2008) a proposé vSPARQL, un langage de requête visuelle contenant un ensemble de notations graphiques qui support la représentation visuelle d'une requête SPARQL sous la forme d'un graphe. Ce graphe contient essentiellement des noeuds et arcs. Basé sur l'analyse des spécifications du SPARQL, le vSPARQL utilise la couleur pour différencier trois différents types de noeuds correspondant à la nature de la variable qu'ils représentent. Des règles de visualisation sont employées pour représenter des aspects variés d'une requête SPARQL tels que "ordering" ou "filtering". À côté des requêtes "SELECT", le vSPARQL supporte aussi la représentation des requêtes "CONSTRUCT". Cependant, des caractéristiques

du SPARQL ne sont pas supportées telles que les requêtes "ASK", "DESCRIBE" ainsi que les options "DISCTINT", "LIMIT", etc. La Figure 2.34 illustre l'outil graphique NITELIGHT (Russell et al. 2008) développé à base du vSPARQL. A la base de la figure est la représentation textuelle de la requête SPARQL.

Balis et al (Balis et al. 2012) ont utilisé l'ontologie comme le langage de requête familier des acteurs du domaine, à l'aide d'une interface graphique orientée utilisateur. La base de cette approche est un outil de recherche nommé QUaTRO2. Trois caractéristiques principales composent QUaTRO2 : Orienté utilisateur final (experts du domaine sans avoir des expériences techniques), indépendant de domaine (l'outil fonctionne indépendamment de sources de données) et haut niveau d'expressivité de requête (l'aspect "orienté utilisateur final" ne limite pas l'expressivité de requête formulée par l'utilisateur). La Figure 2.35 illustre une requête sémantique à base de l'ontologie et sa représentation graphique. Cependant, la visualisation graphique n'est pas assez intuitive et facilement compréhensible.

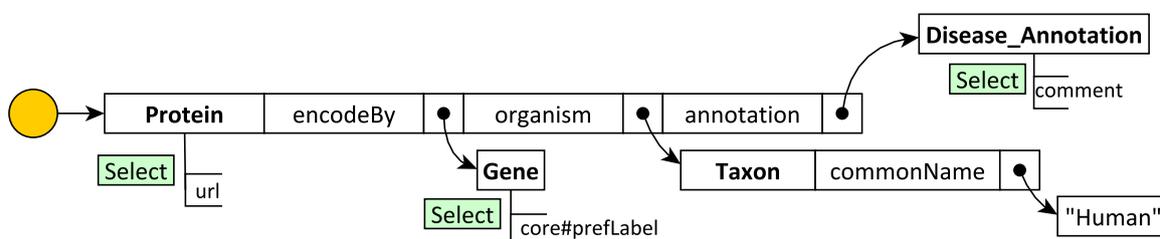


FIGURE 2.35 : Exemple d'une requête sémantique à base de l'ontologie et sa représentation graphique (Balis et al. 2012).

Les travaux présentés ci-dessus ainsi que ceux de e Zainab (e Zainab et al. 2015), Borsje (Borsje et Embregts 2006) ont suivi la même approche : profiter des techniques de visualisation, d'interaction et du Web Sémantique pour faciliter le processus de formulation de requête des utilisateurs finaux. Dans notre approche, nous allons suivre également ces principes en proposant un ensemble de règles de représentation et notations graphiques pour représenter une requête sémantique d'utilisateur qui est éventuellement long, complexe et conjonctive. Ces règles sont définies principalement à base d'une analyse minutieuse des spécifications du langage SPARQL.

Discussion

Les requêtes ontologiques formulées par les acteurs métiers ne peuvent pas être exécutées sur les sources de données. Elles doivent être en fait, transformées en requête formelle (représentée informatiquement en langage de requête traitable par le SGBD). Cette transformation est fait grâce à un ensemble de mappings prédéfinis. Cependant, la construction des mappings entre les ontologies et les sources de données n'est pas facile. Les techniques du Web Sémantique sont parmi les premières techniques liant les ontologies à des volumes importants de données. Elles pourront être utilisées pour surmonter cela. Nous présentons les principes de cette technique dans ce qui suit (Figure 2.36).

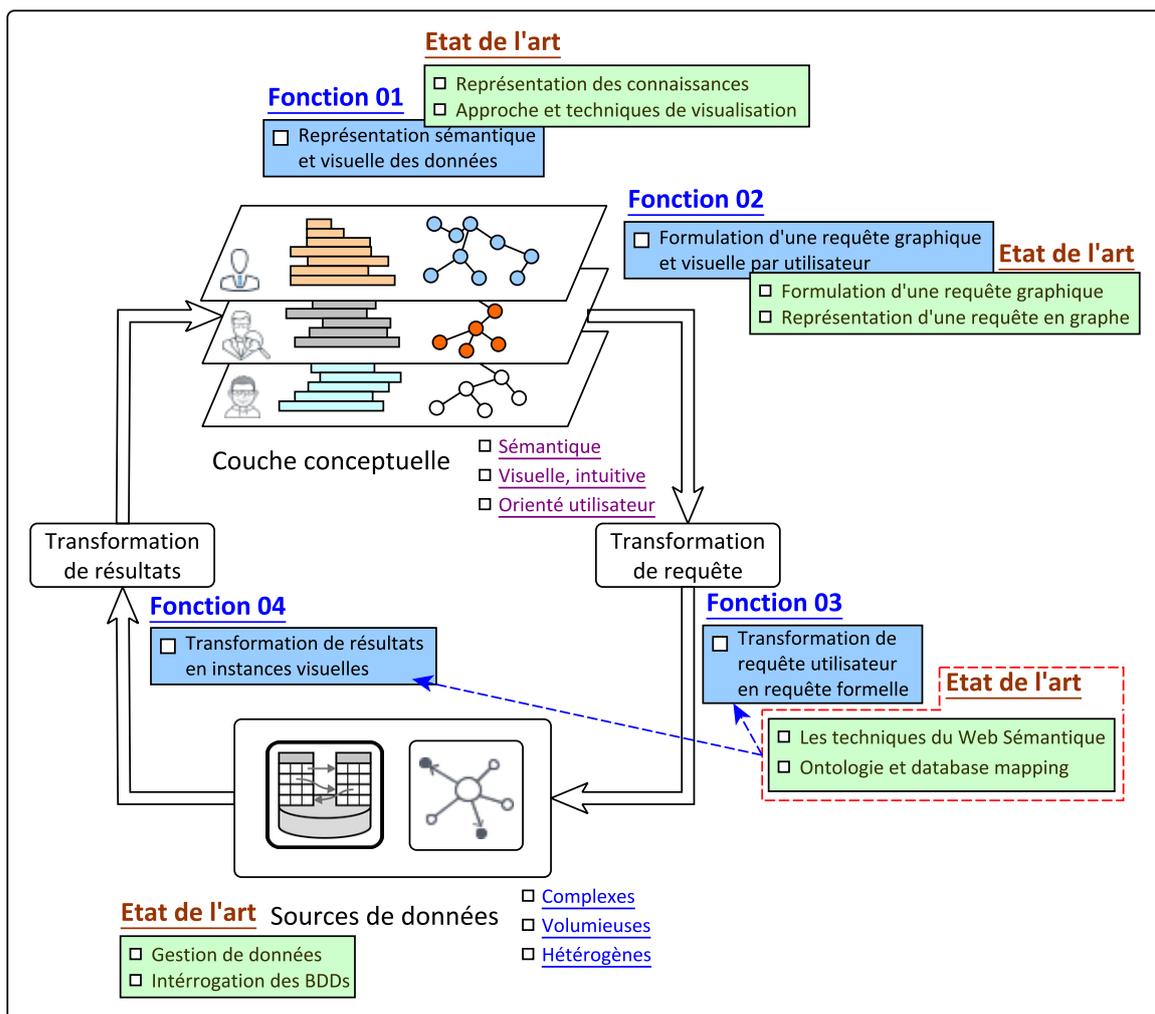


FIGURE 2.36 : L'état de l'art sur les techniques du Web Sémantique et le mapping entre BDDs et ontologies.

2.6 WEB SÉMANTIQUE ET INGÉNIERIE DES CONNAISSANCES

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in co-operation" (Berners-Lee et al. 2001)

Le succès du World Wide Web repose sur sa simplicité. Le caractère restrictif du langage HTTP et du langage HTML a permis aux développeurs de logiciels, aux fournisseurs d'informations et aux utilisateurs de faciliter l'accès à des nouveaux médias (Fensel 2004).

Toutefois, l'explosion de la quantité de contenu web, a conduit à quelques problèmes : Tout d'abord, le contenu devient de plus en plus difficile à localiser. Les requêtes complexes sont difficiles ou impossibles, car cela nécessite généralement l'intégration d'informations provenant de plusieurs sources distribuées (Horrocks 2008). Deuxièmement, les outils de navigation actuels rendent des informations pour la consommation humaine, toutes les informations représentées dans le Web réel ont été écrites en langage naturelle et les machines ne peuvent pas les traiter correctement. Les activités Web se concentrent principalement sur les

activités Machine-à-Humain qui ne sont particulièrement pas bien supportées par les outils logiciels. Les informations sur le Web sont accessibles via la recherche par mots-clés et par le navigateur, les résultats de recherche sur le Web sont très globaux, de faibles précisions et ils sont très sensibles au vocabulaire.

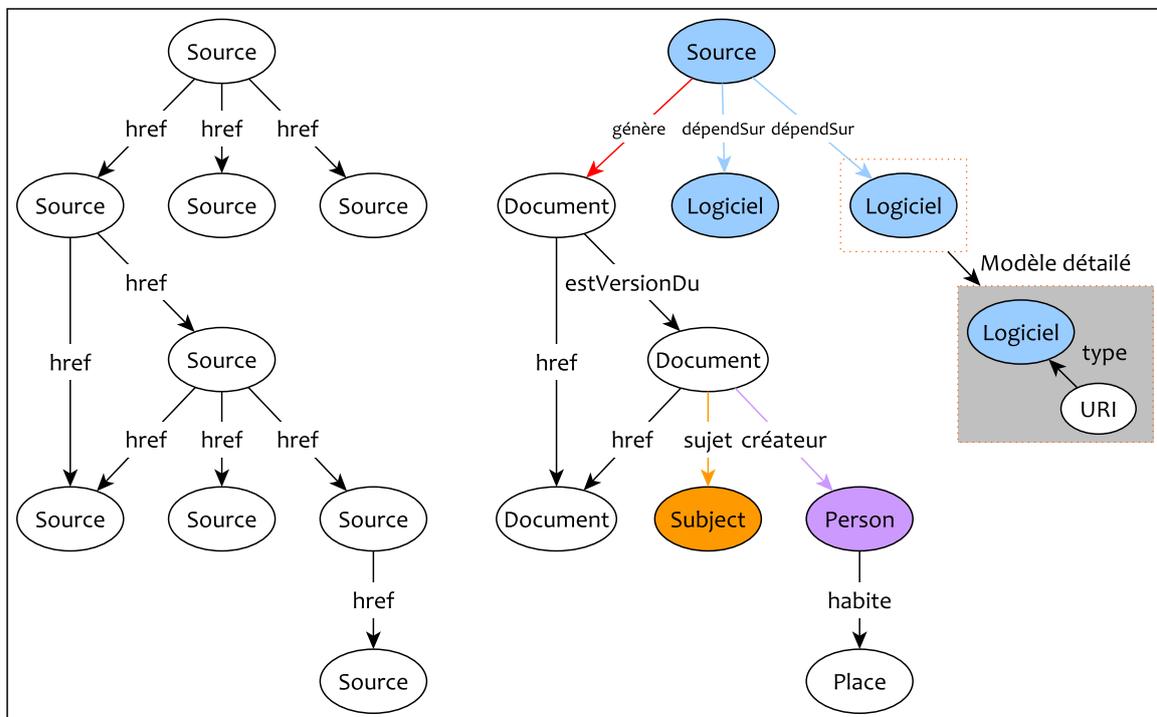


FIGURE 2.37 : Différents types de ressources et de liens dans le Web Sémantique ⁴⁴.

Le Web Sémantique est né comme une extension du Web actuel. L'objectif principal du Web sémantique est de rendre à une vaste gamme d'informations et de services accessibles sur Internet d'être exploités plus efficacement par les humains et par les outils automatisés.

"Semantic Web is actually an extension of the current one in that it represents information more meaningfully for humans and computers alike. It enables the description of contents and services in machine-readable form, and enables annotating, discovering, publishing, advertising and composing services to be automated." ⁴⁵

Le Web sémantique est basé sur la sémantique des données "machine traitables" : La technologie de base est l'Ontologie. C'est une spécification formelle et explicite de conceptualisation qui fournit une compréhension commune et partagée d'un domaine qui peut être communiqué à des personnes des systèmes d'application (Fensel 2004). Les ontologies définissent une sémantique du monde réel permettant de lier le contenu traitable par machine à la signification pour l'homme sur la base d'une terminologie consensuelle (Fensel 2003). La vision du Web sémantique est de rendre des données sur le web sémantique en les représentant en RDF et en les reliant aux ontologies (Bizer 2003, Lv et Ma 2008).

44. <https://www.w3.org/2001/12/semweb-fin/w3csw>

45. <https://www.w3.org/2001/sw/>

2.6.1 L'ontologie et le web sémantique

A partir de quatre caractéristiques principales de l'ontologie : Formalité, Explicité, Consensus et Conceptualité, Grimm (Grimm et al. 2011) a mis en évidence cinq avantages de base et quatre fonctionnalités génériques de l'ontologie (Figure 2.38).

Cinq avantages :

- Standardisation du langage de représentation,
- Standardisation du méta-schéma,
- Provision/ standardisation des connaissances du domaine,
- Langage de modélisation déclaratif riche,
- Sémantique Machine-traitable.

Quatre fonctionnalités génériques :

- **Organisation des connaissances** : Les données et informations rendues dans les systèmes d'information sont informelles, semi-structurées ou non-structurées qui ne sont pas traitables par la machine. Les métadonnées peuvent être utilisées pour annoter et structurer ces ressources. Ces métadonnées peuvent être implémentées dans une ontologie, autrement dit, les annotations des sources de données se réfèrent aux ontologies du domaine.
- **Recherche d'information** : Les ontologies peuvent être utilisées pour fournir un accès sémantique aux ressources Web au lieu des moteurs de recherche traditionnels qui sont basés sur les mots-clés.
- **Intégration** : Les ressources Web sont donc souvent distribuées et hétérogènes, car elles peuvent venir de différents systèmes de gestion des données qui utilisent différents schémas de données. Pour traiter ces ressources ensemble, les ontologies peuvent être utilisées pour mapper différents schémas de données.
- **Traitement formel** : Les ressources web annotées par les ontologies formelles peuvent être traitées automatiquement par la machine. Des moteurs d'inférences peuvent être utilisés pour générer de nouvelles connaissances sur la base de ces ressources ontologiques.

Des combinaisons des avantages de base conduisent différentes fonctionnalités spécifiques (Figure 2.39) qui peuvent être classées en deux catégories : La recherche sémantique et l'Intégration sémantique d'informations.

2.6.2 La recherche sémantique

*"Semantic Search is defined as search for information based on the intent of the searcher and contextual meaning of the search terms, instead of depending on the dictionary meaning of the individual words in the search query"*⁴⁶

Dans un contexte de recherche, la recherche sémantique utilise des technologies sémantiques permettant aux humains ou machines de trouver dans les ressources web des informations

46. <http://www.techulator.com/resources/5933-What-Semantic-Search.aspx>

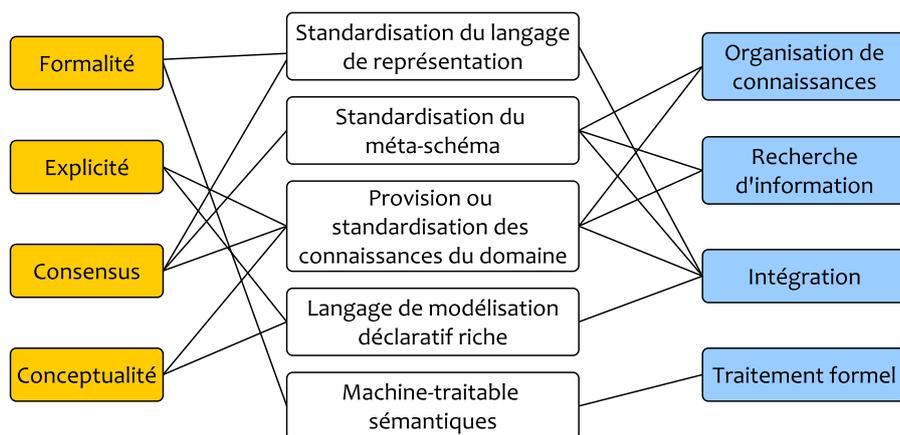


FIGURE 2.38 : Avantages de base (essentiels) et fonctionnalités génériques de l'ontologie (Grimm et al. 2011).

plus appropriés aux besoins d'information (Grimm et al. 2011). Pour assurer une bonne recherche sémantique, le système de recherche doit :

- Capturer le contexte et la sémantique des besoins en information.
- Employer des métadonnées sémantiques ou des connaissances du domaine représentant explicitement et sémantiquement les sources d'information disponibles afin de répondre précisément aux besoins en information.

Différentes approches sont proposées (Lei et al. 2006, Guha et al. 2003, Tran et al. 2007, Zhong et al. 2002). La Figure 2.40 illustre le processus de recherche sémantique basée sur l'ontologie. Ce processus est devisé en cinq composants :

1. Identification du contexte de la requête,
2. Levée d'ambiguïté et reformulation de la requête,
3. Récupération de l'information basée sur les connaissances,
4. Ajout des métadonnées sémantiques,
5. Post-traitement des résultats de la recherche.

La recherche sémantique emploie des technologies sémantiques pour soutenir des agents humains ou automatisés dans le processus de recherche - dans un contexte de recherche donné - d'une ou plusieurs sources d'information (basées sur le Web) afin de satisfaire un besoin en information donné.

2.6.3 L'intégration sémantique d'informations

*"Data integration is the combination of technical and business processes used to combine data from disparate sources into meaningful and valuable information. A complete data integration solution delivers trusted data from a variety of sources"*⁴⁷

L'un des objectifs du Web Sémantique est de permettre aux agents de recherche ("query agents" en Anglais) de traiter et d'intégrer des données provenant de ressources hétérogènes

47. <http://www.ibm.com/analytics/us/en/technology/data-integration/>

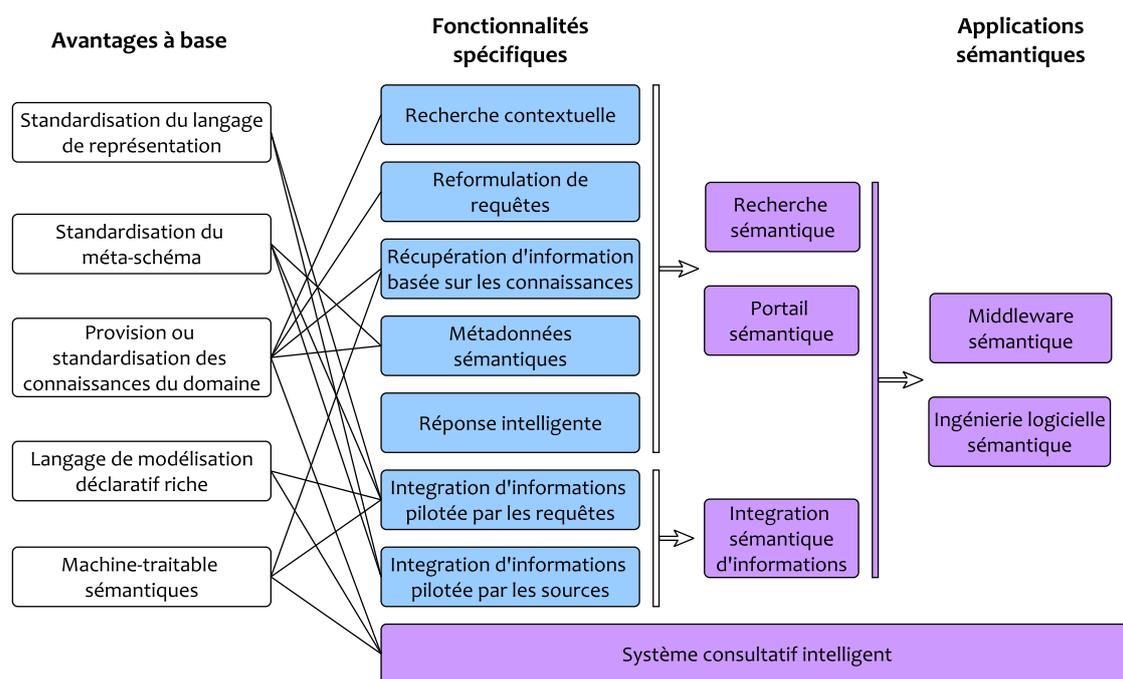


FIGURE 2.39 : Avantages de bases de l'ontologie et les fonctionnalités spécifiques dans le Web Sémantique (Grimm et al. 2011).

au niveau conceptuel (Zhao et Chang 2007). L'intégration des données combine des données résidant dans différentes sources afin de fournir aux utilisateurs une vue unifiée de ces données (Lenzerini 2002). L'intégration des données, vue comme une base pour la réutilisation des données, pour le traitement de requêtes à partir de sources multiples (hétérogènes) et pour les systèmes logiciels interopérables, est une motivation majeure pour le Web Sémantique comme un Web de données (Grimm et al. 2011). Médiateur est un des principales approches du Web Sémantique.

Médiateur ("query-driven" en Anglais) : Médiateur est une approche traditionnelle d'intégration des bases de données hétérogènes. Dans cette approche, la recherche de données se déroule en trois étape⁴⁸.

- Les besoins de données sont exprimés par une requête formalisée par l'utilisateur.
- La requête utilisateur est traduite sous forme de requêtes appropriées à chaque source de données en utilisant l'ensemble de métadonnées prédéfinies.
- Les requêtes traduites sont traitées dans chaque source locale, les résultats sont ensuite intégrés et envoyés à l'utilisateur.

Dans le contexte du Web Sémantique, les schémas des sources de données et le schéma global sont représentés dans les ontologies correspondantes (Figure 2.42). Le mapping entre deux types d'ontologies peut être facilement défini car tous les schémas sont exprimés dans un langage standardisé (Grimm et al. 2011).

La sous-section suivante présente des travaux de la littérature portant sur le mappings entre les BDDs et les ontologies qui joue un rôle indispensable dans le développement du Web

48. <http://www.rsrit.com/blog/2014/05/03/query-driven-approach-data-warehouse/>

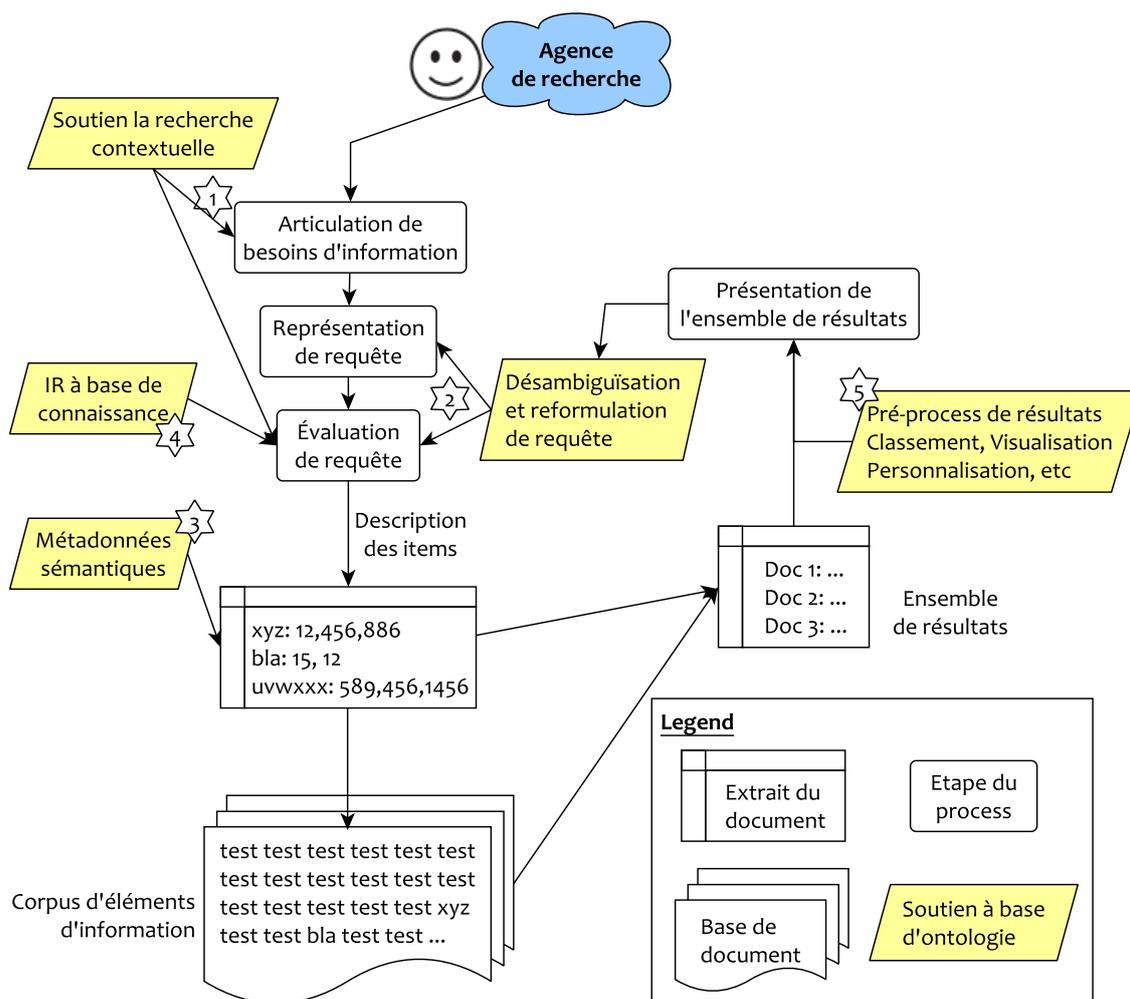


FIGURE 2.40 : Le processus de la recherche sémantique (Grimm et al. 2011).

Sémantique. Nous réutilisons aussi ces techniques pour la transformation de requête dans l'approche proposée

2.6.4 Mapping entre BDDs et ontologies

La vision du Web sémantique est de rendre des données sémantiquement accessibles sur le web en les représentant en RDF ou en les reliant aux ontologies bien acceptées (Bizer 2003, Lv et Ma 2008). "Les bases de données relationnelles (BDDRs) sont considérées comme l'une des solutions de stockage les plus populaires pour divers types de données et elles ont été reconnues comme un facteur clé dans la génération d'énormes quantités de données pour les applications du Web Sémantique" (Spanos et al. 2012). Le Web Sémantique utilise les ontologies pour ajouter à ses ressources une sémantique explicite et partagée. Le problème de combler l'écart entre les BDDRs et les ontologies a attiré l'intérêt de la communauté du Web Sémantique, dès les premières années de son existence. Ce problème est couramment appelé le problème du mapping de la base de données⁴⁹ à l'ontologie (Spanos et al. 2012).

Cette section présente les motivations, les typologies et les langages utilisés pour représenter

49. Base de données = Base de données relationnelle

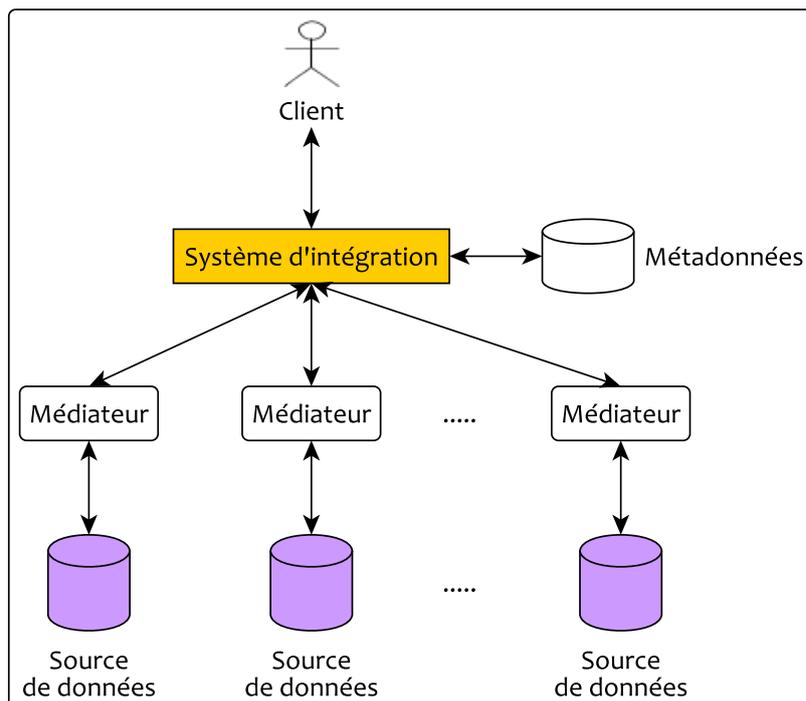


FIGURE 2.41 : Intégration des données (top-down approche) (Saini 2014).

les mappings. Les éléments présentés sont principalement issus des travaux très intéressants sur ce sujet de (Spanos et al. 2012, Zhou et al. 2008, Konstantinou et al. 2008, Hu et Qu 2007, Astrova et Kalja 2006, Bizer 2003, Sequeda et al. 2009). Selon Spanos (Spanos et al. 2012), plusieurs avantages peuvent être obtenus à partir du mapping de la BDD à l'ontologie. Nous résumons ici cinq avantages les plus typiques : **Annotation sémantique des pages Web dynamiques**, **Intégration de bases de données hétérogènes**, **Accès aux données basé sur l'ontologie**, **Génération de masse de données Web sémantiques**, **Apprentissage de l'ontologie**.

1. **Annotation sémantique des pages du Web dynamiques** : Les ontologies peuvent être utilisées pour annoter sémantiquement des ressources web qui sont statiques. Sur le Web dynamique, les données affichées viennent des bases de données en dernière à travers des services web⁵⁰, elles ne peuvent pas être utilisées directement par les applications sémantiques. Une solution possible est d'annoter directement le schéma de la BDD. Cette "annotation" est simplement un ensemble de correspondances entre les éléments du schéma et une ontologie déjà existante qui correspond au domaine du contenu de la page dynamique (Volz et al. 2004). Une fois que de tels mappages sont définis, il serait assez trivial de générer des pages dynamiques dont le contenu est déjà annoté sémantiquement.
2. **Intégration de bases de données hétérogènes** : L'Intégration des BDDs hétérogènes (différents schémas de données, différentes interprétations de données, etc.) est le problème fondamental des systèmes d'information. Dans l'approche traditionnelle, un ou plusieurs schémas conceptuels sont utilisés pour décrire le contenu de chaque source

50. https://fr.wikipedia.org/wiki/Service_web

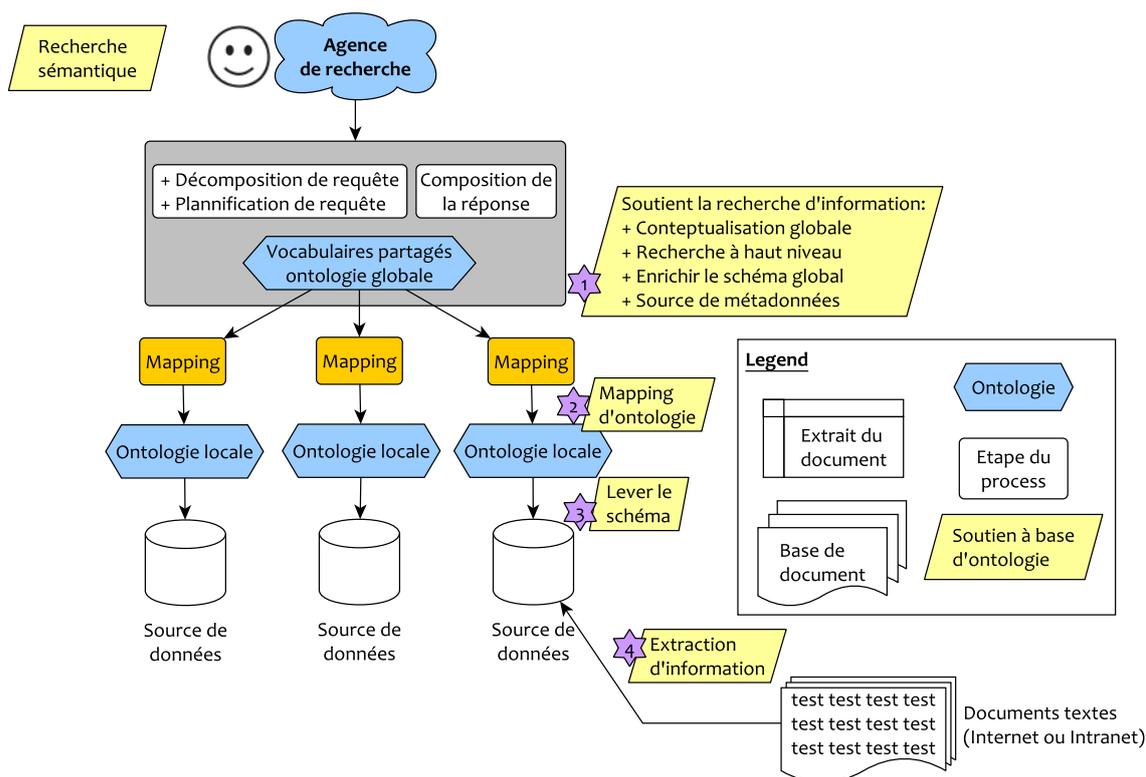


FIGURE 2.42 : Top-down intégration des données basée sur l'ontologie (Grimm et al. 2011).

de données, la requête utilisateur est posée sur un schéma conceptuel global et pour chaque source, un médiateur la reformulera et recoupera les données appropriées (Spanos et al. 2012). Une ou des ontologies peuvent être utilisées à la place des schémas conceptuels, dans ce cas-là, il faut définir l'ensemble de mappings entre ces ontologies et les sources de données.

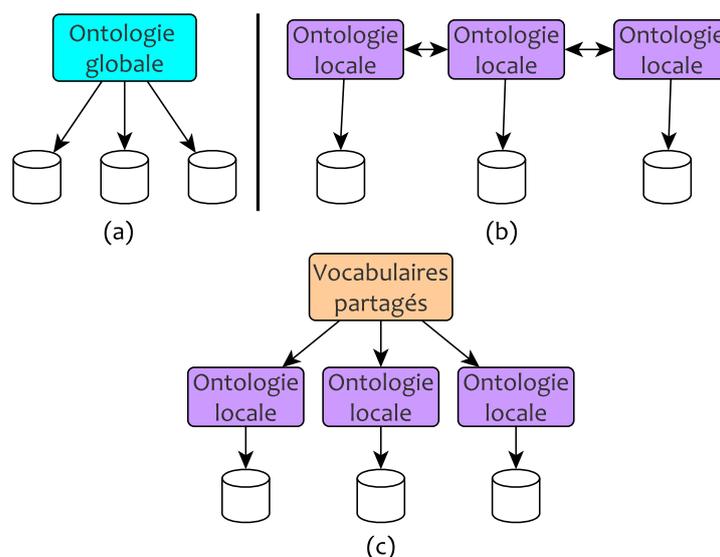


FIGURE 2.44 : Trois approches de l'intégration d'informations basées sur l'ontologie (Wache et al. 2001) : a) approche basée sur une seule ontologie, b) approche basée sur multiples ontologies, c) approche hybride.

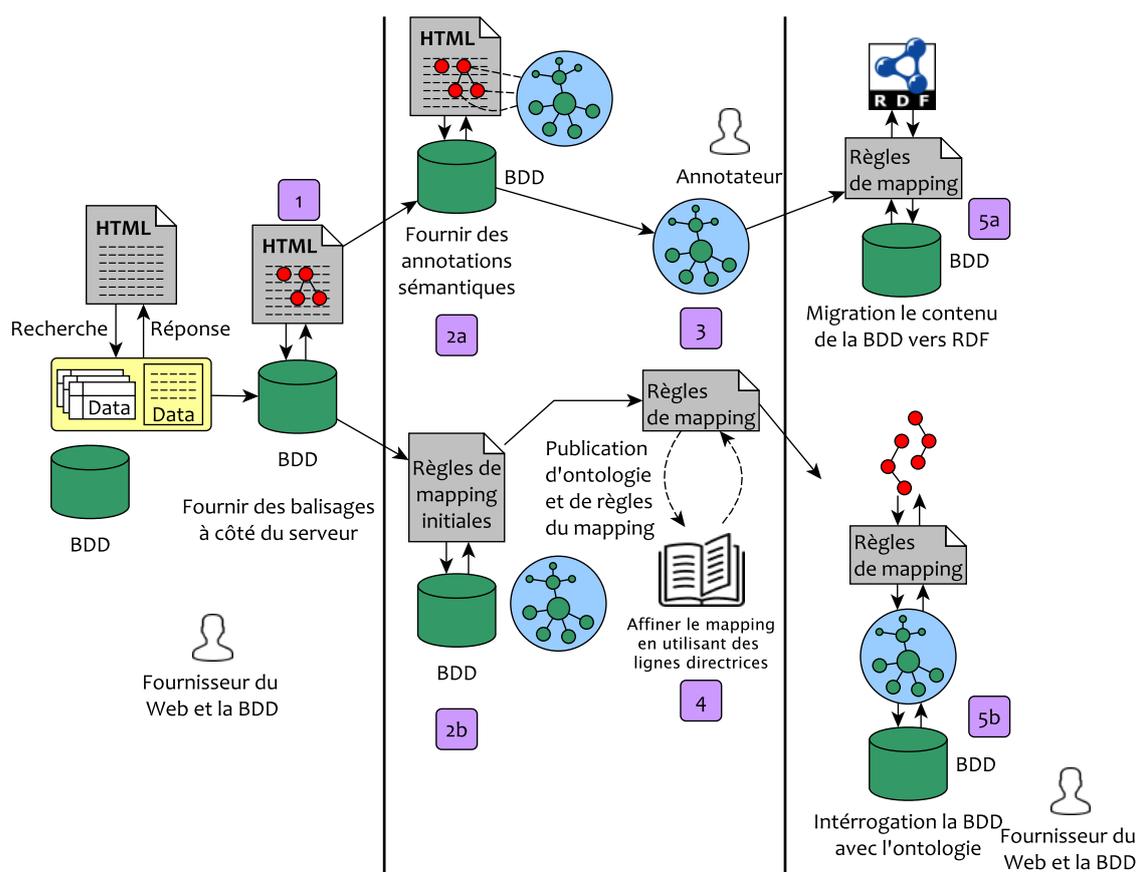


FIGURE 2.43 : Annotation d'un web dynamique (Volz et al. 2004).

Si on définit un mapping comme une correspondance (u, v) , selon Lenzerini (Lenzerini 2002), on peut avoir trois différentes approches :

- **GAV Mapping** : u est une requête conjonctive sur les sources de données, v est le terme ontologique correspondant (Figure 2.44a).
- **LAV Mapping** : u est un terme dans le schéma de données, v est une requête conjonctive correspondante sur l'ontologie (Figure 2.44b).
- **GLAV Mapping** : u et v sont deux requêtes équivalentes sur les sources de données et de l'ontologie (Figure 2.44c).

3. **Accès aux données basé sur l'ontologie** : L'ontologie peut être utilisée dans les SGBDs dans une couche conceptuelle entre les utilisateurs et les sources de données. De tels systèmes permettent aux utilisateurs finaux de requêter eux-mêmes la BDD sans avoir besoin de comprendre la structure et la manière dont les données sont organisées logiquement ou physiquement. L'ontologie fournit une représentation abstraite (et sémantique) de données.

Les utilisateurs finaux formulent leurs besoins d'informations en termes des concepts ontologiques et les requêtes d'utilisateur sont ensuite transformées en requêtes correspondantes sur les sources de données en utilisant l'ensemble de mappings entre ces dernières et l'ontologie (Spanos et al. 2012).

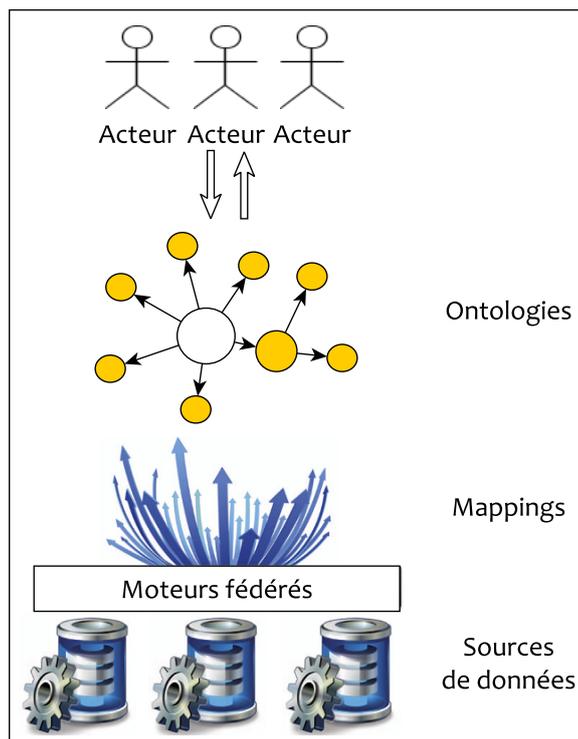


FIGURE 2.45 : Accès aux données basé sur l'ontologie (Calvanese et al. 2016).

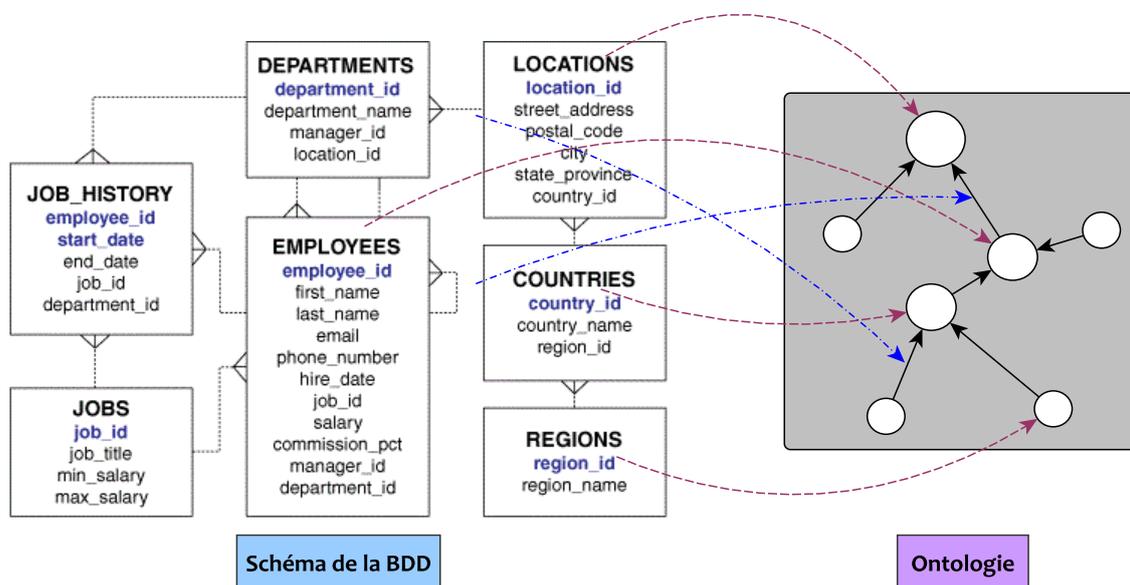


FIGURE 2.46 : Exemple de Construction de l'ontologie à partir du schéma de la BDD.

4. **Génération de masse de données Web sémantiques** : Le Web Sémantique demande une représentation sémantique explicite et partagée d'informations (Zhao et Chang 2007). Comme une solution, les données massives contenues dans les BDDs (notamment les BDDRs) peuvent être transformées (automatiquement) en RDF pour qu'elles puissent être utilisées par les applications du Web Sémantique.
5. **Apprentissage de l'ontologie** : La construction d'ontologies est difficile et fastidieuse. L'apprentissage de l'ontologie (Ontology Learning) vise à construire des ontologies (au-

tomatiquement ou *semi-automatiquement*) à partir des sources déjà existantes (*non* ou *semi-structurés documents, site web, etc.*) (Lehmann et Voelker 2014). Car les BDDs contiennent des données structurées et leurs schémas (*logique* et *conceptuel*) sont construits soigneusement en suivant des standards bien connus (*UML* ou *Modèle ER*), elles constituent des sources importantes et fiables pour la construction de l'ontologie du domaine.

Les technologies du Web Sémantique présentées ci-dessus permettent de justifier l'utilisation de l'ontologie dans notre approche. Tandis que, les succès des applications du Web Sémantique aident à consolider la faisabilité de la méthodologie proposée.

2.7 CONCLUSION

Dans ce chapitre nous avons présenté des travaux bibliographiques très divers : des principes de base des BDDs, l'interrogation des BDDs, la représentation des connaissances, les principes de la visualisation, la formulation visuelle d'une requête sémantique et les technologies du Web Sémantique. Nous constatons que la plupart des BDDs actuelles demandent un certain nombre de compétences techniques sur la structuration de données stockées et un minimum de compréhension sur les langages de requêtes formelles pour accéder aux données. Cela rend difficile l'accès aux données des utilisateurs finaux (les *acteurs métiers non-IT*, les *utilisateurs occasionnels*, les *utilisateurs de différentes disciplines*, etc.). Des travaux de recherche permettant une interrogation facile de données sont proposés : systèmes d'interrogation visuelle, recherche par mots-clés/ langage naturelle, recherche sémantique, etc. Parmi ces travaux, la recherche sémantique à base d'ontologies (OBDA) apporte une solution très prometteuse pour améliorer l'accès aux données des acteurs métiers. Cependant, il manque d'une démarche générique pour mettre en place ce type de système de requête. Différents éléments sont étudiés pour atteindre cet objectif :

1. La représentation sémantique des connaissances du domaine proches aux acteurs métiers est à base des ontologies.
2. Les technologies du Web Sémantique fournissent des bases solides pour la construction des mappings entre les ontologies et les sources de données. Cependant, il manque une approche générique et réutilisable permettant de construire ces mappings.
3. Une représentation visuelle permet aux utilisateurs de reconnaître facilement non seulement les informations représentées, mais aussi les informations utiles, les relations et les structurations cachées. La formulation des requêtes sémantiques (ontologiques) peut être facilitée en utilisant des représentations visuelles (conventions graphiques et règles de transformations).

La Figure 2.47 illustre le positionnement des travaux de la littérature par rapport au processus idéal présenté à la fin du chapitre 1. Dans le chapitre suivant, nous allons présenter en détail une méthodologie permettant une multi-utilisation aisée et sémantique des données entre l'ensemble de acteurs métiers. Cette méthodologie utilise l'ontologie pour fournir un accès sémantique aux données. Une fois cette approche proposée et justifiée, nous nous concentrons sur la visualisation de l'ontologie et de son utilisation pour faciliter la formulation de

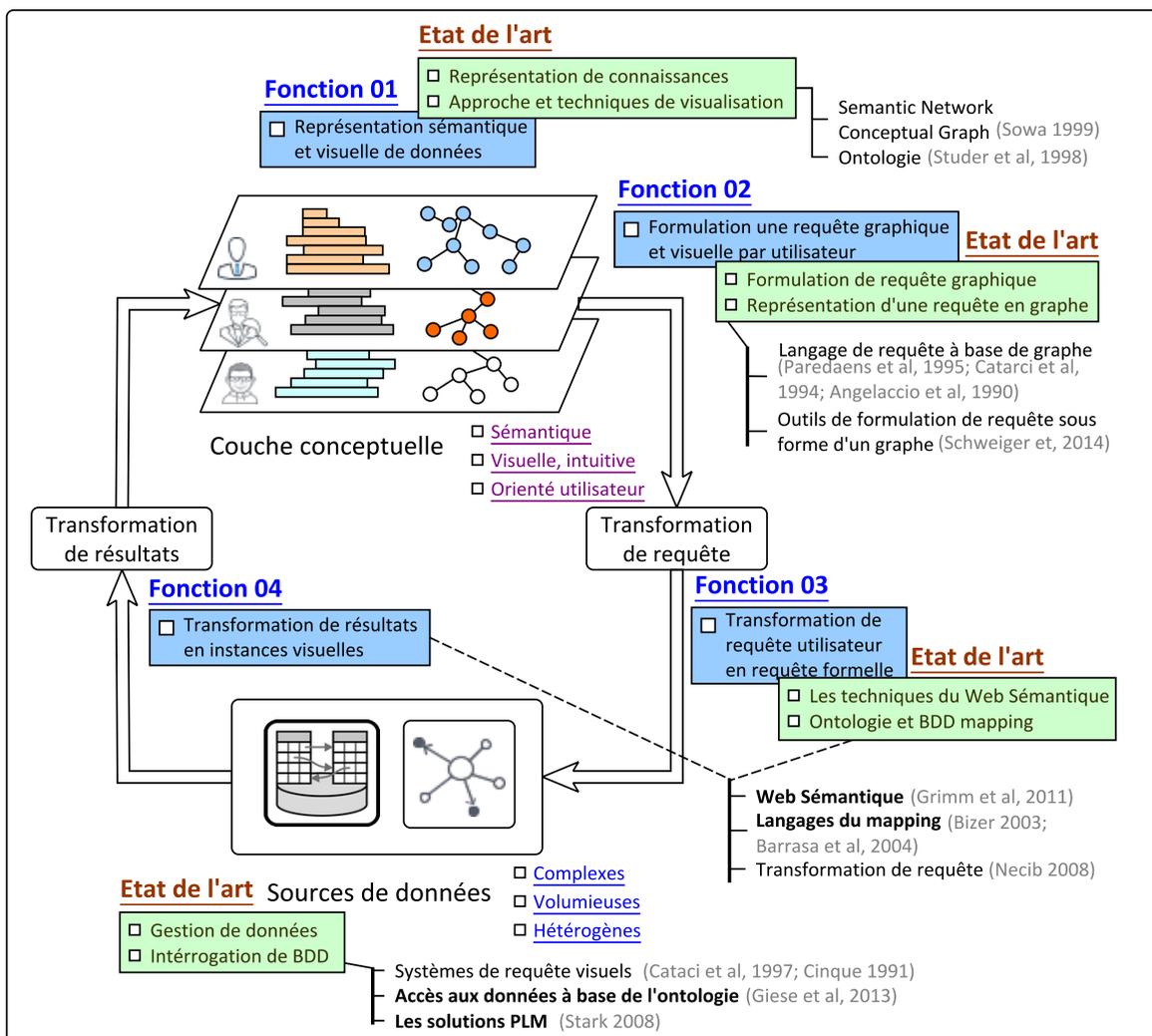


FIGURE 2.47 : Positionnement de la bibliographie par rapport au processus idéal.

requêtes. Nous proposons également une démarche générale permettant de construire des ontologies d'application et des mappings entre ces ontologies et les sources de données.

MÉTHODOLOGIE

3

SOMMAIRE

3.1	MÉTHODOLOGIE PROPOSÉE	92
3.1.1	Représentation visuelle de la sémantique du domaine	92
3.1.2	Accès Visuel et sémantique aux données	96
3.2	LA DÉMARCHE DE CONSTRUCTION DES ONTOLOGIES	106
3.2.1	Le cas d'étude	106
3.2.2	Acquisition de connaissances	107
3.2.3	Modélisation de connaissances	113
3.2.4	Edition de l'ontologie	116
3.2.5	Validation de l'ontologie	117
3.3	LA DÉMARCHE DE CONSTRUCTION DES MAPPINGS	118
3.3.1	Typologies du mapping	118
3.3.2	Langages du mapping	120
3.3.3	La construction des mappings	127
3.4	CONCLUSION	130

INTRODUCTION

Les travaux bibliographiques ont complété et justifié le contexte et la problématique que nous avons présentés dans la chapitre 1. Ils nous donnent aussi des arguments indispensables et une base solide pour notre proposition présentée dans ce chapitre. Avant d’aller dans le détail de la méthodologie proposée, nous rappelons les points importants du contexte et de la problématique.

Contexte

- **Une grande quantité de données complexes** : Les SGBDs doivent gérer une grande quantité de données qui sont complexes, indépendantes et hétérogènes. Il existe donc une énorme quantité d’interrelations entre les données.
- **La multi-utilisation de données** : L’ensemble des acteurs métiers travaillent sur un même référentiel commun.

Problématique

- Les données ne sont pas exploitées au maximum** : Dans le contexte de multi-utilisation de données, chaque utilisateur introduit son propre point de vue quand il ajoute des nouvelles données, elles sont donc difficilement compréhensibles par les autres utilisateurs.
- Les données sont difficilement accessibles** : L’accès aux données dans les SGBDs n’est pas facile, il est très limité et restreint aux acteurs métiers non-techniciens. Les données sont difficilement accessibles en raison du fait que, dans les SGBDs actuels, l’interrogation des données demande des *compétences techniques avancées* sur la *syntaxe du langage de requête* et une *compréhension assez solide* sur la *structuration des données stockées dans les BDDs*. Il y a donc un **gap** entre le **point de vue informatique** des techniciens qui construit les BDDs et le **point de vue métier** des utilisateurs finaux qui utilisent et interprètent des données. Ce gap limite la capacité d’accès aux données de ces utilisateurs.
- Les données restent inexploitées dans les BDDs, car elles sont difficiles à interpréter par l’ensemble des acteurs métiers. Il manque en fait des provenances nécessaires permettant aux utilisateurs qui ne sont pas acteurs des données de les comprendre. Autrement dit, **la sémantique des données et relations entre elles ne sont pas capturées pendant l’utilisation des acteurs métiers.**

Nous présentons dans ce chapitre une méthodologie permettant une **multi-utilisation des données entre l’ensemble des différents acteurs métiers**. Cette méthodologie profite des avantages des ontologies et de leur puissance de visualisation graphique pour fournir aux différents utilisateurs finaux un **accès visuel et sémantique aux données**.

Bien que les ontologies soient déjà utilisées dans la littérature pour rendre possible l’accès sémantique aux données des SGBD relationnelles, nous mettons en évidence le rôle et l’importance de la visualisation sémantique intégrée dans une couche conceptuelle.

Comme nous avons constaté ci-dessus, l'utilisation de données est améliorée si et seulement si les utilisateurs finaux peuvent interpréter et requêter eux-mêmes des données correspondant à leurs besoins en informations spécifiques. Pour l'interrogation des données, une solution possible est de faire monter en compétence technique des acteurs métiers par des formations et des entraînements dédiés. Cependant, cette approche est réalisable pour un petit groupe d'utilisateurs. Dans le contexte de multi-utilisation des données entre différents types d'acteurs métiers, elle prend souvent beaucoup de temps et dans certains cas, elle est infaisable et impraticable parce que chaque acteur a son propre niveau de compétence technique. Elle est également inappropriée pour les SI ou la BDD qui souvent évoluent dans leur structure.

De même, l'enrichissement de données au cours de l'activité de chaque utilisateur, amène à une multiplication d'étiquetage de ces données. Par conséquent l'accès et l'interprétation de ces données deviennent limités, ce qui empêche le partage des connaissances entre les acteurs.

Pour assurer la multi-utilisation et améliorer l'exploitation (*rechercher, interpréter*) de données de différents acteurs métiers, nous proposons d'utiliser une **représentation sémantique et intuitive** interface comme une couche conceptuelle entre le SGBD et les acteurs métiers. Par ce passage unique, les utilisateurs expriment leurs besoins d'information sur cette couche conceptuelle et non directement sur les sources de données. Nous allons présenter en détail cette démarche dans les suivantes sections.

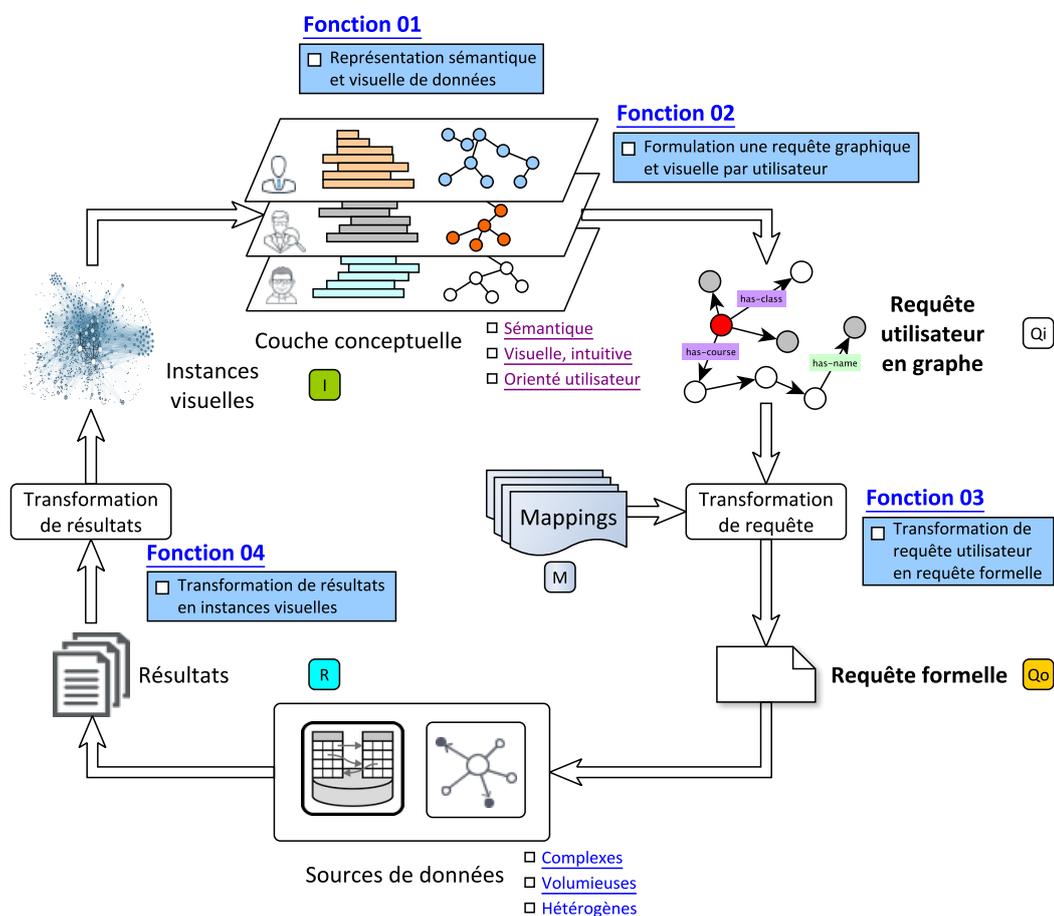


FIGURE 3.1 : La méthodologie proposée pour la multi-utilisation et l'accès sémantique aux données.

3.1 MÉTHODOLOGIE PROPOSÉE

Nous décomposons la méthodologie présentée brièvement ci-dessus en une démarche détaillée. Notre méthodologie comprend quatre fonctions principales expliquées ci-dessous et à la Figure 3.1 :

- **Représentation des connaissances du domaine** : Nous utilisons les ontologies d'application pour représenter la sémantique des données et, les connaissances du domaine qui couvrent le système d'information dédié. Cette représentation de connaissance joue le rôle d'une couche conceptuelle intermédiaire entre les utilisateurs et les sources de données. Nous profitons des avantages de la visualisation graphique pour que la visualisation soit le plus possible intuitive et compréhensive.
- **Formulation d'une requête par graphe** : Sur la couche conceptuelle, les utilisateurs expriment leurs besoins en information en utilisant des termes ontologiques. La requête utilisateur est représentée visuellement sous la forme d'un graphe (**Qi**) et cela pendant tout le processus de formulation.
- **Transformation de requêtes** : La requête utilisateur représentée sous la forme d'un graphe est transformée en une requête formelle (**Qo**) sur les sources de données. Pour réaliser cette transformation, il faut définir un ensemble de mappings (**M**) entre les ontologies et les sources de données. Ce type de travail a bien été étudié par des communautés (*Web Sémantique, bases de données, IA, etc.*) (cf section 2.6.4).
- **Transformation des résultats** : Les résultats (**R**) de la requête utilisateur doivent être transformés en instances visuelles (**I**) pour avoir une interprétation optimale par les utilisateurs. Idéalement, ces résultats sont affichés sur la couche conceptuelle en les reliant avec les ontologies prédéfinies.

Dans les sections suivantes, nous allons décrire en détail notre approche. Nous mettons en évidence tout d'abord l'importance de la représentation sémantique et visuelle dans la couche conceptuelle puis, nous décrivons comment, par l'utilisation de l'ontologie, il est possible d'élaborer cette couche. Nous détaillons ensuite la démarche de formulation d'une requête graphique et la transformation de ces dernières en requêtes formelles en s'inspirant du travail de Soyly (Soyly et al. 2016). Enfin, après avoir présenté la démarche de construction des mappings et des ontologies d'applications utilisées dans la couche conceptuelle, nous présentons les principes de la transformation de résultats en instances visuelles.

3.1.1 Représentation visuelle de la sémantique du domaine

3.1.1.1 Représentation sémantique du domaine par l'Ontologie

La sémantique ou les connaissances du domaine peuvent exister explicitement ou implicitement. Tandis que les connaissances explicites sont transmises entre les individus, les connaissances tacites (*implicites*) sont personnelles et complexes à formaliser, ce qui rend difficile de communiquer ou de partager avec les autres. Elles doivent être transformées et représentées explicitement grâce aux techniques de représentation des connaissances (De Smedt 1988, Doherty et al. 2006) (section 2.3) comme les ontologies. Des définitions et usages de

l'ontologie ont été présentée en détail dans la section 2.3. Brièvement, elle est “une spécification explicite et formelle d'une conceptualisation partagée” (Studer et al. 1998), l'ontologie utilise des concepts, relations, axiomes, instances, etc. explicitement et formellement définis pour décrire la sémantique attribuée aux données/ informations d'une manière explicite. L'ontologie représente une connaissance consensuelle, un point de vue général, compréhensif, partagé et accepté par les individus. Pour cette raison, nous utilisons l'ontologie dans la couche conceptuelle pour capturer et représenter aux acteurs métiers la sémantique des données, informations et connaissances du domaine.

3.1.1.2 Visualisation des ontologies

Naturellement, le cerveau humain peut assimiler un grand nombre de données et d'informations qui sont représentées graphiquement (Tricot 2006). La visualisation joue donc un rôle indispensable dans les systèmes d'information actuels pour une meilleure interprétation/ exploitation de données et d'informations. La visualisation permet aussi aux utilisateurs d'interagir avec une énorme quantité de données facilement et efficacement afin de trouver les informations utiles le plus rapidement possible. En même temps, la visualisation peut mettre en évidence des relations abstraites qui sont cachés derrière des éléments visualisés (Cox 1999). Par conséquent, elle facilite le processus de perception, d'interprétation et de raisonnement. Pour les raisons ci-dessus, les connaissances du domaine dans la couche conceptuelle doivent être représentées graphiquement et visuellement pour une compréhension et une interprétation optimale.

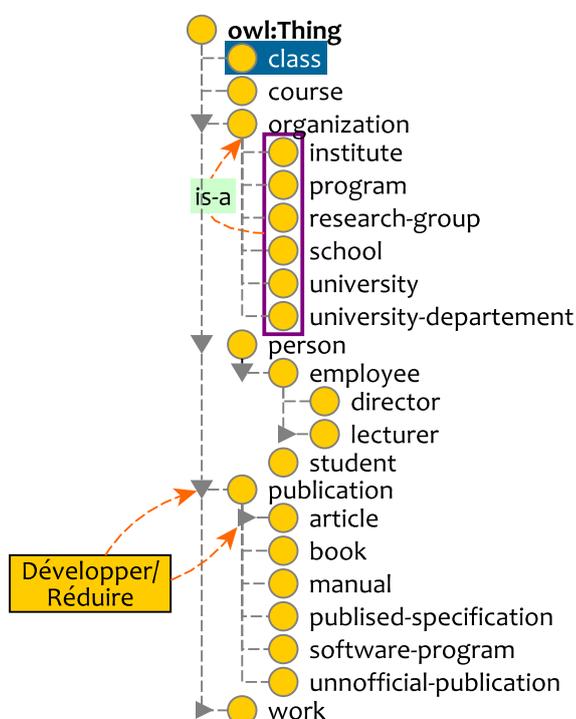


FIGURE 3.2 : Arbre indenté des concepts ontologiques

Le monde réel est reconnu et modélisé dans le cerveau humain par des concepts et des associations entre eux. Des dépendances et associations complexes peuvent être plus facilement

comprises lorsque nous les illustrons en utilisant une représentation visuelle au lieu des expressions textuelles ou verbales (Marchese et Banissi 2013). Du coup, la représentation des connaissances sous forme de réseaux sémantiques et notamment les graphes conceptuels (Sowa 1999) fait appel à cette capacité d'interprétation, puisque les connaissances sont schématisées sous forme d'arbres et de graphes de *concepts/ relations*.

Etant utilisées pour exprimer explicitement les connaissances du domaine, les ontologies devraient être représentées visuellement pour assurer une interprétation efficace, une exploitation optimale et un partage aisé entre différents acteurs du domaine.

Parmi les techniques de visualisation d'ontologies présentées dans la section 2.4.4, nous proposons d'utiliser deux techniques : "**arbre indenté**" et "**graphe**".

- Selon notre observation, un "**arbre indenté**" est une technique simple mais efficace pour *présenter/ illustrer* des structures hiérarchiques des concepts ontologiques, les relations parent-enfants (les relations "*is-a*"). En partant du noeud racine, avec l'aide des outils d'interaction (*développer/ réduire* un concept), les utilisateurs peuvent naviguer au travers de l'arbre de concepts pour avoir une vue générale (*abstraite*) ou détaillée de la structuration des concepts.

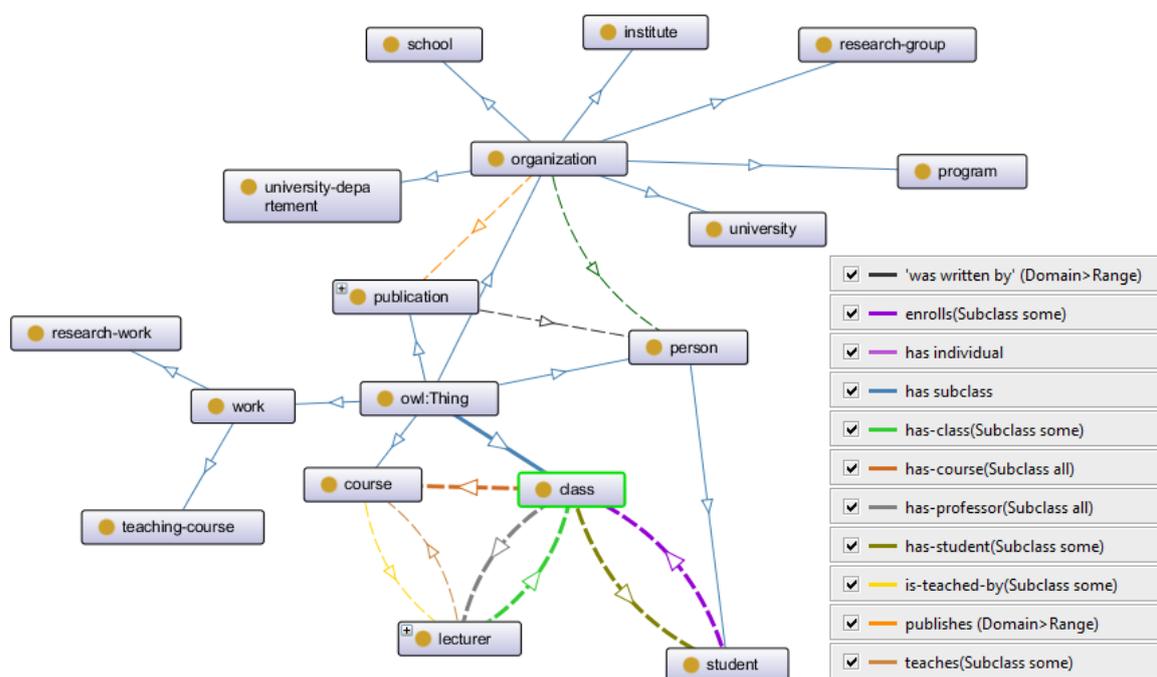


FIGURE 3.3 : Visualisation graphique de toutes les relations sémantiques entre concepts ontologiques.

- Dans la théorie des graphes, un **graphe** est une structure composée d'un ensemble de noeuds (sommets) et d'un ensemble d'arêtes qui établissent des relations (connexions) entre les noeuds⁵¹. La représentation basée sur le graphe est souvent utilisée pour les ontologies ou chaque noeud illustre un concept et chaque arête illustre une relation sémantique entre les concepts. Cette représentation fournit une visualisation intuitive de toutes les relations sémantiques qui existent dans l'ontologie.

51. https://en.wikipedia.org/wiki/Graph_theory

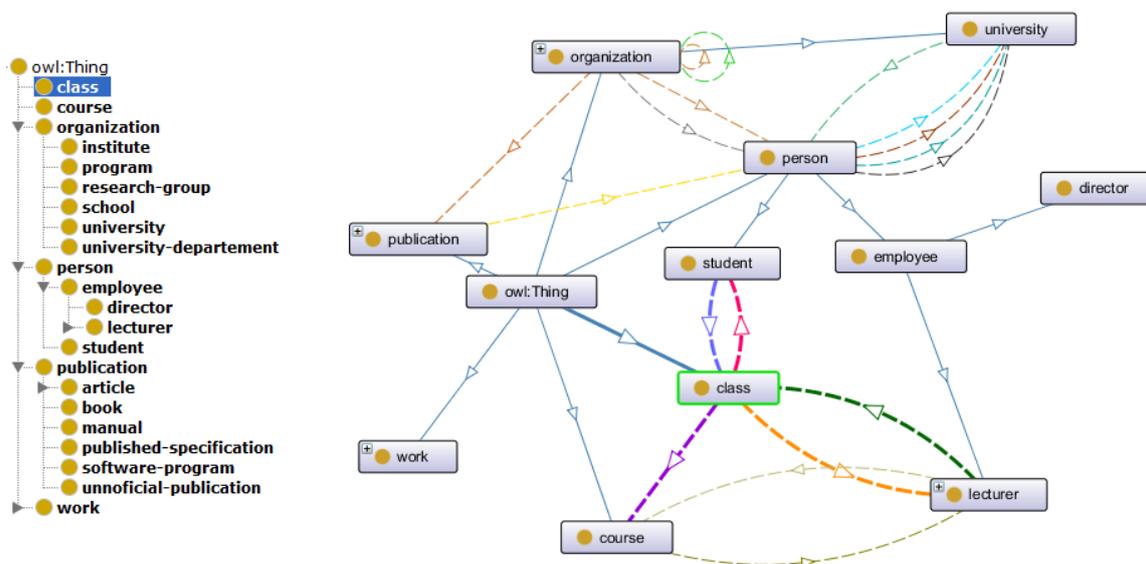


FIGURE 3.4 : La représentation visuelle de l'arbre des concepts ontologiques et les relations sémantiques entre eux dans une seule vue.

- Nous pensons qu'il est possible que les utilisateurs puissent manipuler avec l'arbre indenté de la hiérarchie des concepts et le graphe illustré des relations sémantiques entre concepts, dans une seule interface. Par exemple, en naviguant dans l'arbre des concepts, les utilisateurs peuvent *consulter/ voir* directement quels sont les concepts et les relations sémantiques concernant le concept sélectionné. Cela permet une exploration de ces concepts reliant une spécialisation vers les sous-concepts et une description de son écosystème à travers les relations avec les autres concepts du domaine.

La Figure 3.4 illustre une seule vue combinant la liste indentée de concepts et le graphe de relations entre ces concepts ontologiques. Dans la Figure 3.5, nous ajoutons un texte en haut pour justifier l'utilisation de l'ontologie dans la couche conceptuelle.

3.1.1.3 Construction de la représentation sémantique

La représentation visuelle de la sémantique du domaine est divisée en deux parties : La construction des ontologies qui capturent et expriment la sémantique du domaine ; et la visualisation des ontologies. Nous présentons une démarche pratique plus détaillée de cette construction dans la section suivante. Nous synthétisons ici les caractéristiques importantes de cette représentation :

- Visuelle, intuitive et interactive** : La représentation devrait être compréhensive et facilement utilisée.
- Respecter la sémantique du domaine** : Une représentation graphique est d'autant plus efficace qu'elle correspond à la représentation mentale que se fait l'utilisateur des informations (Chabris et Kosslyn 2005). Pour être efficace (compréhensive par les acteurs métiers), la visualisation doit respecter la sémantique du domaine, autrement dit, elle visualise l'ensemble des concepts permettant d'appréhender les objets du domaine.
- Besoin des retours d'expérience** : La construction de la visualisation est un procédé récursif. Dans les premières versions, une visualisation ne peut pas être considérée

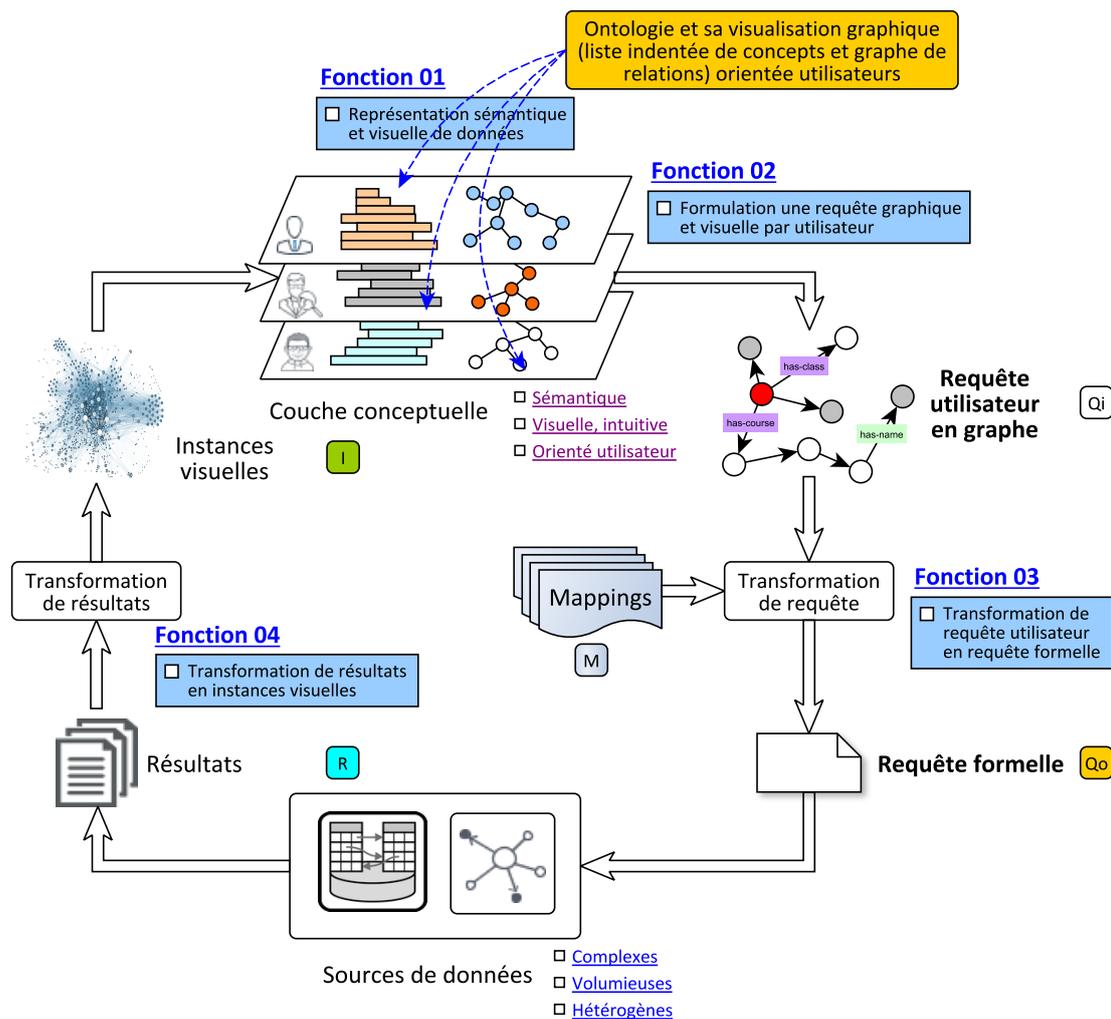


FIGURE 3.5 : Positionnement de la représentation sémantique, visuelle et orientée utilisateur à base d'ontologie, dans l'approche générale.

comme définitive et universelle. Pour que la visualisation soit esthétique, intuitive et efficace, il est important de se laisser la possibilité de faire évoluer la visualisation selon les besoins des utilisateurs. Par conséquent, les retours d'expérience constituent un critère fiable pour juger de l'efficacité des visualisations.

- iv) **Orientée utilisateurs :** Chaque individu possède ses propres compétences (techniques ou métiers), la visualisation devrait être orientée utilisateur. Plusieurs mises en page de visualisation peuvent être proposées pour les différents types d'acteurs métiers.

3.1.2 Accès Visuel et sémantique aux données

Cette approche se réalise en trois étapes : La formulation de requête graphique et sémantique par utilisateur, la transformation de requête et la transformation de résultats.

3.1.2.1 Formulation de requête graphique et sémantique par utilisateur

Contrairement aux moteurs de recherche sur le Web, l'accès aux données dans les systèmes d'information traditionnels n'accepte pas les résultats non pertinents ou non suffisants. Du

coup, afin d'atteindre des réponses précises et complètes, les besoins d'information des acteurs métiers doivent être exactement spécifiés. Les langages de requête structurés tels que SQL ou XQuery sont très expressifs, mais ils demandent des compétences techniques et des connaissances approfondies sur la syntaxe du langage et le schéma du domaine. De plus, les données complexes et hétérogènes peuvent être stockées et enregistrées sous différents modèles de données. Les utilisateurs ne peuvent donc pas formuler eux-mêmes des requêtes souhaitées, sans avoir des connaissances sur la structure des données et des notions sur le langage de requête de la base de données. Ajouter des abstractions au-delà des données, les ontologies fournissent une vue sémantique, commune et unifiée où les données sont organisées et structurées informatiquement. Sur cette couche conceptuelle, nous proposons une démarche de formulation de requête qui est graphique, sémantique et par utilisateur.

- Pour l'aspect "**sémantique**", nous pensons que les acteurs métiers peuvent formuler eux-mêmes des requêtes en utilisant la représentation sémantique de donnée et connaissance du domaine sur la couche conceptuelle, même dans le cas où ils n'ont pas les compétences techniques nécessaires. La requête utilisateur sera donc exprimée sous la forme d'un graphe.
- Pour l'aspect "**par utilisateur**", nous impliquons que la représentation sémantique peut être orientée utilisateur. Chaque groupe d'utilisateurs (ou *un seul individu*) pourrait avoir sa propre représentation sémantique, au travers de laquelle, ils formulent des requêtes.
- Pour l'aspect "**graphique**", nous supposons que les requêtes utilisateurs devraient être formulées de façon visuelle et interactive. C'est-à-dire, l'utilisateur n'exprime pas son besoin en information textuellement ou syntaxiquement, mais graphiquement. La formulation d'une requête est en fait le processus de construction d'un graphe de requête qui utilise un ensemble de règles prédéfinies pour représenter des éléments différents de la requête (objets, attributs, opérations, valeurs d'entrées, conjonctions, etc.). Ce graphe de requête contient principalement des noeuds qui sont les concepts ontologiques et des arcs qui sont les relations entre deux concepts.

L'ensemble des mécanismes d'interaction et de manipulation directe tels que la navigation devra être intégré dans le processus de la formulation de requête. Les utilisateurs peuvent naviguer à travers de l'arbre des concepts ontologiques ou utiliser les fonctions de recherche pour trouver les concepts souhaités. En outre, la visualisation graphique et intuitive de ces concepts et de ces relations sémantiques entre elles permettra aux utilisateurs de mieux comprendre les concepts pendant le processus de formulation des requêtes et même d'apprendre de nouveaux concepts émanant d'autres utilisateurs.

Des nouvelles connaissances (*faits, règles, etc.*) peuvent être extraites à partir de celles déjà déclarées dans les ontologies d'application grâce à des moteurs d'inférences. Ces moteurs peuvent être utilisés pendant la formulation de requête afin d'aider les acteurs métiers à compléter la requête selon leurs besoins d'information. Ils peuvent être utilisés également pour enrichir automatiquement la requête utilisateur initiale à condition que la requête enrichie retourne les résultats plus complets et appropriés. Les requêtes utilisateurs formulées sur

les termes ontologiques seront transcrites en requêtes sur les sources de données à travers un mapping (voir section suivante).

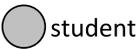
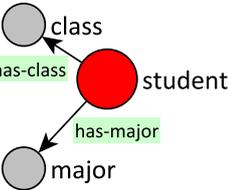
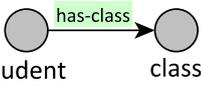
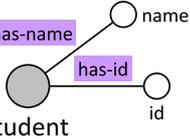
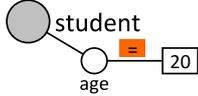
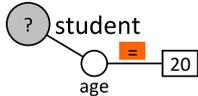
Éléments graphiques	Explication
	Le rond gris (un noeud) à taille moyen représente un concept ontologique. Le nom du concept est affiché à coté du rond.
	Le noeud sélectionné en cours est affiché par un rond rouge à taille la plus grande. Les autres noeuds sont restés à leurs couleurs originales.
	Un arc représente une bi-relation entre deux concepts (noeuds). La direction de l'arc illustre le sens de la relation. Le nom de la relation (ObjectProperty) est affiché dans un rectangle vert non-bordé.
	Les attributs d'un concept sont représentés par les ronds blancs à taille petite et affichés autour du concept. Les DatatypeProperty propriétés sont affichées dans un rectangle violet non-bordé.
	La valeur d'un attribut est représentée dans un rectangle orange non-bordé.
	Le point d'interrogation affiché à l'intérieur du rond indique que les valeurs du concept correspondant seront récupérées dans la requête SELECT.

TABLE 3.1 : Les conventions graphiques utilisées pour la représentation graphique des requêtes SPARQLs.

3.1.2.1.1 Formulation de requêtes en graphe

Les requêtes utilisateurs sont formulées au fur et à mesure avec l'aide des outils de visualisation et d'éléments graphiques.

En partant du noeud racine, un utilisateur peut naviguer à travers l'arbre des concepts ontologiques pour trouver les concepts correspondant aux données souhaitées et les ajouter à la requête. L'utilisateur peut également chercher les concepts ontologiques dans le graphe des relations. Chaque fois que l'utilisateur sélectionne un concept (dans l'arbre ou dans le graphe de relations), l'ensemble des concepts et des relations reliées à ce concept sont mis en évidence. Cette fonction aide l'utilisateur à reconnaître facilement comment les concepts sont interconnectés afin d'enrichir et de compléter la requête. La requête utilisateur est représentée graphiquement en temps réel dans une zone dédiée durant le processus de formulation. L'utilisateur peut à tout moment modifier les paramètres déjà définis : *supprimer/ ajouter* un concept, changer une opération ou la valeur d'un attribut, etc. Un mécanisme de suggestion et des interactions entre la requête et la visualisation des ontologies sont utilisés pour réduire le temps de formulation des requêtes.

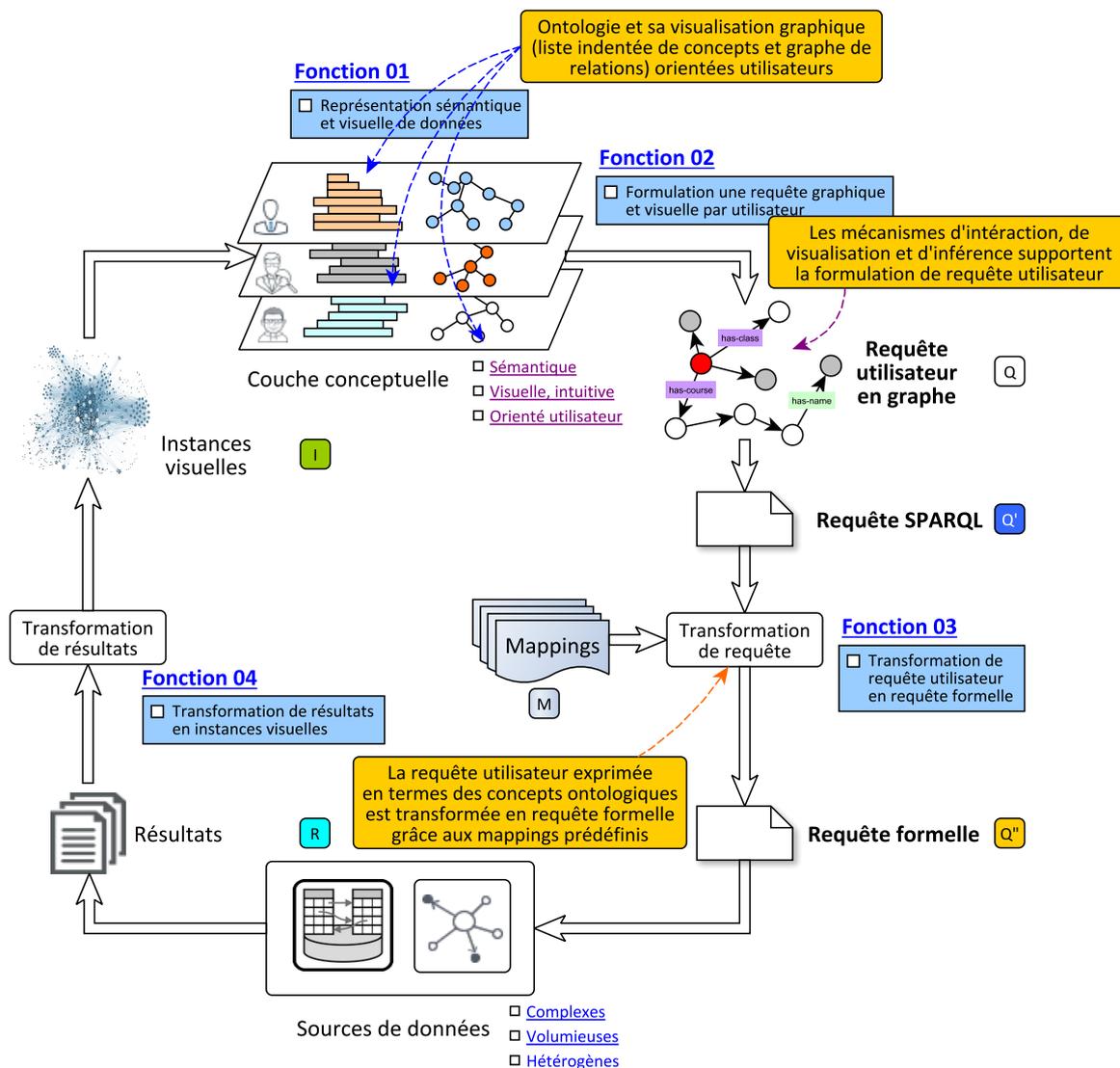


FIGURE 3.6 : La requête utilisateur est formulée graphiquement et exprimée en termes des concepts ontologiques grâce aux mécanismes d'inférence et d'interaction.

3.1.2.1.2 Mécanismes d'inférences

Des mécanismes d'inférence peuvent être utilisés pour compléter ou enrichir la requête utilisateur initiale. Il y a deux approches possibles :

1. La **suggestion** : Les mécanismes d'inférence sont utilisés pendant le processus de formulation de requête graphique. Dans ce cas-là, ils ont pour objectif de suggérer des concepts et des relations pertinentes permettant d'enrichir et de compléter la requête utilisateur. La requête devra être formulée et façonnée au fur et à mesure notamment quand les utilisateurs ne connaissent pas trop la BDD ou quand ils n'ont pas une idée claire de leur besoin en information.
2. L'**enrichissement de la requête initiale** : Dans cette approche, les mécanismes d'inférence sont utilisés quand la requête utilisateur en graphe est déjà réalisé. Une fois que la requête graphique utilisateur est terminée, elle est transformée en requête SPARQL. Basé sur les logiques déclarées dans l'ontologie, les mécanismes d'inférence vont enri-

chir et compléter la requête SPARQL transformée. Par conséquent, les résultats obtenus sont plus complets et plus riches que ce que les utilisateurs “entendent”.

Nous avons choisi la première approche où les mécanismes d’inférence sont implémentés ensemble avec les mécanismes d’interaction et de visualisation, et sont utilisés pendant la formulation de requête graphique. Cependant, à la fin, la requête en graphe aussi est transformée en SPARQL avant d’être transformée en requêtes formelles sur les sources de données.

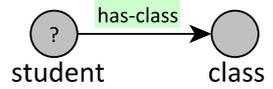
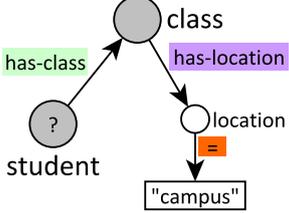
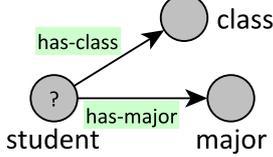
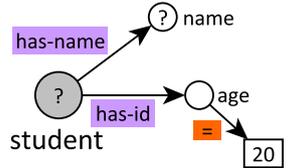
Eléments graphiques	Description	Requête SPARQL correspondante
	Selectionner une classe des instances.	<pre>SELECT ?std WHERE ?std rdf:type schl:student.</pre>
	Les propriétés sont représentées par des prédicats dans les triples.	<pre>SELECT ?std WHERE ?std rdf:type schl:student.</pre>
	Selectionner des instances d'une classe restreinte par des conditions sur une autre classe. Le chemin à parcourir entre des noeuds correspond à un ensemble des triples successifs.	<pre>SELECT ?std WHERE ?std rdf:type schl:student. ?cl rdf:type schl:class. ?lc rdf:datatype schl:location. ?std schl:has-class ?cl. ?cl schl:has-location ?lc. FILTER (?lc = "campus"^^xsd:string).</pre>
	Selectionner des instances d'une classe restreinte par des relations avec des autres classes.	<pre>SELECT ?std WHERE ?std rdf:type schl:student. ?cl rdf:type schl:class. ?mj rdf:type schl:major. ?std schl:has-class ?cl; schl:has-major ?mj.</pre>
	Selectionner des instances d'une classe restreinte par des conditions sur ses attributs.	<pre>SELECT ?std, ?name WHERE ?std rdf:type schl:student. ?name rdf:datatype schl:name ?age rdf:datatype schl:age ?std schl:has-name ?name. ?std schl:has-age ?age.FILTER (?age = 20).</pre>

TABLE 3.2 : Tableau de transformation entre les représentations graphiques d’une requête utilisateur et les requêtes SPARQL correspondantes.

3.1.2.2 Les conventions de visualisation

Pour représenter une requête utilisateur sous forme d’un noeud-arc graphe, nous utilisons un ensemble d’éléments graphiques identifiés à base des spécifications du langage SPARQL ⁵² et en s’inspirant des travaux de (Haag et al. 2015, Russell et al. 2008). En général, un concept (un *noeud*) est représenté par un rond et une relation est représentée par l’arc entre deux noeuds. Ces éléments sont se sont distingués par la *forme*, la *taille*, la *couleur* et la *phrase textuelle*.

Forme et taille :

- Les concepts sont représentés par les ronds de taille moyenne.

52. <https://www.w3.org/TR/rdf-sparql-query/>

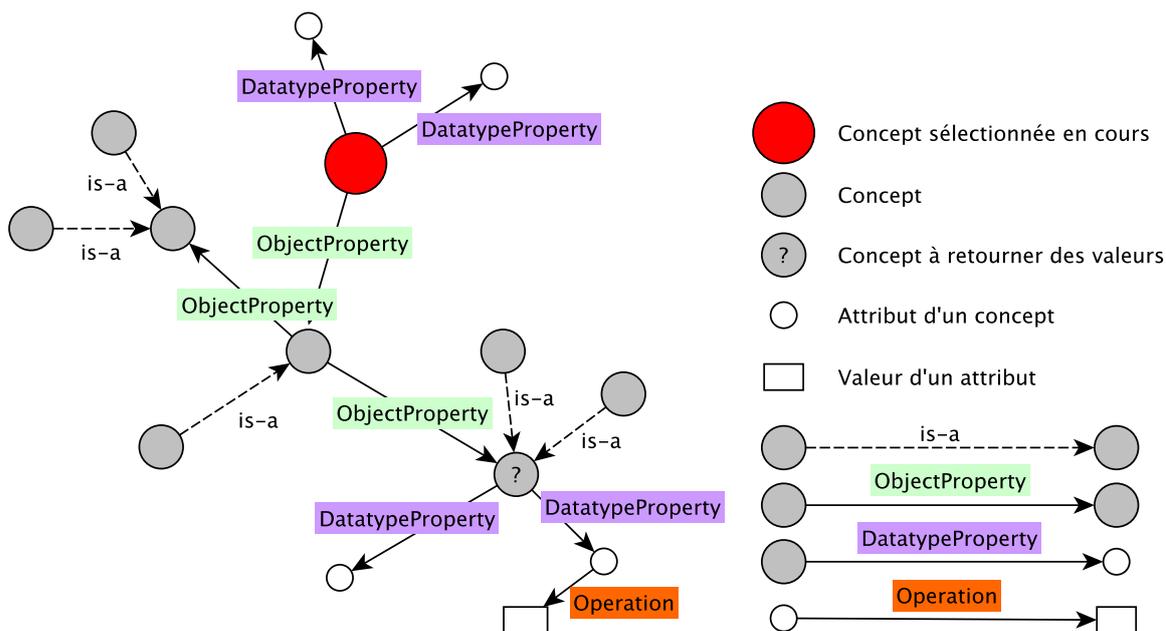


FIGURE 3.7 : La représentation graphique de la requête utilisateur grâce à l'ensemble des éléments graphiques (voir Tableau 3.1).

- Les attributs d'un concept ("Datatype") sont représentés par les ronds de taille petite autour du rond représenté le concept.
- La valeur d'un attribut est affichée dans un rectangle bordé.
- Les relations entre deux concepts ("ObjectProperty"), entre un concept et un attribut ("DatatypeProperty"), ou les opérations entre un attribut et sa valeur sont représentées dans des rectangles non bordés.

Couleur :

- Les noeuds représentés des concepts sont affichés en gris. Les noeuds représentés des attributs d'un concept sont affichés en blanc. Le noeud sélectionné en cours est affiché en rouge et à la taille la plus grande.
- Les rectangles représentant les "ObjectProperty" sont affichés en vert, les rectangles représentant les "DatatypeProperty" sont affichés en violet et les rectangles représentant les opérations sont affichés en orange.

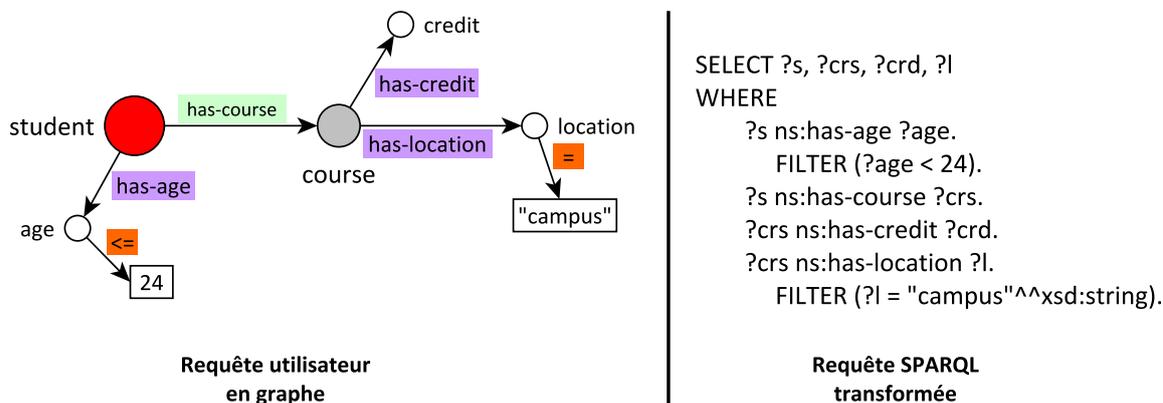


FIGURE 3.8 : La requête utilisateur en graphe est transformée en requête SPARQL.

Une fois la requête graphique utilisateur complète, elle est transformée en requête SPARQL. Une requête SPARQL est une collection dite de "triples". Chaque triple est représenté par un noeud "sujet" connecté au noeud "objet" à travers de la relation "prédicat". La Figure 3.7 illustre un exemple de la requête SPARQL et sa représentation graphique selon les conventions de visualisation ci-dessus et les règles de transformations présentées dans le Tableau 3.2.

3.1.2.3 Transformation de requête

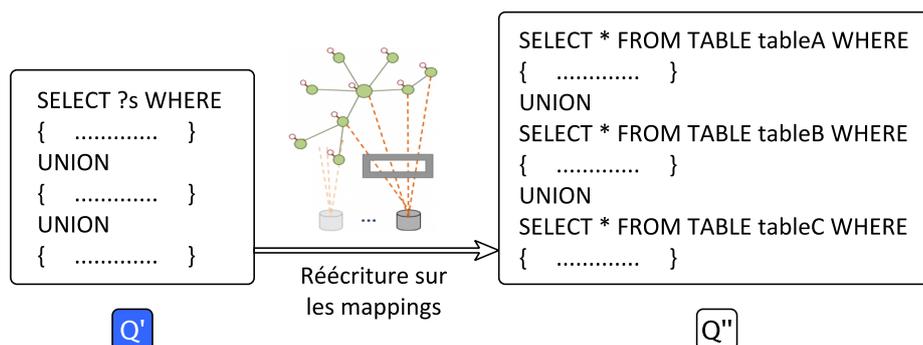


FIGURE 3.9 : La réécriture sur les mappings. Entrée : La requête SPARQL, Sortie : La requête en langage de requête sur les sources de données (SQL par exemple).

Dans cette approche, les sources de données sont fixées et connues à l'avance, le schéma global utilisé dans la couche conceptuelle est basé sur l'approche "Global as View". Cette approche exige que chaque élément du schéma global soit défini comme vues sur les sources de données.

Dans cette phase de réécriture, pour chaque segment p de la requête (Q') sur les ontologies, on cherche la requête q correspondante sur les sources de données où (p, q) est un triple prédéfini dans les mappings.

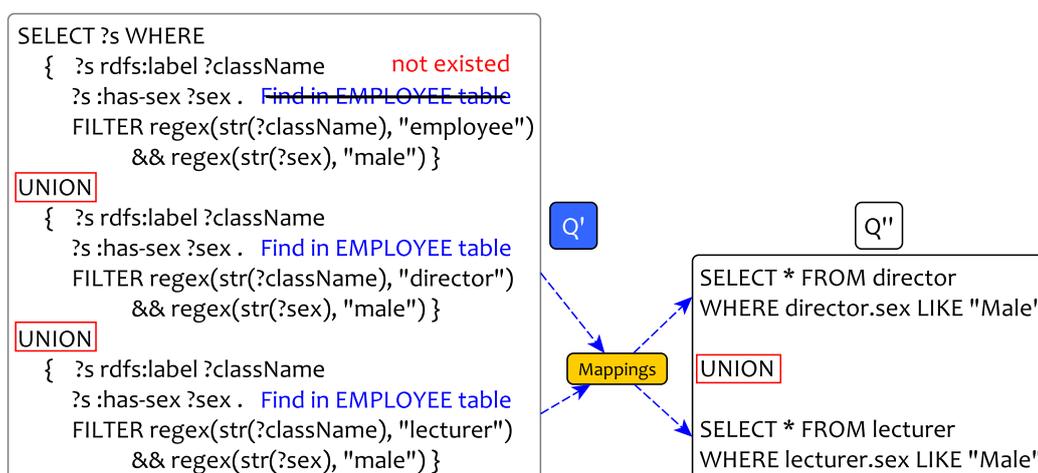


FIGURE 3.10 : La transformation de requête : Un exemple illustré sur les mappings.

Les requêtes transformées sur les sources de données sont ensuite exécutées par le moteur d'évaluation des requêtes du système d'information. La Figure 3.11 illustre les résultats obtenus de la requête Q'' sous la forme d'une table.

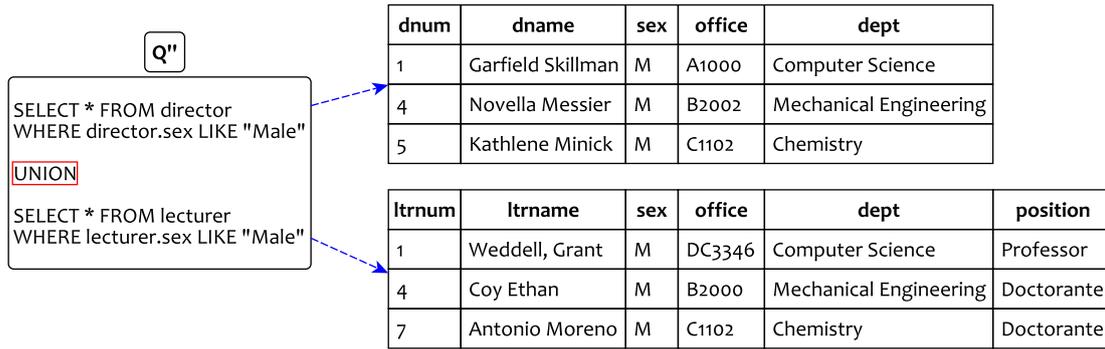


FIGURE 3.11 : Les résultats des requêtes.

3.1.2.4 Transformation de résultats

Les corrélations entre résultats peuvent être identifiées à partir des relations entre concepts ontologiques. Les résultats de la requête utilisateurs peuvent être transformés en instances visuelles et sont ensuite affichés graphiquement sur l'interface de requête (Figure 3.12).

Nous proposons d'utiliser des techniques de visualisation de données et d'informations qui sont déjà développées pour pouvoir fournir une meilleure représentation graphique et dynamique des résultats. Des outils de navigation et d'interaction devraient être intégrés pour aider les acteurs métiers à interpréter et explorer des données.

3.1.2.5 Mapping entre l'ontologie et les sources de données

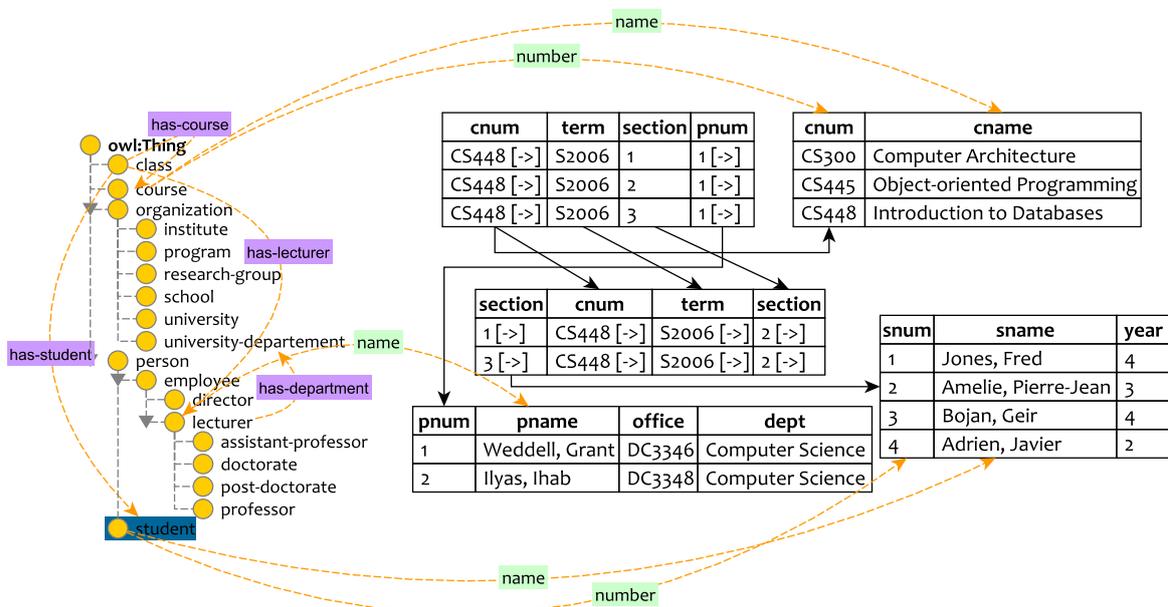


FIGURE 3.13 : Les mappings relient sémantiquement les concepts ontologiques aux données dans la base de données relationnelle.

La requête utilisateur en graphe est transformée en requêtes formelles sur les sources de données en utilisant un ensemble de mappings prédéfinis. Ces mappings sont utilisés aussi dans la transformation de résultat. En effet, les sources de données sont généralement autonomes

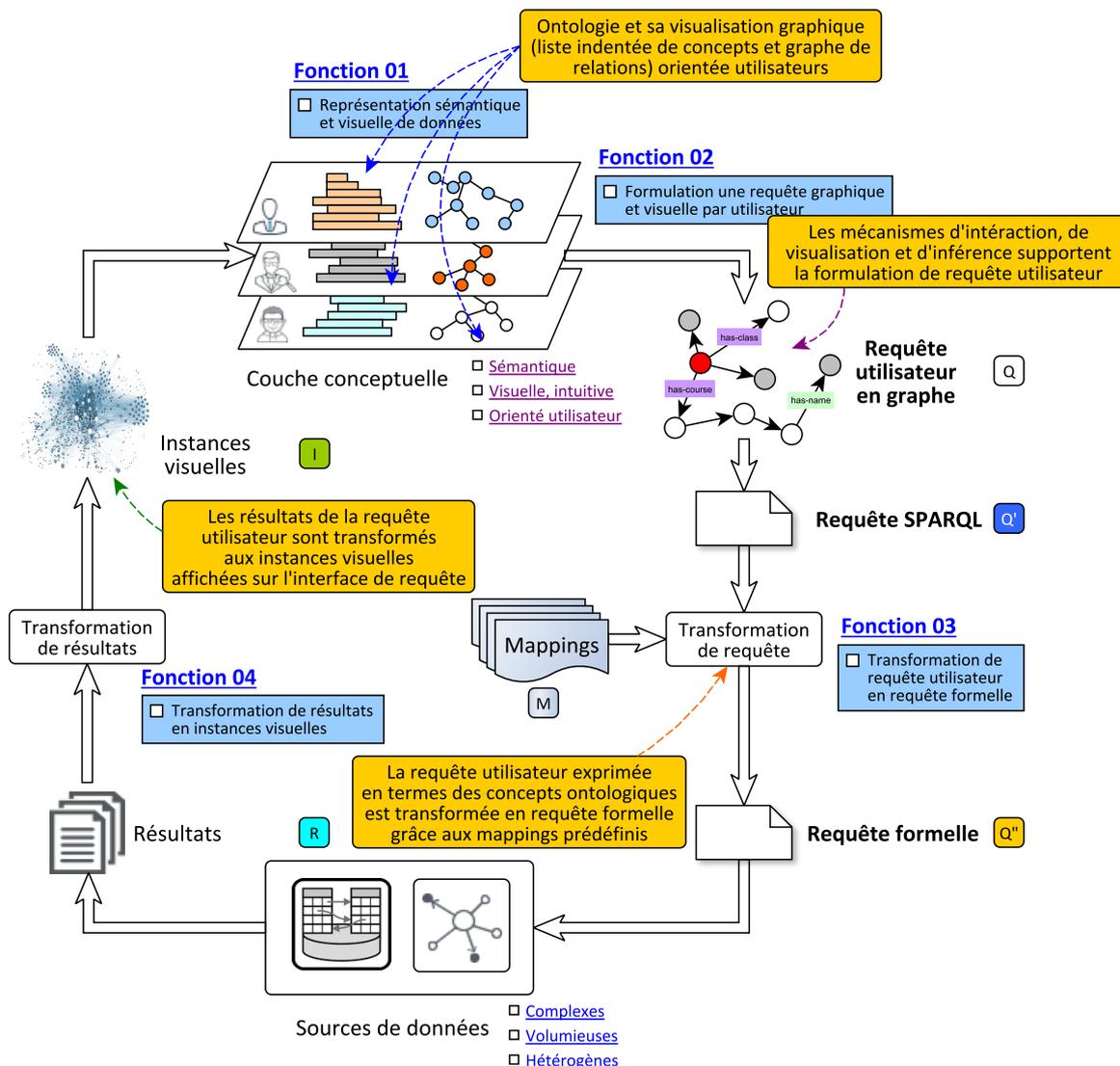


FIGURE 3.12 : La requête utilisateur exprimée en termes des concepts ontologiques (Q') est transformée en requêtes exprimées en langage de requête formel grâce à l'ensemble des mappings prédéfinis. Les résultats obtenus sont ensuite transformés aux instances visuelles affichées sur l'interface de requête.

et donc la manière dont ils sont structurés en général ne tient pas compte de l'ontologie, ce qui est plutôt une représentation indépendante du domaine d'intérêt. Les mappings sont des ponts entre les ontologies et les sources de données. Ils relient la couche conceptuelle (la représentation sémantique exprimée par les ontologies) à la couche de données du système ou les relations sémantiques; en d'autres termes, c'est un pont entre les concepts ontologiques et les données dans la base de données.

En général, les mappings sont constitués par un ensemble "d'assertions", chacune spécifie une relation entre la source et la cible. Dans sa forme la plus générale, une assertion précise une correspondance entre une vue sur la source (les ontologies) et une vue sur la cible (les sources de données). Selon Mariano (Mariano 2012), les mappings sont un ensemble de 2-tuples de deux requêtes, une source les données et une sur les ontologies, avec la même signature :

$$M = (p_i, q_i); i = 1..n$$

où : p_i est la requête sur les ontologies, q_i est la requête sur les sources de données.

Un tuple associe les données spécifiées par la requête Q avec les réponses pour la requête Q' . Plusieurs langages permettant d'exprimer le mapping entre la base de données relationnelle et l'ensemble de données comme proposés dans Direct Mapping⁵³, eD2R (Barrasa et al. 2003), Virtuoso RDF Views (Erling et Mikhailov 2009), etc. Figure 3.14 illustre un exemple du mapping entre deux requêtes, une sur la table "Director" de la base de données relationnelle et une sur le concept ontologique "Director" en turtle syntaxe (ontop⁵⁴ projet et R2RML syntaxe⁵⁵).

Les travaux de recherche ainsi que les expériences acquises dans le projets industriel ANR BIOMIST ont montré que la spécification des mapping est une activité très complexe qui exige une compréhension profonde à la fois de l'ontologie et des sources de données. En effet, les sources de données sont généralement autonomes et préexistantes dans l'application OBDA et donc la manière dont ils sont structurés en général ne tient pas compte de l'ontologie, ce qui est plutôt une représentation indépendante du domaine d'intérêt.

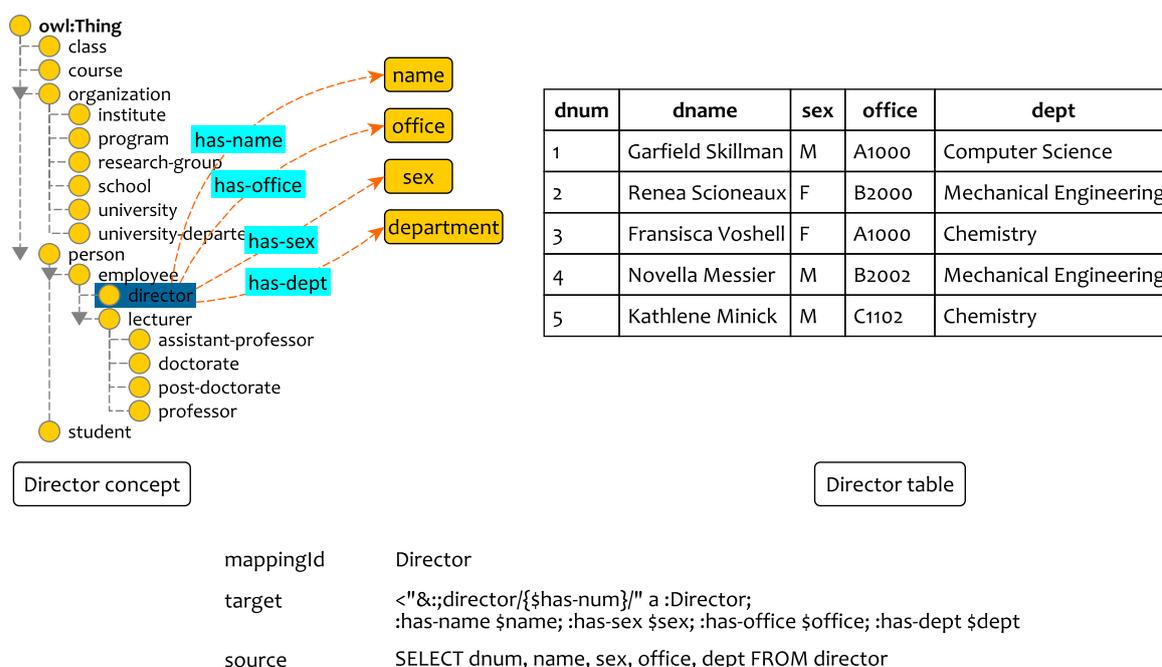


FIGURE 3.14 : Exemple d'un mapping entre deux requêtes, une sur les ontologies (SPARQL) et une sur les sources de données (SQL).

L'ensemble des mapping entre les concepts ontologiques et ses représentations dans les sources de données est souvent produit par les IT-experts qui ont des expériences et compétences nécessaires en traitant des besoins d'information des utilisateurs dans les systèmes d'information traditionnels. Ces personnes s'occupent également des missions de maintenance des systèmes OBDA comme l'évolution des données, les changements de la classification et la mise à jour des ontologies et des mapping. Le mapping avec les classes de la

53. <https://www.w3.org/DesignIssues/RDB-RDF.html>

54. <https://github.com/ontop/ontop>

55. <https://www.w3.org/TR/r2rml/>

ontologie est détaillée et grande, plus sa conception est difficile. En général, la conception d'une ontologie d'application passe par quatre étapes principales⁵⁶ :

- a) **Acquisition de connaissances,**
- b) **Modélisation de connaissances,**
- c) **Représentation de connaissances,**
- d) **Validation d'ontologie.**



FIGURE 3.16 : Le processus de construction des ontologies d'applications en quatre étapes.

3.2.2 Acquisition de connaissances

L'acquisition de connaissances est la première étape du processus de conception. L'objectif de cette phase est d'identifier des concepts et relations sémantiques pertinents à mettre dans l'ontologie cible. Nous rappelons qu'un concept est une expression linguistique d'une conceptualisation, et une relation binaire entre deux concepts est une représentation de la façon dont ils sont interdépendants. Typiquement, une relation spécifie dans quel sens un concept est lié à un autre concept dans l'ontologie (Khou et Na 2006). Pour les ontologies d'application qui sont construites pour des applications spécifiques et souvent prédéfinies, cette étape peut être facilitée en réutilisant des modèles conceptuels des BDDs du système d'information cible ou l'ontologie du domaine déjà existante car une ontologie d'application est souvent une spécification de cette dernière.

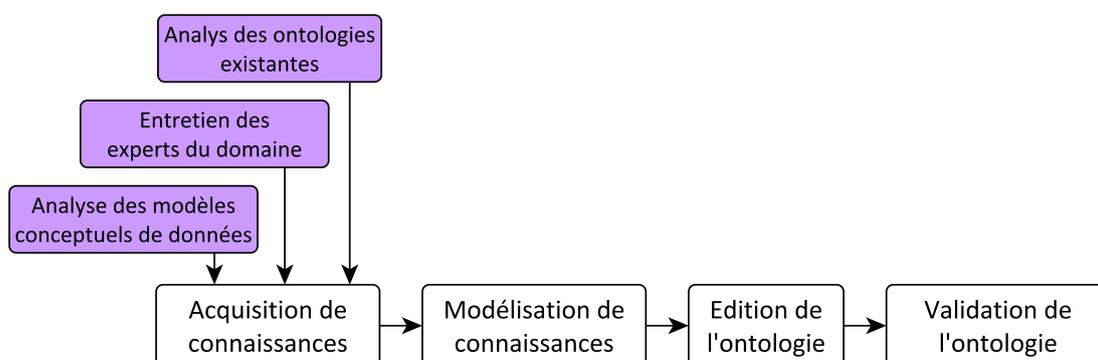


FIGURE 3.17 : Acquisition de connaissances pour la construction des ontologies d'application.

3.2.2.1 Entretien des experts du domaine

La construction des ontologies est une tâche difficile et fastidieuse. Une des caractéristiques les plus importantes des ontologies est qu'elles fournissent une description sémantique et consensuelle d'un domaine. Autrement dit, la définition des concepts ontologiques et des relations sémantiques entre eux doit être compréhensive, acceptée et partagée largement par la communauté. Les ontologies doivent être soigneusement définies par un groupe d'experts qui ont des compétences profondes du domaine. Les connaissances représentées dans une

56. http://liris.cnrs.fr/amille/enseignements/DEA-ECD/ontologies/construction_ontologie.htm

ontologie souvent dépassent le savoir-faire d'un seul individu ou d'un petit groupe d'acteurs métiers. Du coup, la collaboration entre différents experts assure aussi la richesse de connaissances décrites dans l'ontologie, notamment les ontologies du domaine à grande échelle.

Pour que l'ontologie construite représente des connaissances consensuelles et partagées, une première phase de développement, lorsque le consensus n'est pas encore atteint, est nécessaire (Keita 2007). Dans cette phase, les différents experts du domaine utilisent des outils supportés pour exprimer et modéliser leurs propres points de vue. Des discussions et échanges doivent être ensuite réalisés entre ces experts pour pouvoir aller vers le consensus.

3.2.2.1.1 Démarche de l'entretien

Dans le cadre de la construction des ontologies d'application, les entretiens ont pour objectif de comprendre les processus métiers, les besoins et les manières d'utilisation des données de chaque acteur métier. La démarche de l'entretien doit prendre en compte les points suivants :

Préparation pour l'entretien : L'entretien demande la participation des acteurs métiers de différentes disciplines/ groupes de recherche, il faut planifier tout d'abord un rendez-vous pour l'entretien qui favorise tous les participants.

Durant l'entretien :

- Nous recommandons d'utiliser une méthodologie de capitalisation des connaissances comme MASK (Ermine 2003) par exemple.
- **Entretien commun :** Une présentation de la démarche utilisée devrait être ensuite faite en commun. Il faut assurer que tous les participants comprennent l'objectif de l'entretien et les exigences que nous attendons de leurs parts.
- **Entretiens privés :** Chaque acteur métier (ou *un petit groupe d'acteurs*) a son propre point de vue. Après l'entretien en commun, des entretiens privés sont nécessaires. Lors de ces entretiens, nous essayons de comprendre les processus métiers, et les manières d'utiliser/ de traiter de données de chaque interviewé. Les questions à proposer peuvent être :
 - * *Comment les données sont nommées informellement ou autrement dit, quels sont les vocabulaires fréquemment utilisés pour appeler des données? La définition des termes?*
 - * *Quelle est la hiérarchie des vocabulaires (taxonomie)?*
 - * *Quelles sont les relations sémantiques entre les vocabulaires utilisés?*

Après l'entretien : Le compte rendu des entretiens est noté en utilisant une méthodologie de capitalisation des connaissances ou des logiciels spécifiques de mind-mapping tel que Free-mind⁵⁷. Un bilan est rédigé et diffusé à l'ensemble des participants. Si nécessaire, l'interview des experts peut être répété jusqu'à ce qu'un consensus soit trouvé.

3.2.2.1.2 Entretien des chercheurs du GIN

L'entretien des chercheurs au GIN a eu lieu du 10 et 11 Mars 2014 avec une dizaine de participants venant de différentes disciplines. La liste des participants et leurs rôles sont illustrés

57. http://freemind.sourceforge.net/wiki/index.php/Main_Page

No	Position	Données manipulées
1	Directeur de recherche	Imagerie cognitive
1	Directeur de groupe	Données génériques
1	Chercheur	Neuroimagerie cognitive, données comportementales et fonctionnelles
1	Ingénieur chercheur	Données comportementales, collecte de cohorte sur Internet, etc.
1	Chercheur	Données anatomiques
1	Chercheur	Fibres blanches, données anatomiques
1	Assistant Ingénieur	Recrutement de sujets et collection de données, MRI
1	Assistant Ingénieur	Traitement des données, MRI, Gestion de BDD du GIN

TABLE 3.3 : Description des participants à l'entretien.

dans le Tableau 3.3. Comme présenté ci-dessus, l'objectif de cet entretien est de comprendre le travail des chercheurs, leurs besoins en données, le rôle de chaque personne dans le système d'information et surtout les vocabulaires qu'ils utilisaient pour nommer les données au cours d'activités quotidiennes.

Comme résultat, concernant l'accès aux données, l'interview a montré que les chercheurs ne peuvent pas formuler des requêtes longues et complexes sans l'aide des techniciens de la BDD en raison du manque de compréhension de la structure de la base de données, de la sémantique des données et des relations entre elles. Dans certains cas, ils se doutent même que le technicien ajoute de nouvelles données dans la base de données en raison de l'énorme quantité d'interdépendances.

a) Recueil des données

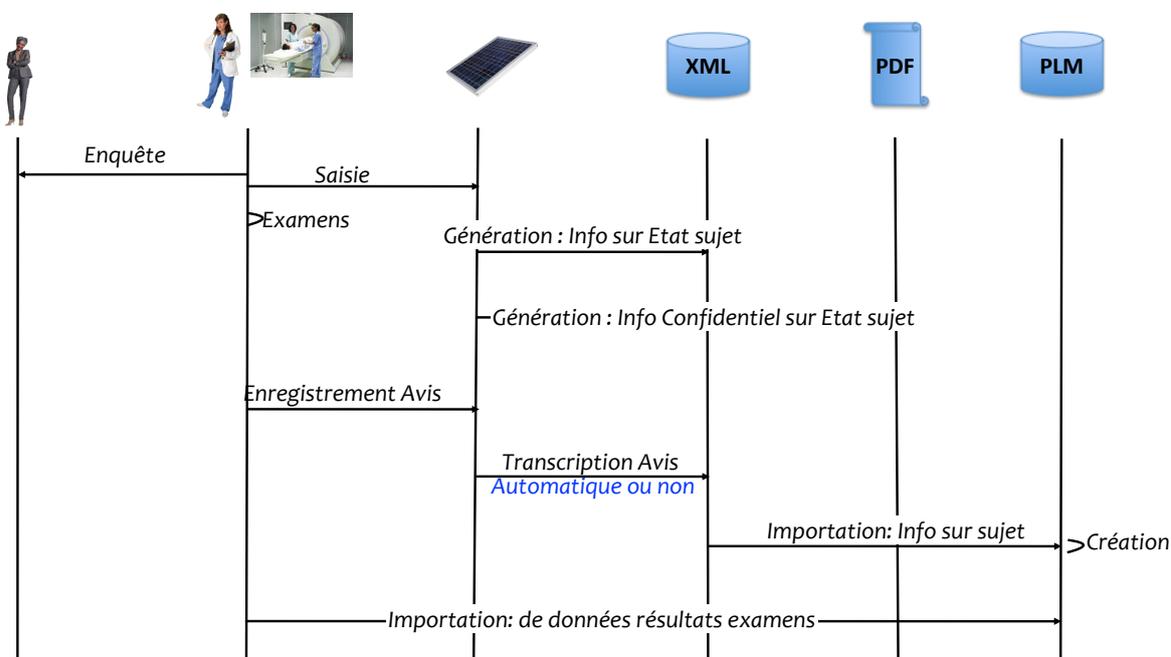


FIGURE 3.18 : Processus de recueil des données au GIN : Des questionnaires prédéfinis sur les tablettes sont utilisés pour recueillir des données statistiques sur les sujets. Les sujets passent ensuite des examens et acquisitions effectués par les spécialistes.

b) Exploitation des données

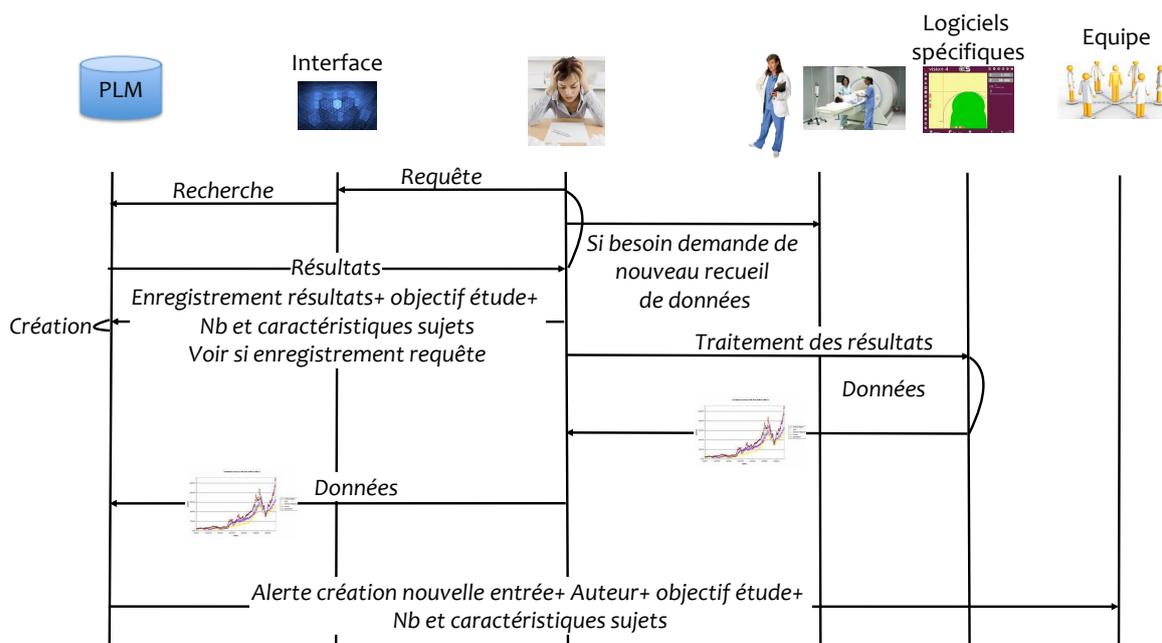


FIGURE 3.19 : Processus d'exploitation des données au GIN : Les chercheurs effectuent des recherches de données à travers une interface de requêtes. Ils font ensuite des traitements sur les données en utilisant des logiciels spécifiques.

c) Mise à jour des données

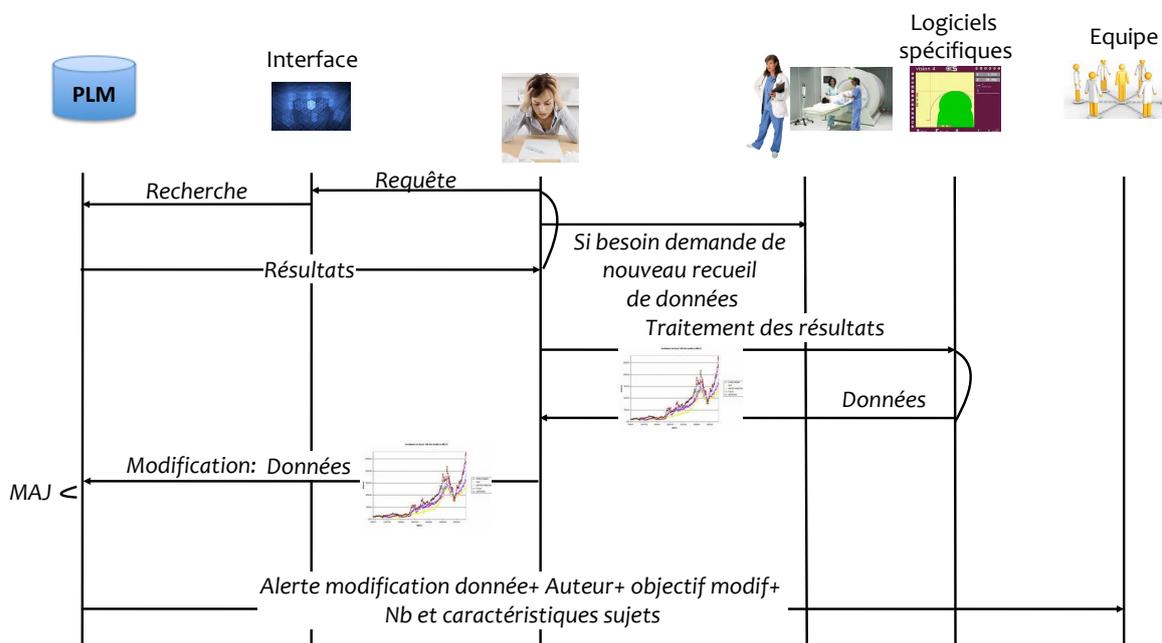


FIGURE 3.20 : Processus de mise à jour des données au GIN : Les nouvelles données (des données d'acquisition ou des résultats de traitement de données) sont ajoutées à la base de données PLM.

Nous avons identifié également le processus de recherche en Neuroimagerie : de l'acquisition de données jusqu'à la publication de résultats. Les Figures 3.18, Figure 3.19 et Figure 3.20 illustrent trois cas d'utilisation : **recueil des données**, **exploitation des données** et **mise à jour des données**. L'analyse de ces trois processus nous a permis d'identifier les relations sémantiques entre différents types de données : Une acquisition est effectuée sur un sujet

d'étude, les données délivrées sont obtenues au travers d'un ensemble des traitements, etc. Ces éléments sont indispensables pour la définition des ontologies d'application.

Nous avons noté que chaque chercheur utilise son propre vocabulaire pour appeler les données. Ces vocabulaires sont compréhensifs par certains chercheurs mais ils ne sont pas interprétables par les autres chercheurs au GIN ou ceux venant de l'extérieur.

3.2.2.2 Analyse des modèles conceptuels de données

Les BDDs (notamment les *BDDRs*) sont construites en trois étapes : construction du modèle conceptuel, construction du modèle logique et construction du modèle physique (voir plus la section 2.1.3). À haut niveau d'abstraction, le modèle conceptuel et le modèle logique souvent prennent en compte la logique du domaine et la sémantique de données qu'ils modélisent, ils sont alors une ressource précieuse pour la définition des ontologies d'application.

Nous pouvons réutiliser tout d'abord l'ensemble des entités prédéfinies dans le modèle d'Entité-Association comme une base pour la définition des concepts ontologiques. Chaque entité devient une classe ontologique (*un concept*) tandis que chaque attribut attaché à une entité devient sa propriété ("*DatatypeProperty*"). Toutes les relations entre les entités (relation forte ou faible) (Chen 1976) et leurs cardinalités correspondantes sont examinées. Chaque relation entre entités devient une objet propriété ("*ObjectProperty*") (voir section 2.3.4.3). Il existe de nombreux travaux dans la littérature (Fahad 2008, Chujai et al. 2014) portant sur les méthodologies de transformation des modèles Entité-Association en ontologies OWL qui peuvent être utilisés comme base pour la construction de notre ontologie.

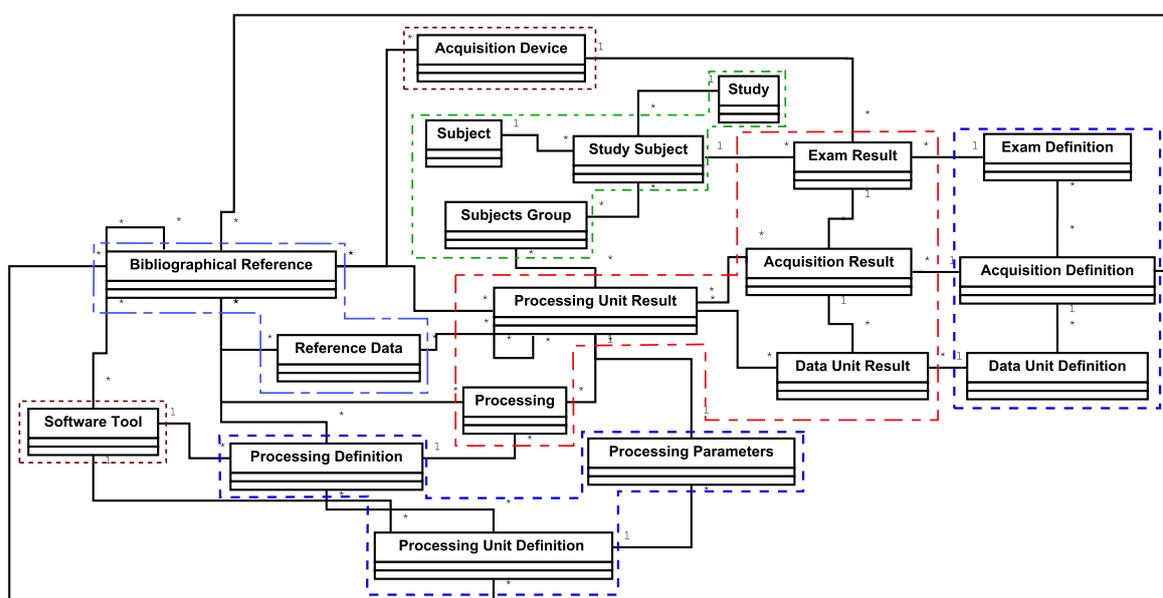


FIGURE 3.21 : Le modèle de données BMI-LM implémenté dans le système PLM Teamcenter (Allanic et al. 2013) avec la classification des objets métiers.

Analyse du MDD BMI-LM : Nous avons échangé plusieurs fois avec Marianne Allanic, l'auteur de BMI-LM, sur la conception du modèle de données et de la sémantique du domaine "GIN". Elle nous a expliqué les significations des objets métiers et des relations entre eux.

Objet	Description
Acquisition	Période indivisible d'acquisition de données
Acquisition Definition	Description d'un protocole d'acquisition
Acquisition Device	Description d'un dispositif utilisé pendant un examen
Bibliographical Reference	Article scientifique
Data Unit Result	Donnée acquise isolée
Data Unit Definition	Description d'une unité de données
Exam	Ligne continue d'acquisitions
Exam Definition	Description de la chaîne d'acquisitions
Subjects Group	Ensemble de sujets dans l'étude regroupés selon un critère
Processing Definition	Description d'une chaîne de traitement
Processing	Chaîne de traitement
Processing Parameter	Jeu de paramètres utilisés pour un traitement
Processing Unit Description	Description d'une unité de traitement
Processing Unit Result	Traitement effectué sur des données
Reference data	Donnée d'entrée d'un traitement hors du contexte d'une étude
Software Tool	Description d'un logiciel de traitement
Study	Etude de recherche
Study Subject	Sujet dans le contexte d'une étude
Subject	Sujet unique dans la base

TABLE 3.4 : Les descriptions des objets métiers du MDD BMI-LM (Allanic 2015).

Nous avons consulté également la description détaillée de BMI-LM présentée dans son manuscrit de thèse (Allanic 2015), ce qui justifie pourquoi "tels objets métiers" sont sélectionnés et "telles relations" sont utilisées. La Figure 3.3.2 illustre le diagramme UML du MDD BMI-LM utilisé pour le système PLM au GIN. Après des mises à jour pour satisfaire aux tests de validation, la version actuelle contient **19** objets métiers ("*business objects*") qui sont décrits dans le Tableau 3.4. Ces derniers pourraient être classés en **cinq** catégories différentes en fonction de la nature de données qu'ils représentent. Cette classification est ensuite utilisée pour la définition des concepts généraux de notre ontologie d'application.

- i) **Result** : Acquisition, Examen, DataUnit, Processing, ProcessingUnit.
- ii) **Definition** : Acquisition Definition, Examen Definition, Processing Definition, DataUnit Definition, Processing Parameter.
- iii) **Tool** : Acquisition Device and Software Tool.
- iv) **Source** : Study, Subject, Study Subject, Subject Group.
- v) **Reference** : Reference Bibliography, Reference Data.

Nous analysons également toutes les relations entre les objets métiers BMI-LM ainsi que leurs cardinalités. Ces relations sont conformes à trois cas d'utilisation de données présentés ci-dessus. Des relations sémantiques entre différentes classes de données peuvent être identifiées rapidement, comme par exemple : Les *données d'acquisitions* sont **obtenues** à travers des *examens/ tests* sur les *sujets* qui **participent** à une *étude*, les *chercheurs* **utilisent** des *outils/ logiciels spécifiques* pour **analyser** des *données d'acquisition*, afin d'**obtenir** des *résultats de traitement*. Ces premières analyses nous permettent d'identifier des relations entre différents concepts de l'ontologie à construire.

Note : Les documents, rapports produits à partir des activités quotidiennes des experts sont

aussi une ressource importante à étudier. Ces documentations contiennent des vocabulaires fréquemment utilisés du domaine, qui peuvent être les termes ontologiques "pertinents".

3.2.2.3 Réutilisation des ontologies existantes

La réutilisation des ontologies existantes est nécessaire pour réduire le coût de conception et pour assurer que les connaissances représentées dans l'ontologie construite sont consensuelles. Différentes applications du domaine modélisent une partie commune du domaine. Une ontologie d'application peut être construite à base des autres ontologies d'application déjà existantes. Pour le cas d'études au GIN, les concepts ontologiques sont identifiés à l'aide de l'approche top-down, à côté des résultats de l'entretien avec les chercheurs au GIN et de l'analyse du modèle de données BMI-LM, nous avons étudié aussi certaines ontologies noyaux (applicables pour tout domaine tels que SUMO⁵⁸, BFO⁵⁹, UFO⁶⁰, DOLCE⁶¹) et ontologies du domaine qui ont été proposées pour la communauté de Neuroimagerie comme (*ontovip* (Gibaud et al. 2014), *ontoSPEC* (Kassel 2005), *ontoNeuroLog* (Temal et al. 2008), *QIBO* (Buckler et al. 2013), etc.). Après avoir analysé ces ontologies, nous avons réutilisé uniquement les concepts qui sont propres aux données BMI du GIN comme : Traitement de dataset, acquisition de dataset, sujets, etc. (Figure 3.22).

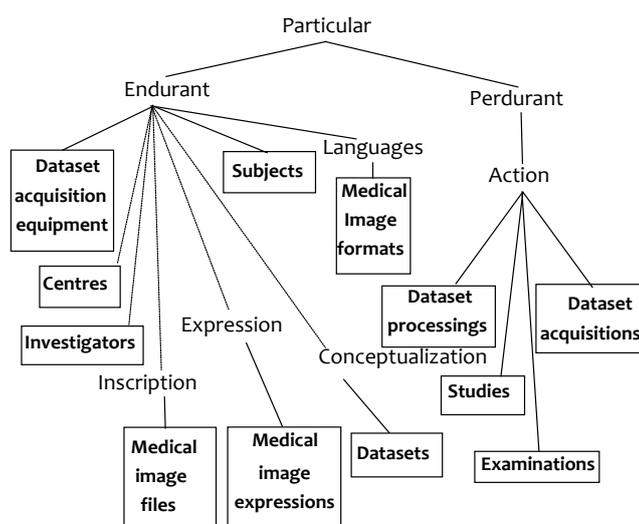


FIGURE 3.22 : Top concepts ontologiques de l'ontoVIP - Project Credible⁶².

3.2.3 Modélisation de connaissances

3.2.3.1 Démarche de modélisation

La modélisation des connaissances permet tout d'abord de sélectionner parmi l'ensemble des concepts et des relations identifiés dans l'étape "acquisition des connaissances" ceux qui sont les plus appropriés, et de définir explicitement des notions des concepts et des rela-

58. <http://www.adampease.org/OP/>

59. <http://ifomis.uni-saarland.de/bfo/>

60. <https://oxygen.informatik.tu-cottbus.de/drupal7/ufo/>

61. <http://www.loa.istc.cnr.it/old/DOLCE.html>

62. <https://credible.i3s.-unice.fr/>

tions. Nous déterminons ensuite une hiérarchie entre les concepts initiaux, ajoutons des nouveaux concepts plus génériques ou plus spécifiques pour compléter cette hiérarchie. Les sous-concepts ont tous les attributs du concept parent avec leur propres attributs. Puis, nous ajoutons des nouvelles relations reliant les concepts initiaux aux nouveaux concepts qui viennent d'être ajoutés. Enfin, les termes utilisés pour nommer les concepts et les relations devraient être modifiés pour qu'ils soient suffisamment abstraits et compréhensifs par les acteurs métiers cibles.

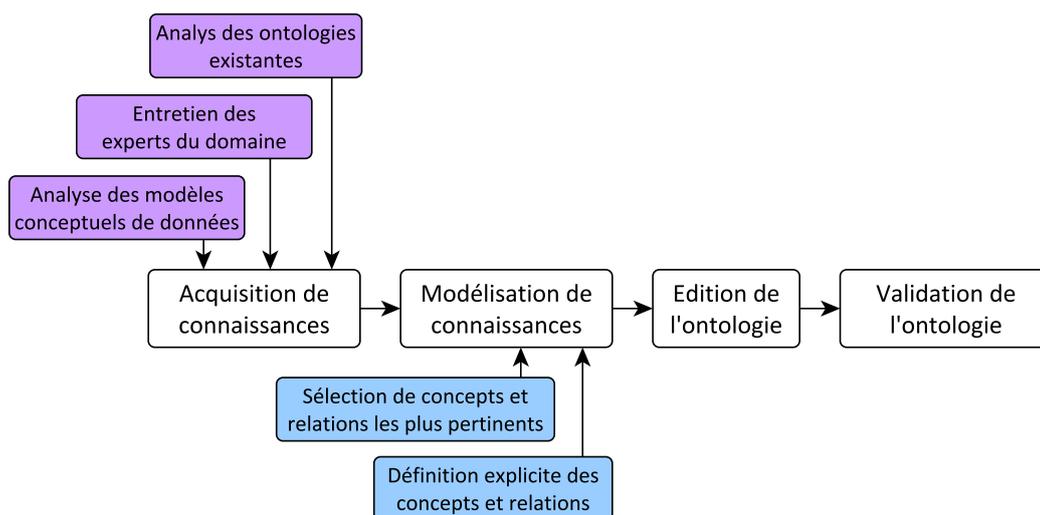


FIGURE 3.23 : La modélisation de connaissances pour la construction des ontologies d'applications.

3.2.3.2 Modélisation de l'ontologie d'application GIN

Basé sur cinq catégories des objets métiers présentés ci-dessus, la Figure 3.24 illustre l'arbre des concepts ontologiques proposés pour les chercheurs GIN. Ces concepts sont divisés en six catégories :

- i) **Data** : Acquired Data, Reference Data, Processed Data, Reference Bibliography.
- ii) **Protocol** : Acquisition Definition, Processing Definition.
- iii) **Process** : Acquisition, Processing.

CONCEPTS			Object métier	Objets métiers (nom dans la BDD)
rootConcept				
data	acquired-data		Exam Result	GIN4_ExamRes
			Dataunit	GIN4_DataUnit
			Acquisition Result	GIN4_Acquisition
		processed-data	Processing Unit Result	GIN4_ProcessUR
			Processing Result	GIN4_ProcessRes
	reference	reference-bibliography	Bibliography Reference	GIN4_RefBib
		reference-data	Reference Data	GIN4_Template
	investigator	acquisition-executive		
		processing-executive		
	process	acquisition		
processing				
protocol	acquisition-protocol		Examination Definition	GIN4_ExamDef
			Acquisition Definition	GIN4_AcqDef
			Dataunit Definition	GIN4_DataUnitDef
	processing-protocol		Processing Definition	GIN4_ProcessDef
			Processing Unit Definition	GIN4_ProcessUD
		Processing Parameter	GIN4_ProcessPar	
source	study		Study	GIN4_Study
	subject		Study Subject	GIN4_StudySub
tool	acquisition-tool		Acquisition Device	GIN4_Scanner
	processing-tool		Processing Tool	GIN4_Tool

TABLE 3.5 : Comparaison des vocabulaires utilisés dans le modèle de données BMI-LM et l'ontologie d'application proposée.

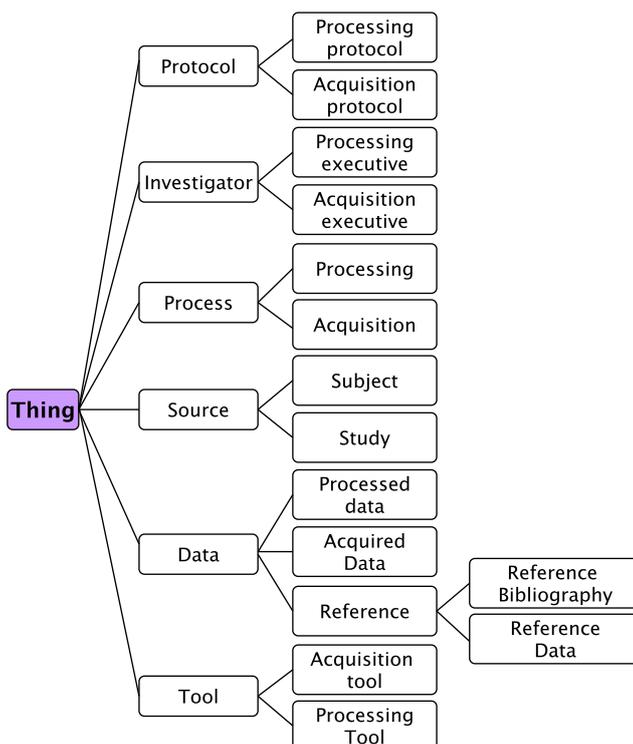
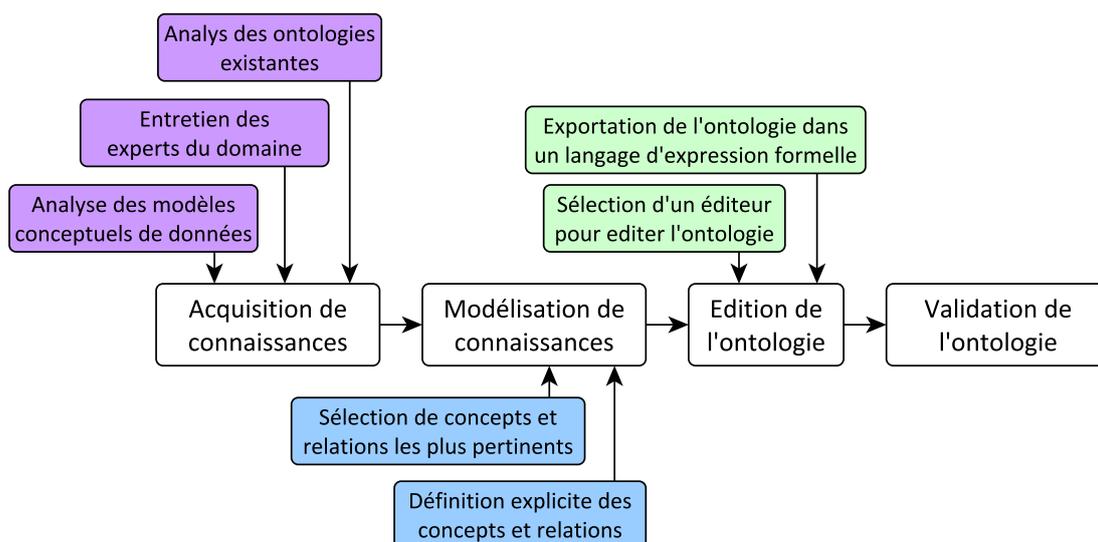


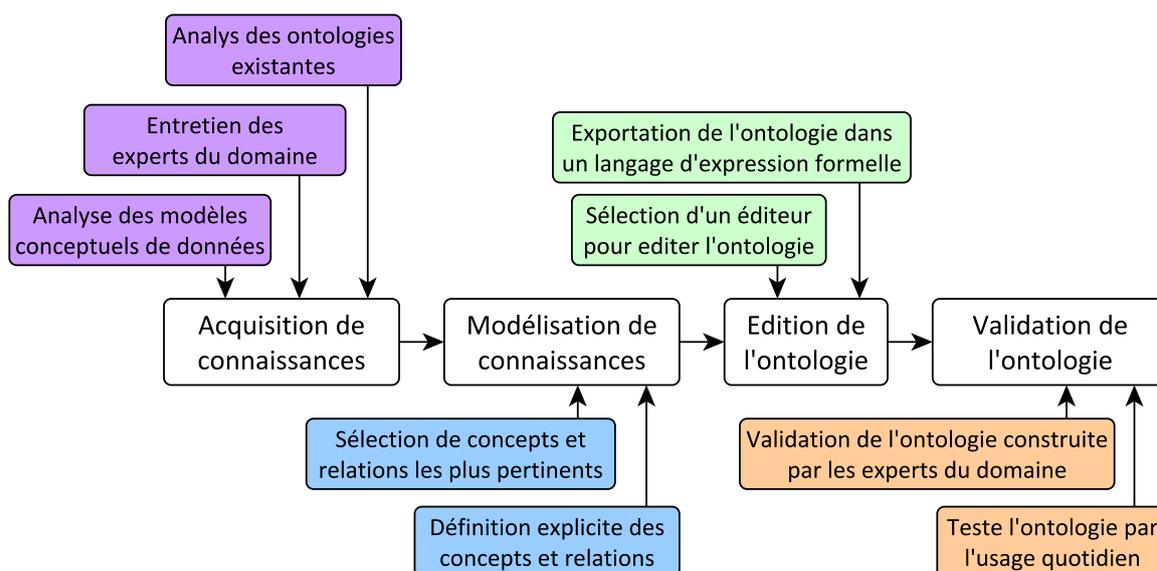
FIGURE 3.24 : L'arbre des concepts ontologiques proposés pour les chercheurs au GIN.

FIGURE 3.26 : *Edition de l'ontologie en utilisant des logiciels d'éditeurs.*

Des logiciels d'édition d'ontologie tels que Protégé⁶⁶, NeOn⁶⁷, Vitro⁶⁸, etc. sont recommandés à utiliser pour limiter les erreurs. Ces logiciels permettent d'éditer et de construire les ontologies dans une interface visuelle et aisée. Il est possible d'exporter les ontologies éditées sous différents formats tels que *OWL* ou *RDF/XML*.

Dans le cas d'études GIN, nous avons utilisés le logiciel Protégé pour éditer des ontologies car il est supporté par plusieurs plateformes. Les ontologies sont ensuite exportées en langage *OWL* exploitant la puissance de l'inférence pour générer des requêtes.

3.2.5 Validation de l'ontologie

FIGURE 3.27 : *La validation de l'ontologie construite par les experts du domaine et l'usage quotidien.*66. <http://protege.stanford.edu/>67. http://neon-toolkit.org/wiki/Main_Page.html68. <http://vitro.mannlib.cornell.edu/>

La validation des connaissances est une étape importante. Les ontologies visent à fournir une représentation consensuelle du domaine, elles doivent être acceptées et utilisées largement par la communauté. Les ontologies proposées doivent donc être testées par l'utilisation quotidienne des acteurs métiers du domaine. Dans le cas où des ontologies d'application sont utilisées pour améliorer l'interrogation de données, il faut les tester avec un ensemble des requêtes utilisateurs, de simples à complexes (*longues* et *conjonctives*) et vérifier si les résultats de ces requêtes peuvent satisfaire aux besoins d'information des utilisateurs. Si les résultats ne sont pas bons ou incomplets, des modifications sont nécessaires.

Il faut considérer également l'aspect "**usage**". Les ontologies d'application implémentées doivent fournir aux acteurs métiers une utilisation suffisamment aisée pour qu'ils puissent interroger eux-mêmes les données de base. L'implémentation des ontologies applications est validée et justifiée seulement si les données sont utilisées plus efficacement.

3.3 LA DÉMARCHE DE CONSTRUCTION DES MAPPINGS

Cette section présente tout d'abord des éléments bibliographiques portant sur le mapping entre les ontologies et données (les *typologies* et les *langages du mapping*). Cette démarche détaille la construction des mappings liant les ontologies d'application construites dans la section précédente avec les schémas des sources de données.

3.3.1 Typologies du mapping

La construction du mapping entre l'ontologie et les BDDs dépend de la relation entre elles. En général, selon Barrasa ([Barrasa Rodríguez et al. 2003](#)), cette relation peut être selon deux cas :

Composants de la BDD	Composants de l'ontologie
Table (T)	RDFS Classes (C)
Ligne (i) de la table T	Une node (n) de la classe C
Colonne (j) de la table T	Attribut (m) de la class C
Value à cellule (i,j)	Value de la propriété (n,m)

TABLE 3.6 : "Table to class, column-to-predicate" Mapping direct entre les composants de la BDD et ceux de l'ontologie.

- **Mapping direct** (*Direct Mapping*) : L'ontologie était construite semi automatiquement à partir de schémas des BDDs. Les mappings entre les BDDs et l'ontologie générée sont donc assez directs en raison des similarités entre les deux.
- Mapping piloté par la sémantique du domaine ou **Mapping manuel** (*Domain Semantics-Driven Mapping*) : L'ontologie existe indépendamment de schémas des BDDs. Les mappings sont plus complexes et peuvent être incomplets (Figure 3.30).

3.3.1.1 Création de l'ontologie à partir de la BDD (Mapping direct - BDD Schéma Ontologie)

Dans cette approche, l'ontologie est construite à partir du modèle relationnel de la BDD (le *schéma de données*). La construction de l'ontologie est directe. Elle peut impliquer simplement

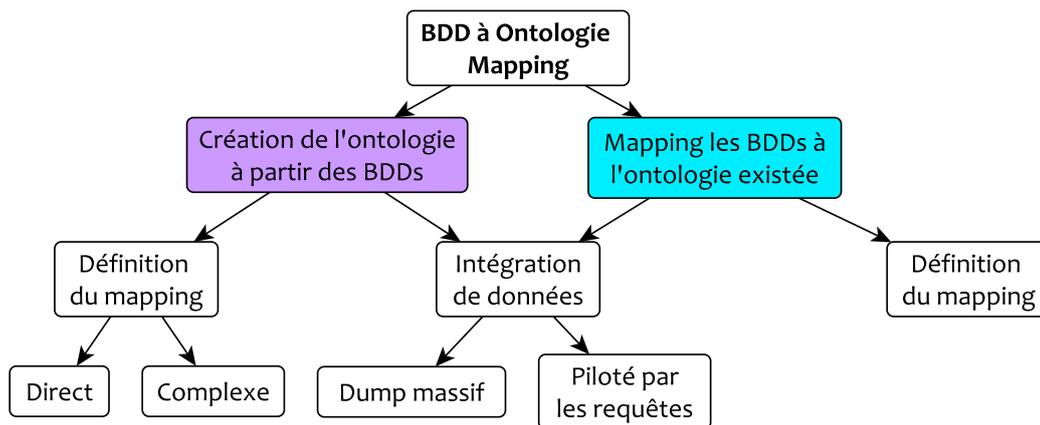


FIGURE 3.28 : La classification des approches utilisées pour mapper d'une BDD à une ontologie (Ghawi et Cullot 2007).

des transformations directes de tables de BDD en concepts ontologiques et des colonnes en propriétés (Ghawi et Cullot 2007). L'ontologie générée est considérée comme une ontologie d'application, qui est utilisable pour les BDDs et les applications dédiées. Les mapping sont donc juste les correspondants entre composants ontologiques (*concepts, propriétés, etc.*) et composants de la BDD originale (*tableau, colonne, etc.*) (Tableau 3.6).

3.3.1.2 Mapping d'une base de données à une ontologie existante

Bien que la BDDR contiennent des données structurées et son modèle de données (Modèle ER) apporte des sens sémantiques très précieuse, les BDDs sont rarement une bonne description sémantique d'un domaine, autrement dit, l'ontologie générée par la transformation directe schéma d'une BDD est souvent insuffisamment sémantique pour représenter le domaine donné.

La raison est simple. À cause des processus d'optimisation et des maintenances conditionnelles, la sémantique initiale dans le modèle conceptuel de la BDD n'est pas prise en compte dans le modèle logique. Le schéma lui-même ne supporte donc pas une description complète du domaine, d'autre relations et concepts potentiels sont cachés dans la BDD ou existent dans une autre BDD (Dolbear et Goodwin 2007, Michel et al. 2014).

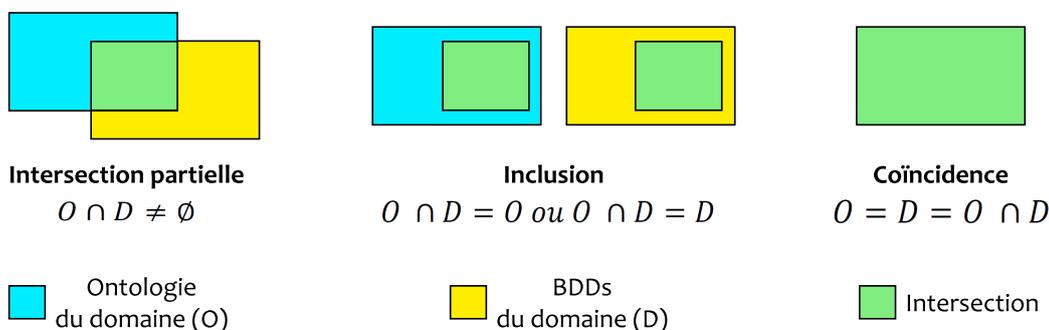


FIGURE 3.29 : Niveaux d'intersection entre les domaines couverts par le DB et l'ontologie (Barrasa et al. 2003).

Les mappings entre eux (BDDR et MDD) sont donc plus complexes, car il y a éventuellement

différents niveaux de chevauchement entre le domaine de la BDD et celui de l'ontologie. Ces domaines ne peuvent pas coïncider et les critères de modélisation utilisés pour concevoir des BDDs sont différents de ceux utilisés pour concevoir l'ontologie (Barrasa et al. 2003).

3.3.2 Langages du mapping

Plusieurs langages ont été proposés pour représenter les mappings entre la BDD et l'ontologie. Car, on peut utiliser OWL ou RDF(s) pour exprimer une ontologie, les langages du mapping sont souvent appelés RDB2RDF or RBD2OWL langages. Dans ce manuscrit, nous utilisons "RDB2RDF langages" pour parler de langages du mapping entre la BDD et l'ontologie.

3.3.2.1 Classification des langages

Michel et al. (Michel et al. 2014) ont classifié "RDB2RDF langages" en trois catégories basées sur la manière de :

- **Décrire le mapping** (*Mapping description*).
- **D'implémenter le mapping** (*Mapping implementation*).
- **D'utiliser des méthodes pour récupérer des données** (*Data retrieval methods*).

En même temps, Herts (Hert et al. 2011) a classifié les langages de mappings en quatre catégories différentes (Figure 3.31) :

- **Mapping direct** (*direct mapping*)
- **Usage général lecture-seul mapping** (*read-only general-purpose mapping*)
- **Usage général lecture-écriture mapping** (*read-write general-purpose mapping*)
- **Usage spécifique mapping** (*special-purpose mapping*).

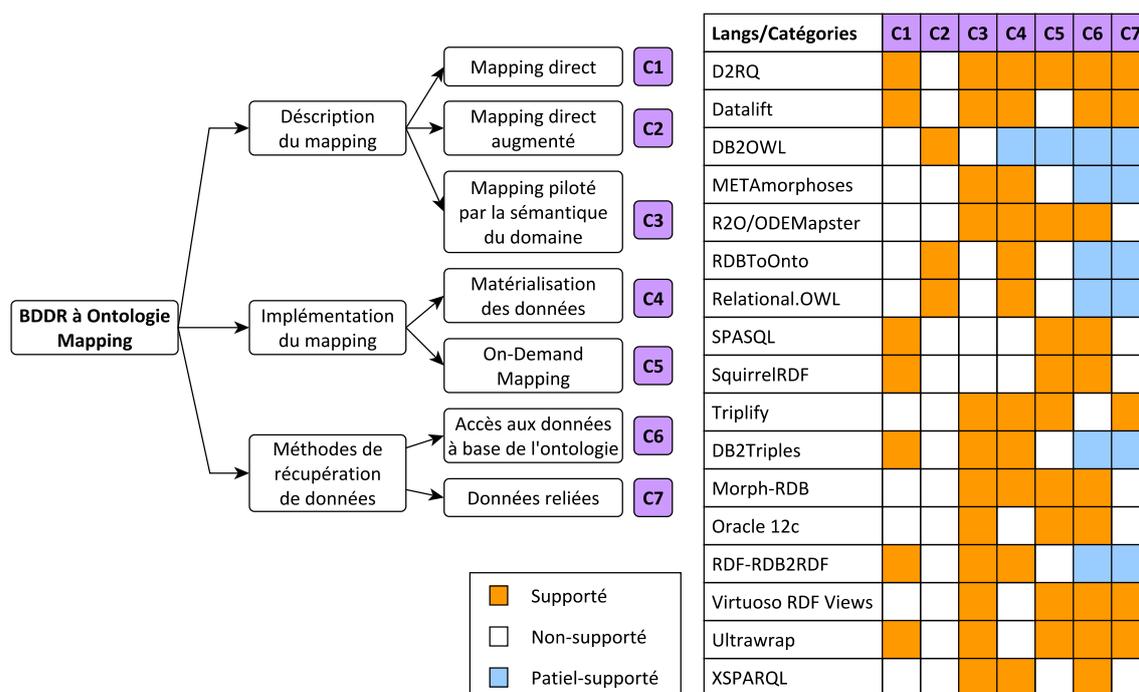


FIGURE 3.30 : Classification de RDB2RDF langages - Michel (Michel et al. 2014).

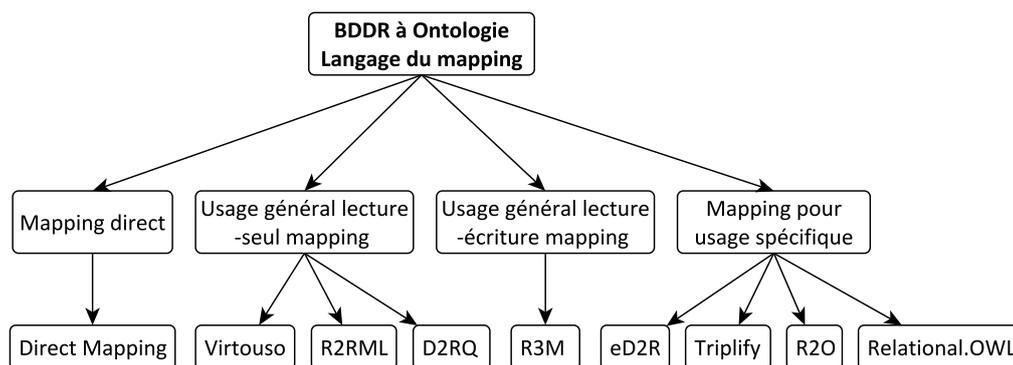


FIGURE 3.31 : Classification de RDB2RDF langages - Hert (Hert et al. 2011).

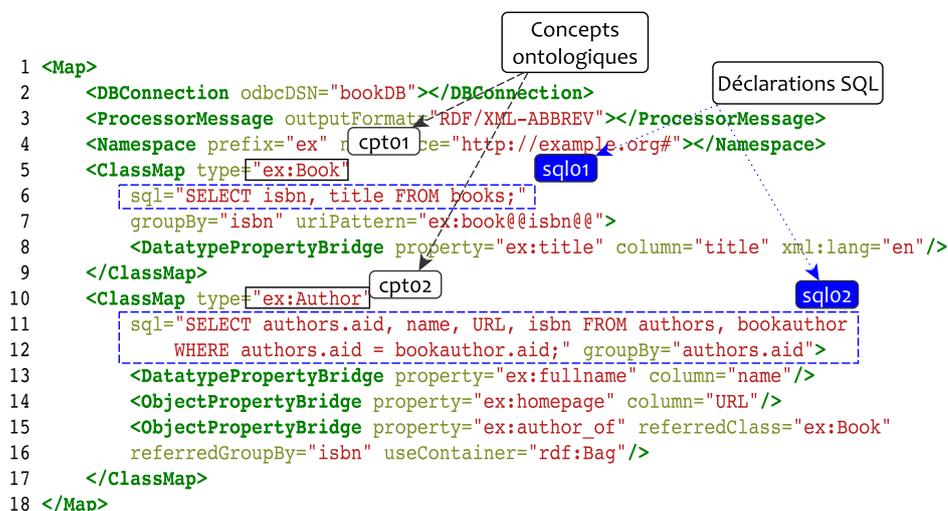


FIGURE 3.32 : Un exemple du D2R (Bizer 2003).

3.3.2.2 Liste de langages du mapping

Nous présentons ici une liste des langages de mappings chronologiquement selon leur création. Certains langages sont présentés en détail, pour les autres, les auteurs n'ont présenté que les idées et des principes de base.

- D2R** (Bizer 2003) : D2R est un langage déclaratif basé sur une syntaxe XML, qui est utilisé pour représenter les mappings entre les concepts ontologiques et les déclarations SQL correspondantes. D2R utilise directement les déclarations SQL dans les règles du mapping. La Figure 3.32 illustre un exemple du mapping. Deux concept "Book" (cpt01) et "Author" (cpt02) sont mappés à deux déclarations SQL (sql01, sql02).
- D2RQ**⁶⁹ (Bizer et Seaborne 2004) : D2RQ est un langage déclaratif qui considère les BDDs non-RDF comme un graphe RDF virtuel. Il est basé sur D2R (Bizer 2003) présenté au-dessus. L'objet central du D2RQ est "**ClassMap**" qui représente une classe ou un groupe de classes de l'ontologie. Chaque "**ClassMap**" possède un ensemble de ponts de propriétés (*Property Bridges*) qui décrit comment les propriétés d'instance sont créées

69. <http://wifo5-03.informatik.uni-mannheim.de/bizer/d2rq/spec/20090810/>

et comment des URI (*Uniform Resource Identifier*) sont inversés dans des valeurs de la BDD (Bizer et Seaborne 2004).

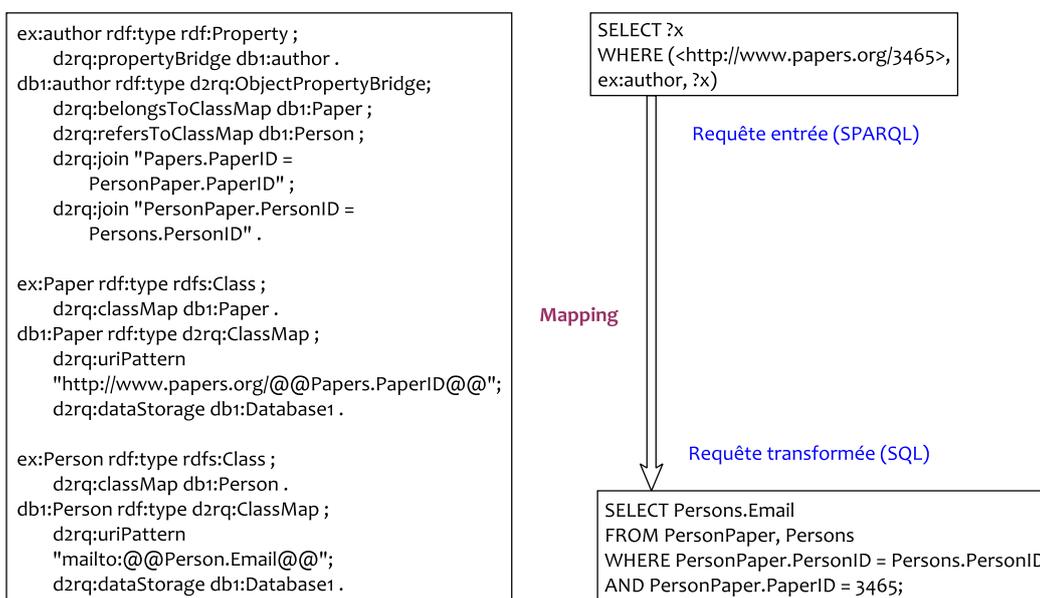


FIGURE 3.33 : La transformation de requête SPARQL en SQL grâce au mapping exprimé en D2RQ (Bizer et Seaborne 2004).

c) **R2O** (Barrasa et al. 2003) :

R2O est un langage extensible et déclaratif pour décrire les mappings entre les schémas relationnels de BDD et les ontologies représentées en RDF(s) ou OWL. R2O est applicable dans le cas où la similarité entre l'ontologie et les modèles de la BDD est faible.

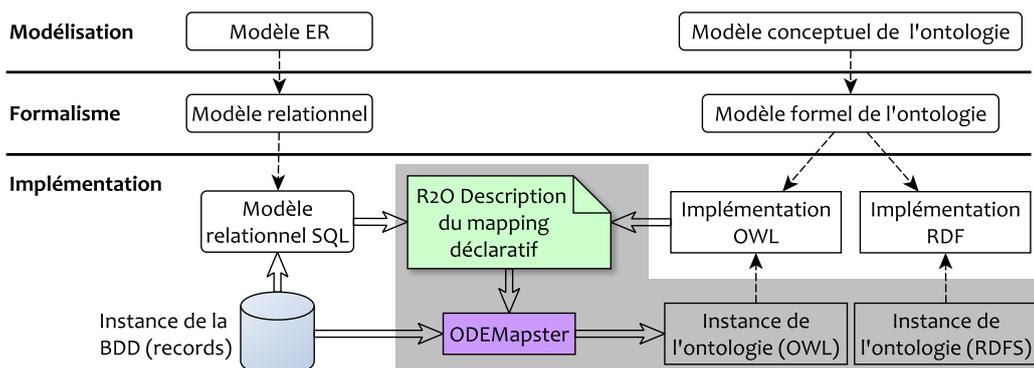


FIGURE 3.34 : Architecture de mapping avec R2O (Barrasa et al. 2003).

Selon Barrasa (Barrasa et al. 2003), les caractéristiques principales du R2O sont :

- **R2O** précise comment les instances de l'ontologie sont créées à partir de données dans de la BDD. R2O peut être utilisé pour compléter automatiquement une ontologie avec des instances extraites du contenu de la base de données.
- **R2O** peut être utilisé pour exprimer des mappings générés par des outils de recherche automatiquement.

- **R2O** peut être utilisé pour l'**auto-vérification**. En raison de son caractère pleinement déclaratif, les incohérences et les ambiguïtés dans les mappings peuvent être détectées automatiquement.
- Les mappings exprimés en **R2O** peuvent également être utilisés pour vérifier l'**intégrité des différentes parties d'une BDD** en fonction d'une ontologie, en appliquant les axiomes de l'ontologie aux éléments de la BDD.
- Les mappings exprimés en **R2O** peuvent être utilisés pour **caractériser automatiquement les sources de données** pour permettre la distribution dynamique de requêtes dans des approches d'**intégration d'informations intelligentes**.

Cependant, la transformation définie en R2O n'est pas bidirectionnelle. Les éléments de la BDD sont transformés en composants de l'ontologie, mais pas inversement. De plus, les mappings exprimés en R2O sont difficilement interprétés par l'humain.

d) **Relational.OWL** ((de Laborda et Conrad 2005) : Relational.OWL utilise OWL pour représenter les schémas des BDDs afin de faciliter la communication entre différents BDDs. Dans ce cas-là, il est considéré comme un format explicite d'échange et un outil d'interopérabilité de différents schémas des BDDs. Différents éléments de BDD sont inclus dans l'ontologie Relational.OWL :

- **Tables et colonnes**
- **Clés primaires et étrangères**
- **Types de données**

Comme l'auteur le mentionne (de Laborda et Conrad 2005), des métadonnées telles que "*index, déclencheurs, tablespace*" ne sont pas incluses dans l'ontologie Relation.OWL. Cependant, la représentation avec les trois métadonnées au-dessus est suffisamment expressive pour la plupart de BDDs existante.

e) **Direct mapping**⁷⁰ (Hu et Qu 2007) : Cette approche est proposée depuis longtemps par Berners-Lee (Berners-Lee 1998). Il mappe directement les tables relationnelles aux classes et les colonnes des tables aux propriétés de l'ontologie (RDF). Pour cette raison, elle est aussi appelée "*table to class, attribute to property mapping*". Les vocabulaires utilisés dans l'ontologie RDF ainsi que les **Uniform Resource Identifier (URI)** des instances de l'ontologie sont extraits à partir du schéma de la BDD. Les principes à base de la transformation :

- **Table-à-classe** (*table-to-class*) : Une table est traduite à la classe ontologique identifiée par un URI suivant la construction : "*namespace/database/table*".
- **Colonne-à-propriété** (*column-to-property*) : Chaque colonne de la table SQL est traduite à la propriété ontologique identifiée par un URI suivant la construction : "*namespace/database/table/column*".
- **Ligne-à-ressource** (*row-to-resource*) : Chaque ligne d'une table est traduite à une ressource dont le type est la classe représentant la table et dont un URI formulé en utilisant la clé primaire de la table : "*namespace/database/table/primaryKey*".

70. <http://www.w3.org/DesignIssues/RDB-RDF.html>

Définition d'une table SQL dans l'ontologie avec Relational.OWL

```

1 <owl:Class rdf:ID="ADDRESS">
2   <rdf:type rdf:resource="&dbns;Table"/>
3   <dbns:hasColumn rdf:resource="#ADDRESS.ADDRESSID"/>
4   <dbns:hasColumn rdf:resource="#ADDRESS.STREET"/>
5   <dbns:hasColumn rdf:resource="#ADDRESS.ZIP"/>
6   <dbns:hasColumn rdf:resource="#ADDRESS.CITY"/>
7   <dbns:hasColumn rdf:resource="#ADDRESS.COUNTRYID"/>
8   <dbns:isIdentifiedBy>
9     <dbns:PrimaryKey>
10      <dbns:hasColumn rdf:resource="#ADDRESS.ADDRESSID"/>
11    </dbns:PrimaryKey>
12 </dbns:isIdentifiedBy>
13 </owl:Class>
14
15 <owl:DatatypeProperty rdf:ID="de l'attribut.ZIP">
16   <rdf:type rdf:resource="&dbns;Column"/>
17   <rdfs:domain rdf:resource="#ADDRESS"/>
18   <rdfs:range rdf:resource="&xsd:string"/>
19   <dbns:length>8</dbns:length>
20 </owl:DatatypeProperty>
21
22 <owl:DatatypeProperty rdf:ID="ADDRESS.COUNTRYID">
23   <rdf:type rdf:resource="&dbns;Column"/>
24   <rdfs:domain rdf:resource="#ADDRESS"/>
25   <dbns:references rdf:resource="#COUNTRY.COUNTRYID"/>
26   <rdfs:range rdf:resource="&xsd;integer"/>
27 </owl:DatatypeProperty>

```

Annotations dans l'image :
- Une boîte violette pointe vers la ligne 1 : "Définition d'une table SQL dans l'ontologie avec Relational.OWL".
- Une boîte bleue encadre les lignes 3 à 7.
- Une boîte verte pointe vers la ligne 10 : "définition de l'attribut".
- Une boîte verte pointe vers la ligne 15 : "définition".

FIGURE 3.35 : Expression de la table "Address" contenant de cinq colonnes (AddressID, Street, Zip, City, CountryId) comme une classe de l'ontologie Relational.OWL (de Laborda et Conrad 2005).

- **Cellule-à-valeur_littérale** (*cell-to-literal_value*) : Chaque cellule avec une valeur littérale est traduite en valeur d'une propriété de données (datatype property).
 - **Cellule-à-ressource_URI** (*cell-to-resource_URI*) : Chaque cellule avec une clé étrangère est traduite en un URI qui est une valeur d'une propriété (object-property).
- f) **Virtouso**⁷¹ (Erling et Mikhailov 2009) Virtouso est utilisé pour générer la vue sémantique RDF de données relationnelles pour les exposer sur le Web Sémantique. Virtouso contient un langage déclaratif de méta-schéma pour la définition des mappings entre les données SQL et les vocabulaires RDF préexistants. En général, Virtouso transforme l'ensemble de résultats de la requête SELECT SQL en l'ensemble de triples RDF (Hert et al. 2011).
- g) **Triplify** (Auer et al. 2009) Triply est utilisé pour publier Linked Data depuis les BDDs relationnelles. "Triplify est basé sur le mapping des requêtes HTTP-URI aux requêtes sur les BDDs relationnelles. Triplify transforme ensuite les relations obtenues en déclarations RDF et publient diverses sérialisations RDF" (Auer et al. 2009).
- h) **R3M** (Hert et al. 2010) R3M est utilisé pour mapper les données relationnelles au vocabulaire RDFs. Il supporte l'accès bidirectionnel basé sur RDF aux données relationnelles : les deux fonctions "lecture" et "écriture" sont supportées. La mise à jour et

71. <https://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html>

la simplicité étaient deux des principaux objectifs de la conception de ce langage de mapping. En se basant sur les principes du mapping direct, **R3M** mappe les tables SQL aux classes ontologiques, les attributs aux propriétés. Un mapping complet contient les mappings de la BDD (*DatabaseMap*), des tables (*TableMap*), des attributs (*AttributeMap*), des tables connectées (*LinkTableMap*). La Figure 3.36 illustre un exemple du mapping R3M.

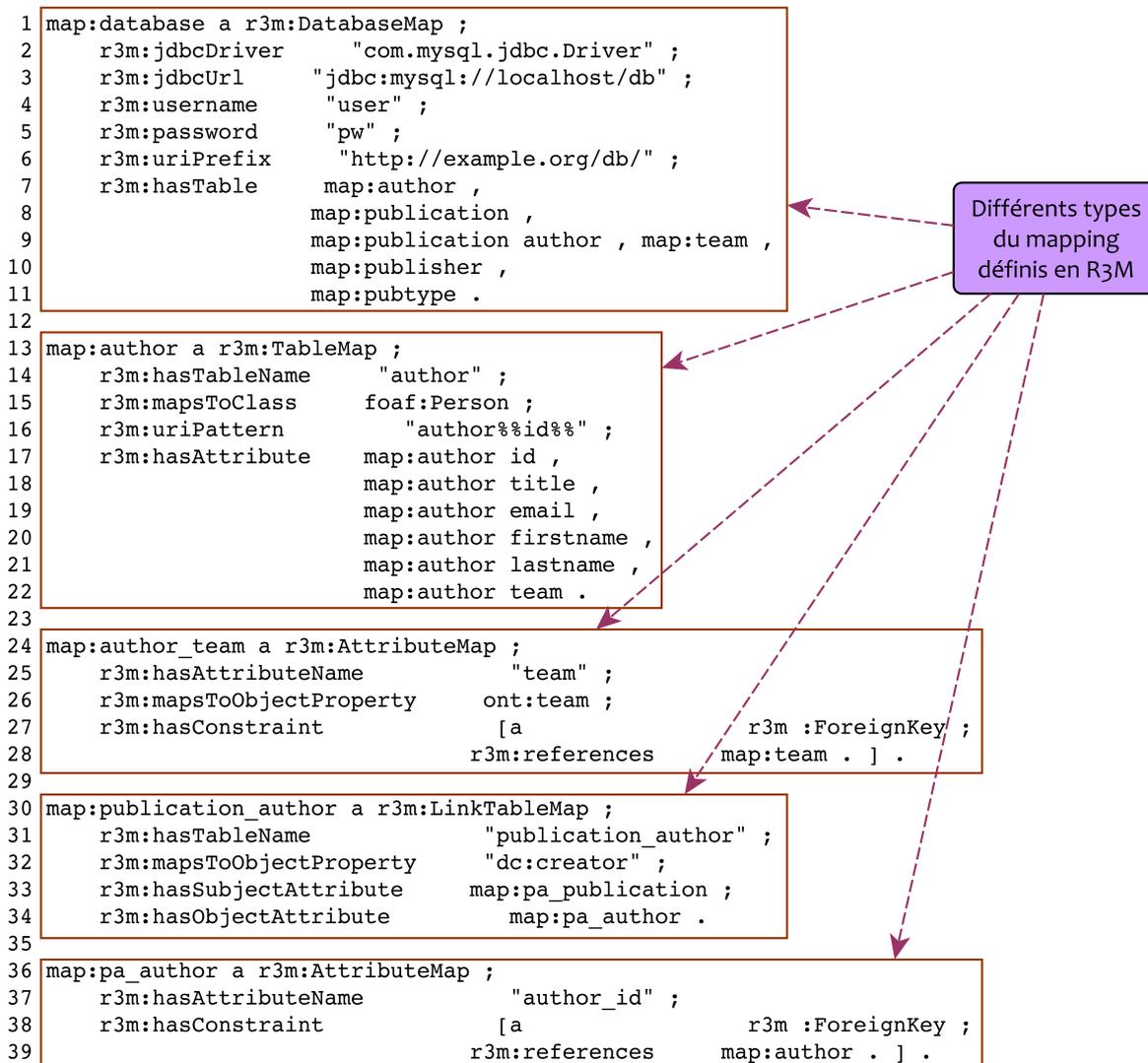


FIGURE 3.36 : Exemple du mapping R3M (Hert et al. 2010).

i) **RDF to RDF Mapping Language (R2RML)** (Michel et al. 2014, Das et al. 2012) : **R2RML** est le langage du mapping le plus connu qui est proposé par le W3C RDB2RDF Working Group. L'objectif principal de ce groupe est de fournir des "must-have" standards, exigences et spécialisations d'un langage du mapping. Les caractéristiques à base du **R2RML** sont :

- Supporter le mapping directe et le mapping piloté par la sémantique du domaine.
- Fournir une quantité suffisante d'information pour le mapping "on-demand" (la transformation de requêtes SPARQL en requêtes SQL et la matérialisation de données).

- Supporter la génération des URIs, la conversion des types de donnée (*DataTypes*), le renommage de colonnes, etc.

Un document de mapping R2RML est un graphe de mapping R2RML qui est écrit en RDF avec la syntaxe Turtle. Chaque mapping contient des TriplesMaps qui spécifie comment mapper des lignes d'une table logique aux triples RDFs. La table logique peut être une table, une vue SQL or le résultat d'une requête SQL. La Figure 3.37 illustre un exemple du mapping R2RML. Il contient deux TriplesMap "*Course*" et "*Participates*" avec leurs correspondantes "*PredicateObjectMaps*".

```

1 @prefix rr: <http://www.w3.org/ns/r2rml#>.
2 @prefix co: <http://courses.org#>.
3 @base <http://example.com/base#>.
4
5 <Course >
6   rr: logicalTable [ rr: tableName "COURSE" ];
7   rr: subjectMap [
8     rr: template "http://univ.edu/course/{ID}";
9     rr: class co:Course;
10  ];
11  rr: predicateObjectMap [
12    rr: predicate co:title;
13    rr: objectMap [ rr: column "TITLE" ];
14  ].
15
16
17 <Participates >
18   rr: logicalTable [ rr: tableName "PARTICIPATES" ];
19   rr: subjectMap [
20     rr: template "http://univ.edu /student/{SID}";
21     rr: class co: Student ;
22   ];
23   rr: predicateObjectMap [
24     rr: predicate co: participatesIn ;
25     rr: objectMap [
26       rr: parentTriplesMap <Course > ;
27       rr: joinCondition [ rr: child "CID"; rr: parent "ID". ];
28     ];
29   ].

```

FIGURE 3.37 : Un exemple du mapping R2RML (Das et al. 2012).

3.3.2.3 Évaluation des langages

Le groupe W3C RDB2RDF a proposé une liste des caractéristiques utilisées pour valider un langage du mapping. Cette liste est composée de 14 (F1-F14) caractéristiques (Tableau 3.7) qui déterminent en fonction du langage du mapping approprié pour une application spécifique.

- Logical Table to Class (F1)
- M N Relationships (F2)
- Project Attributes (F3)
- Select Conditions (F4)
- User-dened Instance URIs (F5)
- Literal to URI (F6)
- Vocabulary Reuse (F7)
- Transformation Functions (F8)
- Datatypes (F9)
- Named Graphs (F10)
- Blank Nodes (F11)
- Integrity Constraints (F12)

F1	Enables the mapping of such a logical table to a class in the RDF vocabulary.
F2	Enables the mapping of link tables to properties in the RDF vocabulary.
F3	Enables projecting a subset of the attributes in the mapping.
F4	Enables the definition of such select conditions in the mapping.
F5	Enables user-defined instance URIs.
F6	Enables the generation of URIs from literal value.
F7	Enables the mapping to existing RDF vocabulary terms.
F8	Enables the conversion of values between the RDB and RDF representations.
F9	Enables the explicit representation of datatype information in the mapping.
F10	Enables the support of named graphs in the RDB-to-RDF mapping.
F11	Enables support for generating blank nodes.
F12	Enables the explicit description of constraints in a mapping language.
F13	Enables the definition of static metadata.
F14	Enables the mapping of one table to n classes.
F15	Enables explicit support for RDF-based write access to relational data.

TABLE 3.7 : Description de chaque caractéristiques (Hert et al. 2011).

- Static Metadata (F13)
- One Table to n Classes (F14)
- Write Support (F15)

Le Tableau 3.8 illustre la comparaison de langages du mapping présenté ci-dessus basés sur ces caractéristiques.

Langages\Carac	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
Direct Mapping	Supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté
eD2R	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Non-supporté	Supporté	Supporté-partiel	Non-supporté	Supporté	Non-supporté
R2O	Supporté	Non-supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Non-supporté	Supporté	Supporté-partiel	Non-supporté	Supporté-partiel	Non-supporté
Relational.OWL	Supporté	Non-supporté	Supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Non-supporté	Supporté	Non-supporté	Supporté	Supporté-partiel	Non-supporté	Non-supporté	Supporté
Virtuoso	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté-partiel	Non-supporté	Supporté	Non-supporté
D2RQ	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Non-supporté	Supporté	Supporté-partiel	Supporté	Supporté	Non-supporté
Triplify	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Non-supporté	Non-supporté	Supporté-partiel	Non-supporté	Supporté	Non-supporté
R2RML	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté	Supporté-partiel	Supporté	Supporté	Non-supporté
R3M	Supporté-partiel	Supporté	Supporté	Supporté-partiel	Supporté	Supporté	Supporté	Supporté	Supporté	Non-supporté	Non-supporté	Supporté	Non-supporté	Supporté	Supporté

Supporté
 Supporté-partiel
 Non-supporté

TABLE 3.8 : Comparaison de langages du mapping avec l'ensemble des caractéristiques proposées par le groupe W3C RDB2RDF (Hert et al. 2011).

3.3.3 La construction des mappings

Les mappings permettent de transformer les requêtes d'utilisateurs qui sont exprimées en termes ontologiques en requêtes formelles qui sont exprimées en un langage de requête formel tel que SQL, XQuery et ainsi de suite. Chaque mapping définit une relation bidirectionnelle entre un concept des ontologies et un schéma de données, permettant de transformer chaque composant de la requête utilisateurs à une sous-requête formelle sur les sources de données.

Dans cette sous-section, nous visons à fournir une approche générale comprenant des recommandations et des étapes nécessaires pour construire des mappings. En général, la construction des mappings n'est pas une tâche facile, elle exige du temps et des efforts de nombreuses personnes, y compris des techniciens et des experts du domaine. La méthodologie de la construction peut également varier en fonction des caractéristiques des ontologies et du SGBD. Dans notre cas, les ontologies sont construites en réutilisant des modèles de données des SGBDs, les mappings seront assez directs. Ils contiennent principalement des relations de 1 à 1 entre un concept ontologique et un schéma des données. Par contre, si les ontologies sont construites indépendamment avec l'existence des SGBDs, les mappings seront plus complexes car ils doivent englober éventuellement des informations supplémentaires telles que les annotations sur les termes ontologiques. Dans le reste de cette section, nous allons présenter les éléments principaux et les briques de base pour la construction des mappings, cependant, nous illustrons à chaque étape, les travaux réalisés reliés au cas d'études du GIN.

3.3.3.1 Choix d'un langage pour exprimer les mappings

Plusieurs langages ont été proposés pour formaliser des mappings (section 3.3.2). Chaque langage a ses propres avantages et inconvénients. Le choix d'un langage approprié dépend notamment de la manière dont les ontologies ont été construites et du besoin d'utilisation du SGBD et des mappings.

Il n'existe pas un langage de mapping en commun, parmi les langages de mapping présentés ci-dessus, **R2O**, **D2RQ** et notamment **R2RML** sont les plus fréquemment utilisés par la communauté. **R2RML** peuvent permettre d'atteindre l'exhaustivité et ainsi être adoptés par les logiciels dans un futur proche (Spanos et al. 2012).

Dans le cas d'études du GIN, pour que les mappings soient facilement intégrés dans le système de requête (présenté dans le chapitre 4), nous utilisons XML pour la représentation des mappings. Cependant, le fichier .xml des mappings est structuré à base de spécifications du **D2R** (Bizer 2003).

3.3.3.2 Définition des mappings

La définition est un processus itératif qui demande de la participation des experts du domaine (*qui ont des compétences métiers*) et des techniciens (*qui sont familier avec les outils informatiques*). Il est nécessaire d'avoir des discussions et des échanges entre les techniciens et les experts pour avoir un ensemble des mappings corrects, complets et utilisables.

Nous proposons une interface de mapping contenant deux zones graphiques : Une pour les concepts ontologiques (*zone 1*) et une pour les schémas des sources de données (*zone 2*). Les chercheurs construisant des mapping choisissent un concept dans *zone 1* et le mappe avec les schémas de correspondances dans la *zone 2*. L'ensemble des mappings sont visualisés intuitivement pour que les utilisateurs puissent facilement consulter, éditer ou supprimer.

Une telle interface peut être construite comme un outil web ou une application informatique. Nous pouvons également utiliser des logiciels de cartographie mentale open-sources. Bien

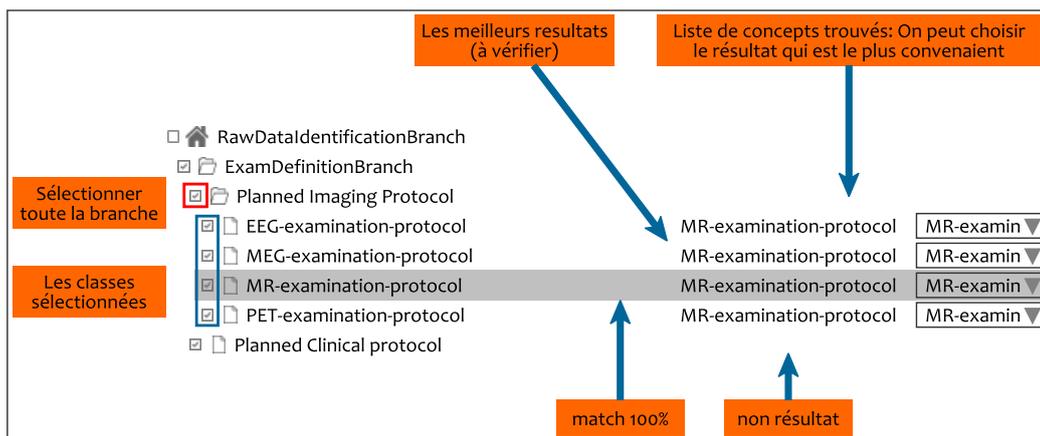


FIGURE 3.38 : Exemple d'un outil web utilisé pour définir manuellement des mappings.

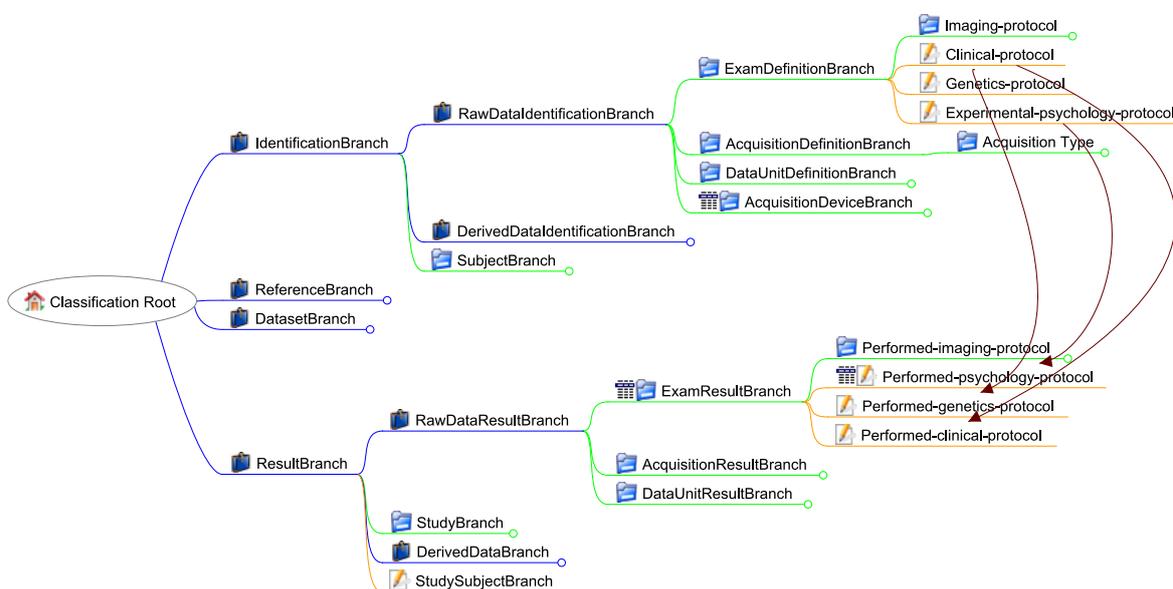


FIGURE 3.39 : La définition des mappings avec Freemind.

que ces logiciels ne soient pas dédiés pour ce type de travaux, il est possible définir facilement les liaisons (*associations*) entre les éléments affichés.

Pour le cas d'études du GIN, nous utilisons le logiciel de cartographie mentale FreeMind pour la définition et la visualisation des mappings. Cependant, ce logiciel devient moins efficace quand les mappings sont nombreux, nous avons développé donc un outil web où les chercheurs peuvent définir manuellement des mappings en utilisant des listes déroulantes correspondantes.

La construction est aussi itérative, c'est-à-dire que la première version des mappings est souvent insuffisante et incomplète, des modifications et ajustements seront alors nécessaires.

3.3.3.3 Génération, teste et maintien des mappings

- **Génération des mappings** : Les mappings définis par les constructeurs sur l'interface de mapping sont exportés en langage du mapping présélectionné et intégrés ensuite

dans le module de transformation des requêtes. Une vérification manuelle des mappings générés sera éventuellement nécessaire.

- **Teste des mappings** : Les mappings sont utilisés pour transformer des requêtes. L'exactitude et l'exhaustivité des mappings devraient être testés avec des requêtes d'utilisateurs : des requêtes simples aux requêtes complexes. Si les requêtes formelles (résultats du processus de transformation) ne donnent pas des résultats correspondants aux besoins des utilisateurs (*incomplets* ou *inexactes*), les mappings doivent être ajustés.
- **Maintien des mappings** : L'utilisation quotidienne des données des utilisateurs génère des changements et des évolutions dans la base de données. Les ontologies du domaine peuvent évoluer par les nouvelles connaissances du domaine. Ces changements demandent des mises à jour des mappings, soit d'ajouter des nouveaux mappings, soit de supprimer ou modifier des mappings existants. La mise à jour peut être réalisée automatiquement ou manuellement. Une fonction d'alerte sera utile. Elle permet aux utilisateurs d'alerter des besoins de la mise à jour des mappings. Le maintien des mappings est indispensable surtout dans le contexte de multi-utilisation de données. Des points de vue multi-utilisateurs différents peuvent porter sur des objets communs présents dans les sources de données, il faut implémenter des moyens permettant d'aligner (mapper) des "ontologies utilisateurs" différentes avec l'ontologie métier du système de requête. Les acteurs métiers peuvent également ajouter ou supprimer un (des) concept(s) ontologique(s) grâce aux classes de données ajoutées ou supprimées.

La généralisation des mappings dépend du langage utilisé pour exprimer des ontologies et l'approche utilisée pour les construire. Ce n'est pas facile à mapper les ontologies avec les schémas de données. Les schémas de données ne fournissent pas de sémantique explicite et formelle pour les données, contrairement aux ontologies. Au contraire, les ontologies sont partagées et réutilisables mais les schémas de données sont utilisés pour une base de données spécifique. Les langages de requête formels sur les données (SQP par exemple) sont souvent plus expressifs que SPARQL sur les ontologies, établir des mappings complets entre les ontologies et les sources de données est impossible. Les mappings obtenus à partir de ce processus de construction sont encore limités. Nous pouvons requêter seulement une partie de la base de données.

3.4 CONCLUSION

Nous avons présenté, dans ce chapitre, une méthodologie pouvant fournir aux acteurs de différentes disciplines un accès sémantique aux données. Nous rappelons que la structuration des données ne permet pas la multi-utilisation des données par des chercheurs de plusieurs domaines et il existe en fait, un gap entre le point de vue informatique (*données*) et le point de vue métier (*acteurs métiers*). Notre approche est donc une couche conceptuelle faisant un lien entre ces deux points de vue.

Depuis longtemps, l'ontologie a été utilisée dans les systèmes d'intelligence artificielle pour la représentation et le raisonnement des connaissances. De nos jours, elle est utilisée comme un composant pivot du Web Sémantique pour rendre les ressources web sémantiquement acces-

sibles par l'humain et la machine en leur fournissant des annotations sémantiques, explicites et formelles. Nous exprimons dans notre contexte l'ontologie sous deux aspects : une représentation semi-informelle visuelle aux acteurs et une description semi-formelle avec OWL en exploitant la puissance de l'inférence pour générer des requêtes.

Les besoins d'information des utilisateurs sont exprimés en termes ontologiques de l'ontologie d'application représentée dans la couche conceptuelle. Nous avons proposé un ensemble de règles de représentation permettant à différents acteurs métiers de formuler leurs requêtes ontologiques de façon graphique et interactive. Des règles de transformation sont utilisées pour transformer le graphe de requête utilisateur en requête SPAQL qui est ensuite transformée en requête formelle sur les données en utilisant des mappings prédéfinis.

Nous avons introduit également des démarches pour la construction des ontologies d'application et des mappings. Ces démarches ne sont pas limitées à un cas d'étude spécifique mais elles sont réutilisables et reproductibles dans de nombreux domaines et d'applications.

Dans le chapitre suivant, nous allons présenter le cas d'application développé dans le cadre du projet BIOMIST. Ce chapitre a pour objectif de tester la faisabilité de la méthodologie proposée en contexte réel et d'ouvrir ensuite des perspectives pour les travaux futurs.

APPLICATION : ACCÈS SÉMANTIQUE AUX DONNÉES DANS LE DOMAINE DU PLM POUR L'IMAGERIE BIOMÉDICALE

SOMMAIRE

4.1	PLM POUR L'IMAGERIE BIOMÉDICALE	133
4.1.1	Adaptation des solutions pour la gestion des données BMI	133
4.1.2	Interrogation de données dans Teamcenter	134
4.2	SYSTÈME DE REQUÊTE VAQUERO	137
4.2.1	Infrastructure	137
4.2.2	Interface de requête	140
4.2.3	Deux ontologies dédiées aux utilisateurs	144
4.3	EXEMPLES	145
4.3.1	Exemple 01	145
4.3.2	Exemple 02	147
4.4	DISCUSSION	148

Ce chapitre s'intéresse à la validation expérimentale de l'approche proposée. Un système de requête nommé VAQUERO (VisuaAlization and QUERy based Ontology) basé sur l'ontologie a été construit suivant la méthodologie présentée dans le chapitre 3. Nous présentons ici des exemples pour montrer la faisabilité du système. Nous rappelons tout d'abord le domaine d'application : Le PLM pour l'Imagerie Biomédicale.

4.1 PLM POUR L'IMAGERIE BIOMÉDICALE

4.1.1 Adaptation des solutions pour la gestion des données BMI

Comme nous avons présenté ci-dessus, l'approche PLM est proposée initialement dans les entreprises industrielles pour faire face à l'augmentation de la complexité, de l'hétérogénéité et du volume de données en ingénierie collaborative. Les solutions PLM permettent notamment de suivre des modifications sur une donnée, de contrôler des accès des utilisateurs, de concevoir et d'appliquer des workflows d'action, etc. (Allanic 2015). La Neuroimagerie est un domaine de recherche qui partage les mêmes problématiques de gestion de données (*structurer, analyser, accéder, partager et réutiliser*) que les entreprises industrielles il y a 20 ans. Naturellement, les solutions PLM peuvent être utilisées pour surmonter ces problématiques.

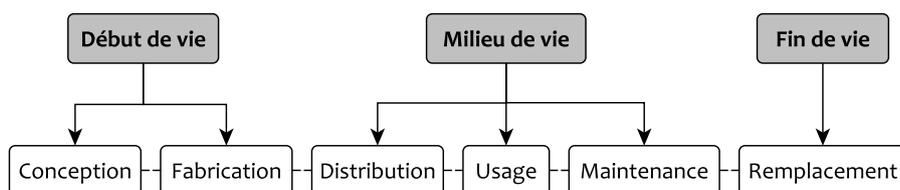


FIGURE 4.1 : Les phases du cycle de vie du produit (Terzi et al. 2010).

Pour Allanic (Allanic et al. 2013), une étude de recherche en Neuroimagerie peut être considérée comme un processus composé de quatre étapes : **spécification de l'étude**, **acquisition de données** (*données brutes*), **traitement de données** (*données dérivées*), et **publication de données** (*données publiées*). Ce processus pourrait également être modélisé comme un cycle ; cela signifie que les données publiées peuvent constituer une base pour les études suivantes (Figure 4.2).

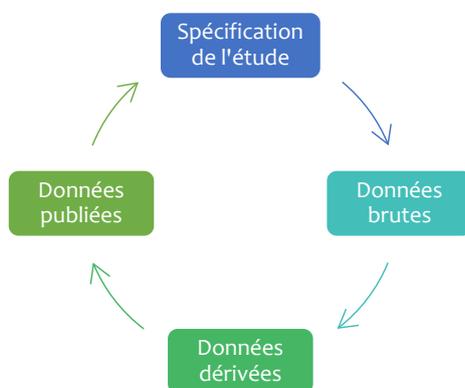


FIGURE 4.2 : Le cycle de vie d'une étude de recherche en Neuroimagerie (Allanic et al. 2013).

De ce point de vue, les solutions PLM peuvent être utilisées pour manipuler une étude de

recherche complexe en Neuroimagerie (Allanic 2015). Le PLM peut permettre d'atteindre les objectifs suivants :

- Les données sont gérées intégralement à partir des spécifications de l'étude.
- Les données sont partagées entre tous les chercheurs.
- L'accès aux données est contrôlé et les modifications des données sont capturées.
- La récupération et la réutilisation des données sont activées.

Pour réaliser ces objectifs, un modèle de données adapté appelé Biomedical Imaging Lifecycle Management (BMI-LM) a été développé pour gérer les données BMI (Allanic et al. 2013). Le BMI-LM a été implémenté et testé au laboratoire GIN (*Imaging Group Neurofunctional*, CEA CNRS Université de Bordeaux), dans le système de gestion de données PLM Teamcenter⁷². Cette implémentation est déployée pour permettre la gestion des données du laboratoire en contexte d'acquisition, de traitement, de validation, d'analyse et de publication de données. Pour Allanic (Allanic 2015), le système PLM est interfacé avec :

- La grille de calcul du laboratoire pour lancer les traitements sur les données,
- Un client pour visualiser le contenu de la base de données sous forme de graphes,
- Le logiciel JMP⁷³, un outil de calcul statistique couramment utilisé en Neuroimagerie.

4.1.2 Interrogation de données dans Teamcenter

Dans tous les systèmes PLM, les produits "gérés" sont souvent complexes et sont le résultat de l'interaction de différents services (*conception, planification des processus, fabrication, etc.*). Les participants (*partenaires, entrepreneurs, etc.*) viennent souvent de différentes entreprises ou services de différents lieux. Les données et informations relatives au produit doivent être stockées dans un référentiel commun (*base de données*) où sa maintenance et son accessibilité doivent être assurées. Il est nécessaire de pouvoir localiser rapidement les données du produit potentiellement pertinentes pour le nouveau projet et d'évaluer leur pertinence (Maranzana 2013). Dans notre cas d'application, les données BMI sont stockées dans la base selon le modèle de données relationnel BMI-LM, elles sont donc interrogeables avec le langage de requête structuré SQL (*Structured Query Language*).

Le système PLM Teamcenter considère l'ensemble de données qu'il stocke comme une base d'objets ou d'items. Un objet est un concept générique auquel est associé les données (Allanic 2015). Les résultats d'une requête sont un ensemble d'objets qui répondent aux besoins d'information de l'utilisateur. Pour les utilisateurs standards, Teamcenter fournit deux fonctions de recherche : La recherche simple et la recherche avancée.

- **Recherche simple** : Ce type de recherche permet de localiser des objets par les mots-clés (recherche textuelle). Les champs disponibles sont : *Nom, ID*, etc. (Figure 4.3).
- **Recherche avancée** : Ce type de recherche permet de combiner plusieurs critères sur l'objet à requêter. Une liste de requêtes les plus fréquemment utilisées sont pré-intégrées (Figure 4.4). Pour des besoins d'information spécifiques, les utilisateurs peuvent définir des nouvelles requêtes grâce à un outil dédié nommé "**Query Builder**".

72. https://www.plm.automation.siemens.com/en_us/products/teamcenter/

73. http://www.jmp.com/en_us/home.html

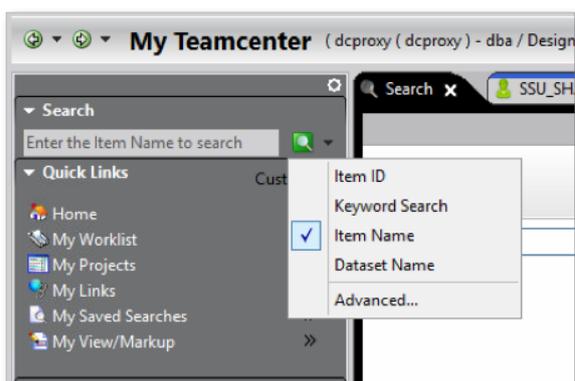


FIGURE 4.3 : Recherche des items par mots-clés

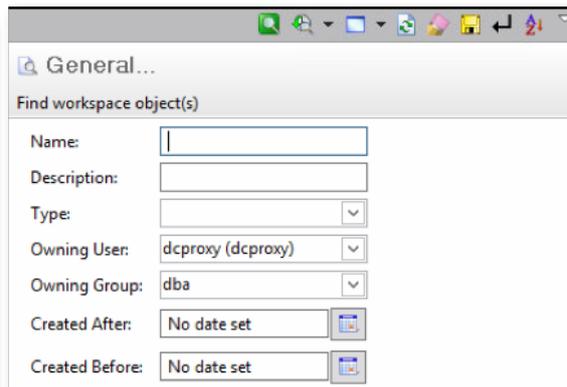
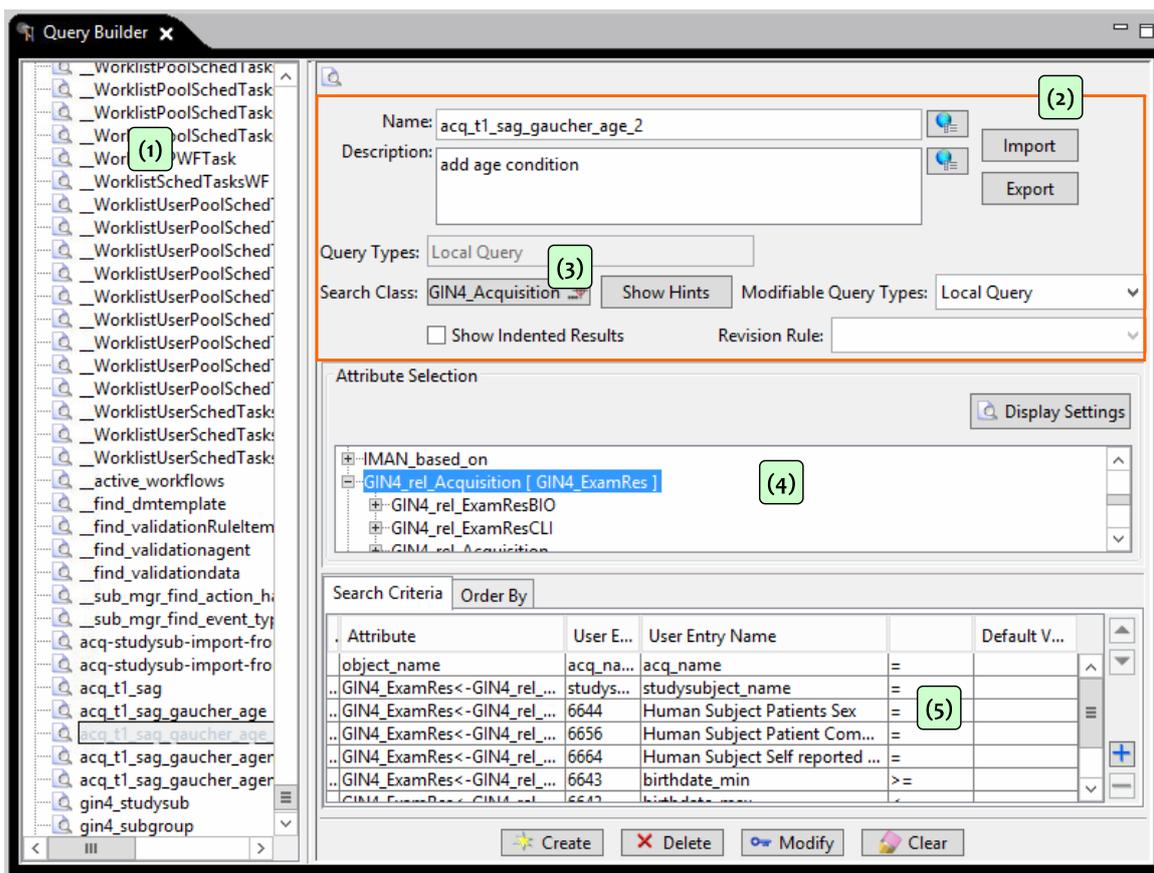


FIGURE 4.4 : Prédéfinie requête avancé.

4.1.2.1 Query Builder

L’outil de recherche Query Builder dans Teamcenter permet aux utilisateurs de construire des requêtes ad-hocs complexes combinant plusieurs critères de recherche sur non seulement un objet requêté, mais aussi sur les autres objets qui lui sont associés.



- (1) Liste des requêtes enregistrées
- (2) Description de la requête enregistrée sélectionnée ou de la nouvelle requête à créer
- (3) Sélection de la classe requêtée
- (4) Affichage de toutes les attributs de la classe sélectionnée et de ses classes inhérites
- (5) Définition des critères de la requête.

FIGURE 4.5 : Query Builder : L’outil de construction de requêtes dans Teamcenter (délivrée de (Siemens 2013)).

4.1.2.2 Limites de Query Builder

- Query Builder est dédié aux utilisateurs avancés. Pour pouvoir construire des requêtes dans Query Builder, il faut connaître de schéma Teamcenter POM (*Gestion des Objets Persistants* ou *Persistent Object Manager*), qui est un arrangement hiérarchique de classes, de sous-classes et d'attributs. Une classe est la définition d'un objet implémenté dans le modèle de données Teamcenter. Autrement dit, la construction d'une requête demande une compréhension assez solide du modèle de données implémenté et la structuration de données de la base. Par exemple, le *zone (4)* (Figure 4.5) affiche tous les attributs des classes associées à la classe requêtée *zone (3)*. Comme illustré dans la Figure 4.6, avec l'objet requêté "GIN4_Acquisition", pour choisir deux objets hérités "GIN4_ExamRes" et "GIN4_StudySub", il faut bien sélectionner les GRM-Rules (*Generic Relationship Management Rule*) correspondantes "GIN4_rel_Acquisition" et "GIN4_rel_ExamResIMA".

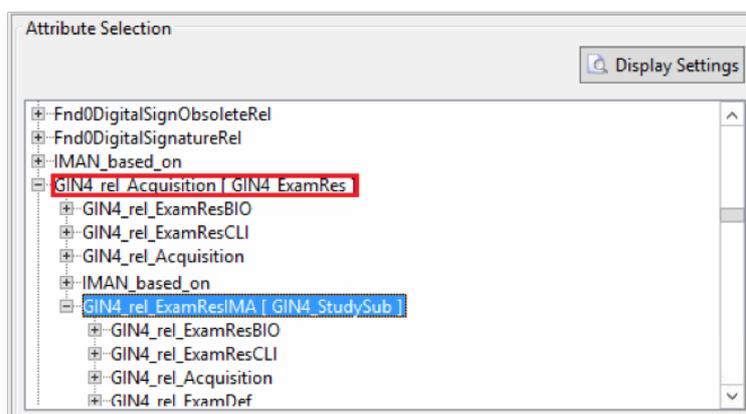


FIGURE 4.6 : Sélection des attributs de la requête.

- Grâce aux requêtes définies avec *Query Builder*, nous pouvons obtenir que les informations associées à l'objet requêté, mais pas à l'ensemble des objets reliés dont les attributs sont définis dans la requête.

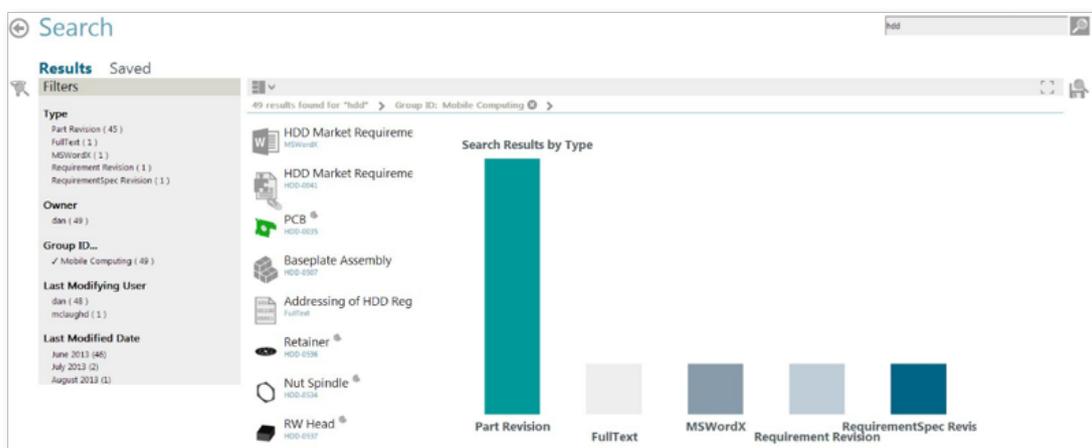


FIGURE 4.7 : Espace de travail actif Teamcenter. Les méthodes statistiques sont utilisées pour représenter les résultats de manière plus significative.

À côté de la version traditionnelle, une version visible, intuitive et active basée sur le Web a été développée pour étendre le système PLM aux divers utilisateurs finaux (Figure 4.7). La recherche d'information implémentée dans cette version est aussi basée sur les mots-clés, mais avec les mécanismes d'interaction et de navigation, elle permet aux utilisateurs de raffiner des requêtes progressivement. Il peut aussi combiner les métadonnées et les mots-clés pour restreindre la recherche.

Dans le reste de cette section, nous décrivons l'implémentation et le démonstrateur de l'approche proposée dans le chapitre 3. Cette implémentation a pour l'objectif de fournir à l'ensemble des utilisateurs finaux multi-métiers un accès sémantique aux données BMI gérées par le système Teamcenter. Les caractéristiques du système Teamcenter sont à considérer pour notre implémentation :

- Les données sont stockées dans le système PLM Teamcenter selon le modèle de données BMI-LM (Allanic et al. 2013).
- PLMXQuery (Sriti et Boutinaud 2012) est implémenté dans le système PLM Teamcenter, le contenu de la BDD PLM Teamcenter est vu comme une base XML (Figure 4.8). L'idée derrière cette approche est de profiter des technologies liées à XML, notamment XPath et XQuery pour interroger la base de données PLM d'une manière plus flexible que le système PLM traditionnel. Dans notre approche, l'ontologie est utilisée pour fournir aux utilisateurs finaux un accès sémantique au contenu XML et les requêtes des utilisateurs formulées à l'aide de concepts ontologiques sont transformées en des requêtes XQuery.

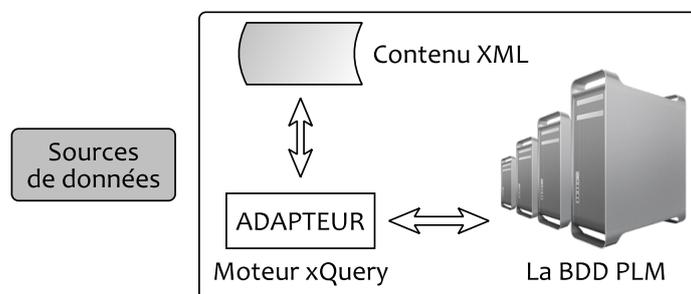


FIGURE 4.8 : Le contenu PLM Teamcenter sont vus comme une base de documents XML (Sriti et Boutinaud 2012).

4.2 SYSTÈME DE REQUÊTE VAQUERO

4.2.1 Infrastructure

Même si la communauté de Neuroimagerie partage de nombreuses normes, chaque groupe de recherche - pour ne pas dire chaque chercheur - utilise son propre vocabulaire pour étiqueter ses données. En outre, la Neuroimagerie est un domaine multidisciplinaire et chaque discipline a ses propres concepts et méthodes d'utilisation des données. Dans ce contexte, Il est difficile d'interroger une base de données inconnue ou occasionnelle, car les utilisateurs ne connaissent pas ni la manière dont les données sont structurées, ni leur sémantique (Pham et al. 2016a;b).

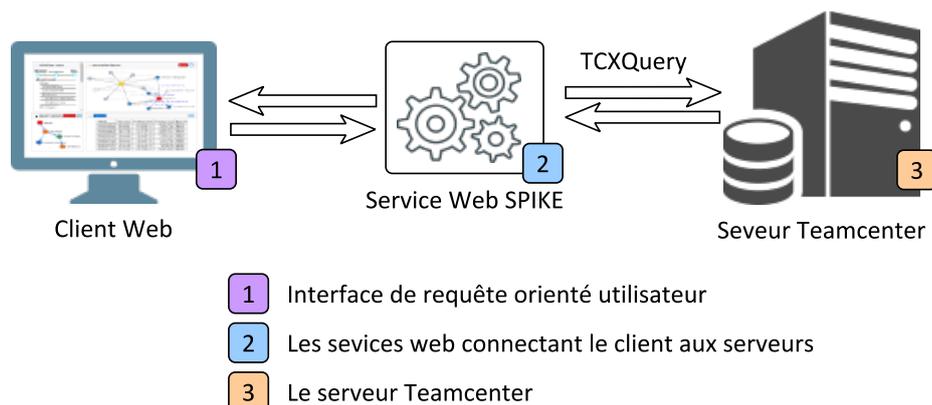


FIGURE 4.9 : L'infrastructure du système de requête VAQUERO.

La Figure 4.9 illustre l'infrastructure du système VAQUERO comprenant trois composants principaux :

- i) Le **client web** est une interface de requête visuelle et interactive sur laquelle les utilisateurs formulent leurs requêtes de données.
- ii) Les **services web SPIKE** permettant de connecter le client et le serveur PLM Teamcenter. Les requêtes xQuery client ainsi que les résultats renvoyés par le serveur Teamcenter sont transmises bidirectionnellement au travers de ces services web.
- iii) Le **serveur PLM Teamcenter** : Les données BMI sont restées dans la BDD Teamcenter, et peuvent être requêtées par le langage de requête xQuery grâce à xQuery Engine (Figure 4.8).

Pour l'aspect "**Orienté utilisateur**", le système VAQUERO, vise à faciliter l'accès aux données de trois types d'utilisateurs correspondant à trois niveaux d'abstraction. L'Interface de requête fournit donc trois mises en page différentes :

- i) **La mise en page du MDD** : Au niveau le plus bas d'abstraction, cette mise en page est dédiée aux utilisateurs *avancés/ techniques* qui ont une compréhension assez solide sur la structuration de données et qui sont familiers avec les termes du modèle de données BMI-LM. Les requêtes peuvent être construites directement en termes d'objets métiers du modèle de données.
- ii) **La mise en page de la classification** : La classification est une extension du MDD qui donne une hiérarchie complète des classes de données. Chaque classe contient un ensemble de données ayant des attributs communs. Une classe est nommée par un nom significatif reflétant ces données. L'implémentation de la classification dans l'interface de requête permet la recherche de facette. Les utilisateurs réguliers peuvent facilement trouver leurs données intéressantes à partir d'une ou plusieurs classes. L'utilisateur peut également naviguer au travers de la classification pour trouver les données qu'ils les intéressent. Les relations *verticales/ horizontales* peuvent être définies entre des classes de la classification, cela permet aux utilisateurs de formuler des requêtes ad-hoc plus complexes sur différentes classes de données. Cette mise en page est dédiée aux utilisateurs formés ou réguliers qui fréquemment manipulent la base PLM.
- iii) **La mise en page des ontologies** : La Figure 4.10 illustre une vue globale de l'infra-

structure du système ou la méthodologie proposée a été implémentée. Les ontologies et ses visualisations graphiques fournissent aux utilisateurs une vue sémantique des données. Les requêtes utilisateur sont exprimées en termes ontologiques et transformées en requêtes xQuery grâce à l'ensemble des mappings prédéfinis. Cette mise en page est dédiée aux utilisateurs finaux non-techniciens ou les utilisateurs occasionnels qui ne connaissent pas le MDD. Différents groupes d'utilisateurs peuvent utiliser différentes ontologies qui sont une instance d'une ontologie générale.

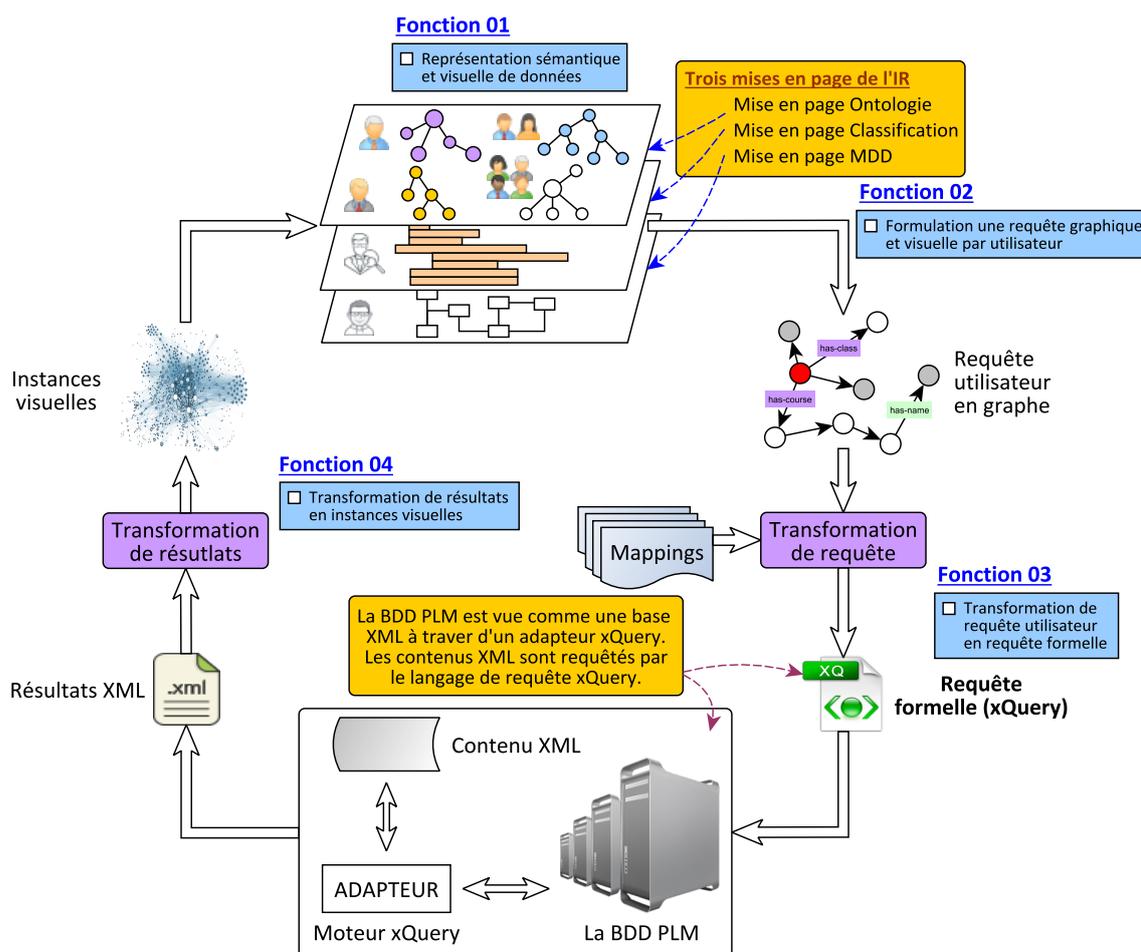


FIGURE 4.10 : Accès sémantique aux données à base d'ontologie, adapté au domaine du PLM pour Imagerie Biomédicale. Les données BMI gérées par les solutions PLM sont accessibles comme une base XML à travers d'un adaptateur xQuery (Sriti et Boutinaud 2012). Les utilisateurs formulent leurs requêtes en termes ontologiques sur une interface visuelle et interactive, ces requêtes sont ensuite transformées en requêtes xQuery grâce à un module de transformation et l'ensemble des mappings prédéfinis. Les résultats des requêtes xQuery sont retransformés en instances visuelles dans l'interface de requête.

Pour l'aspect "**visuel**", comme présenté ci-dessus, jouant le rôle d'un support cognitif externe pour comprendre la complexité (Keller et Tergan 2005), les visualisations graphiques sont utilisées à trois niveaux pour faciliter le processus de consultation des utilisateurs. Tous les objets métier du modèle de données, les classes de classification des données ou les concepts ontologiques sont représentés dans un arbre de navigation alors que toutes les relations éventuelles (*objets métier vs objets métier*, etc.) sont représentées dans une zone graphique, intuitif et interactif pour aider les utilisateurs à définir rapidement et facilement leurs requêtes.

Par exemple, au niveau le plus élevé, un utilisateur commence par naviguer dans l'arbre conceptuel pour sélectionner un concept intéressant. Lorsqu'un concept est sélectionné, le graphique de relations met en évidence toutes ses relations avec les autres concepts, l'utilisateur peut sélectionner un de ces concepts et l'ajouter dans la requête afin de faire une condition de requête. Ce processus est répété jusqu'à ce que la requête soit définie complètement. Au cours du processus de fabrication, la requête formulée par l'utilisateur est représentée graphiquement pour fournir une visualisation illustrée de tous les concepts sélectionnés et des conditions de requête définies. À la fin, cette requête est traduite dans celle qui est exécutable par un processeur de requêtes. Les résultats de la requête sont affichés sur la même interface, sous la forme d'un graphe (*noeuds* pour les *objets* résultants, *arêtes* pour les *relations*).

4.2.2 Interface de requête

L'interface de requête est construite à base du Web pour qu'elle soit visuelle, intuitive et interactive. Des outils supportés déjà disponibles et open sources sont utilisés directement pour fournir aux utilisateurs une meilleure visualisation de l'ontologie, de la classification ou du modèle de données, comme *D3.js* par exemple. La Figure 4.11 illustre une vue globale de l'interface de requête comprenant cinq composants :

- i) **L'arbre de concepts** : Cette zone illustre la hiérarchie de concepts. Un concept peut être un concept ontologique, une classe de données ou un objet métier du MDD. Pour les concepts ontologiques et les classes de la Classification, nous utilisons la liste indentée pour mettre en évidence les relations héritées (parent-enfant, généralisation/ spécification) entre concepts. Un curseur est utilisé pour naviguer entre différentes mises en page de l'interface.
- ii) La **description du concept sélectionné courant** : Cette zone a pour l'objectif de décrire le concept sélectionné courant. Une description contient des annotations et des restrictions sur le concept.
- iii) Le **graphe des relations** : Cette zone illustre visuellement toutes les relations existantes entre l'ensemble des concepts. Les concepts sont représentés par des noeuds, une arrête relie deux noeuds représentant une relation entre deux concepts.
- iv) Le **graphe de requête** : La requête définie par l'utilisateur est représentée en temps réel sous la forme d'un graphe. Des mécanismes d'interaction sont implémentés entre le graphe de requête, le graphe des relations et l'arbre des concepts. Grâce à cela, l'utilisateur sélectionne un noeud dans le graphe de requête, tous les noeuds et relations qui le relient sont mis en évidence dans le graphe de relations. Cela permet à l'utilisateur d'ajouter facilement des nouvelles conditions à la requête.
- v) La **définition des paramètres (conditions) de la requête** : Cette zone représente tous les attributs du concept sélectionné courant dans le graphe de requête. Pour définir une condition de requête, l'utilisateur choisit simplement l'attribut sur lequel la condition est posée, une opération et une valeur appropriée.

Comme présenté ci-dessus, l'interface de requête fournit aux utilisateurs trois différentes mises en pages correspondantes aux trois différents groupes d'utilisateurs qui sont présentées brièvement ci-dessous.

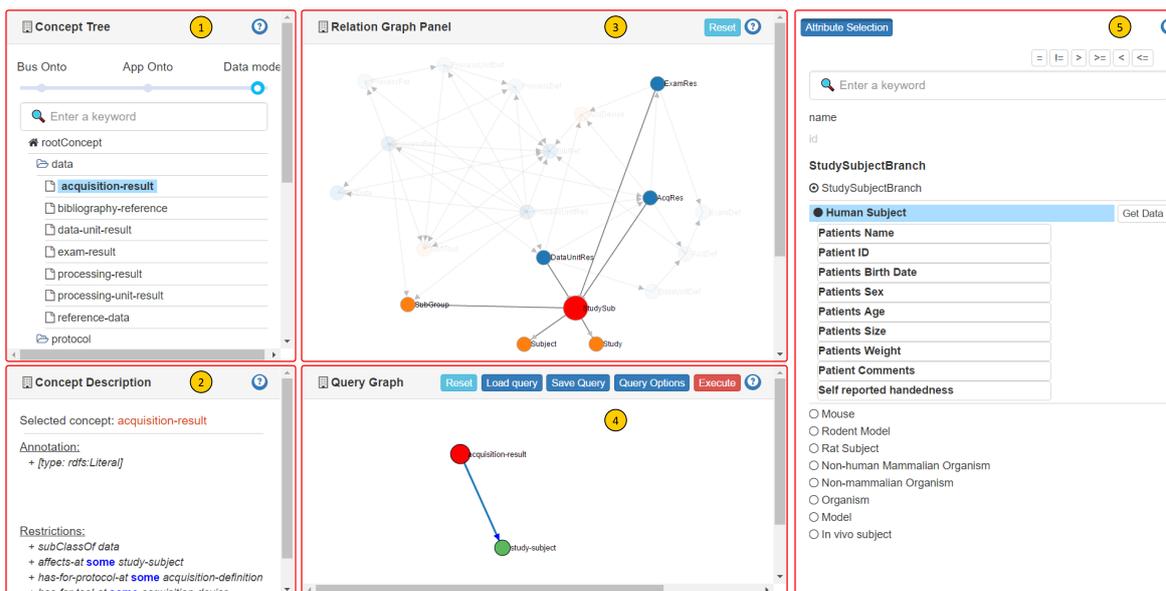


FIGURE 4.11 : Interface de requête implémentée dans VAQUERO.

4.2.2.1 Mise en page "MDD"

Dédiée aux utilisateurs *techniques/ avancés*, cette mise page représente le MDD BMI-LM sous la forme d'un arbre d'objets métiers et d'un graphe de relations entre eux. Les objets métiers sont classifiés en quatre catégories : *Donnée, protocole, source* et *outil*.

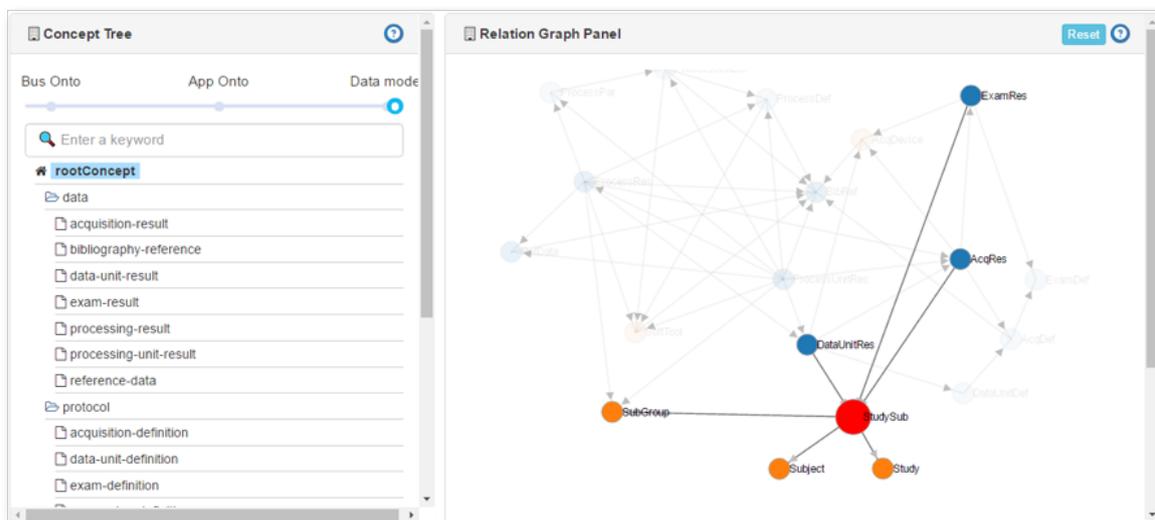


FIGURE 4.12 : La mise en page "MDD" de l'interface de requête.

4.2.2.2 Mise en page "Classification"

Les données BMI sont stockées dans Teamcenter suivant le modèle de données BMI-LM. Le système PLM Teamcenter fournit également un outil de classification pour classer les données en catégories prédéfinies. Pour le laboratoire GIN, la Classification contient environ 700 classes et 3000 attributs qui leur sont associés.

La classification joue un rôle important dans la gestion des données BMI. Les attributs de

classification sont utilisés pour associer toutes les données spécifiques aux classes. Chaque entité (*objet métier*) du MDD BMI-LM est liée à une classe concrète ou abstraite. Afin de permettre l'intégration de données entre différents systèmes d'information en différents endroits, la classification dans Teamcenter a été créée en considérant la notation d'ontologie, ce qui signifie qu'il y a une signification sémantique derrière chaque nom de classe. La Figure 4.13 illustre une vue globale de la Classification implémentée dans Teamcenter.



FIGURE 4.13 : La classification implémentée dans Teamcenter.

La mise en page "**Classification**" de l'interface de requête (Figure 4.14) représente la classification sous la forme d'une liste indentée. L'utilisateur navigue (*développer* ou *réduire*) à travers la classification pour trouver la classe de données qu'il souhaite. Des conditions peuvent être posées sur les attributs de la classe sélectionnée pour filtrer les données classées à cette classe. Des relations entre différentes classes peuvent être définies, grâce auxquelles, les utilisateurs peuvent formuler des requêtes plus complexes sur différents objets métiers. Dans ce cas-là, la requête utilisateur doit être transformée en requête xQuery et traitée pour l'adapter à xQuery (Sriti et Boutinaud 2012) parce que le système Teamcenter ne supporte pas une recherche sur des classes multiples de la Classification.

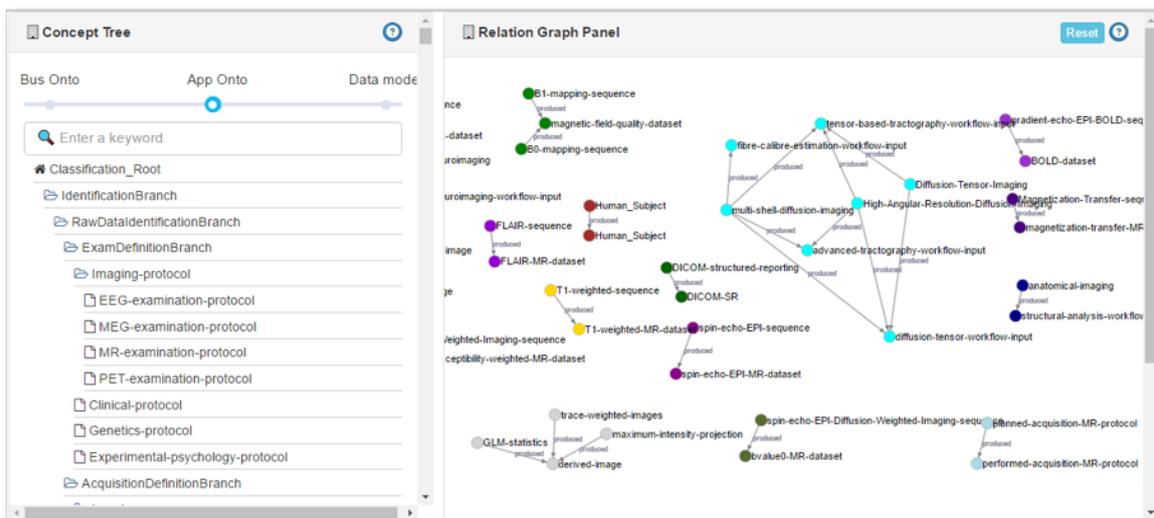


FIGURE 4.14 : La mise en page "Classification" de l'interface de requête.

4.2.2.3 Mise en page "Ontologie"

Cette mise en page est dédiée aux utilisateurs métiers non-techniciens/ occasionnels ou inexpérimentés. En raison du manque de compréhensions nécessaires, les termes ontologiques utilisés choisis sont : *significatifs*, *clairs* et *propres* aux compétences des utilisateurs. Différentes ontologies (qui sont les spécifications d'une ontologie générale) peuvent être utilisées pour différents utilisateurs métiers. Dans le cadre de notre application, nous avons proposé deux ontologies qui sont deux différents groupes de chercheurs : Les chercheurs qui travaillent avec les données "Imageries" et les chercheurs qui travaillent avec les données "Psychologies". La Figure 4.15 illustre la mise en page "Ontologie" de l'Interface de requête. Comme deux mises en page ci-dessus, elle contient un arbre de concepts ontologiques et un graphe de relations entre ces concepts.

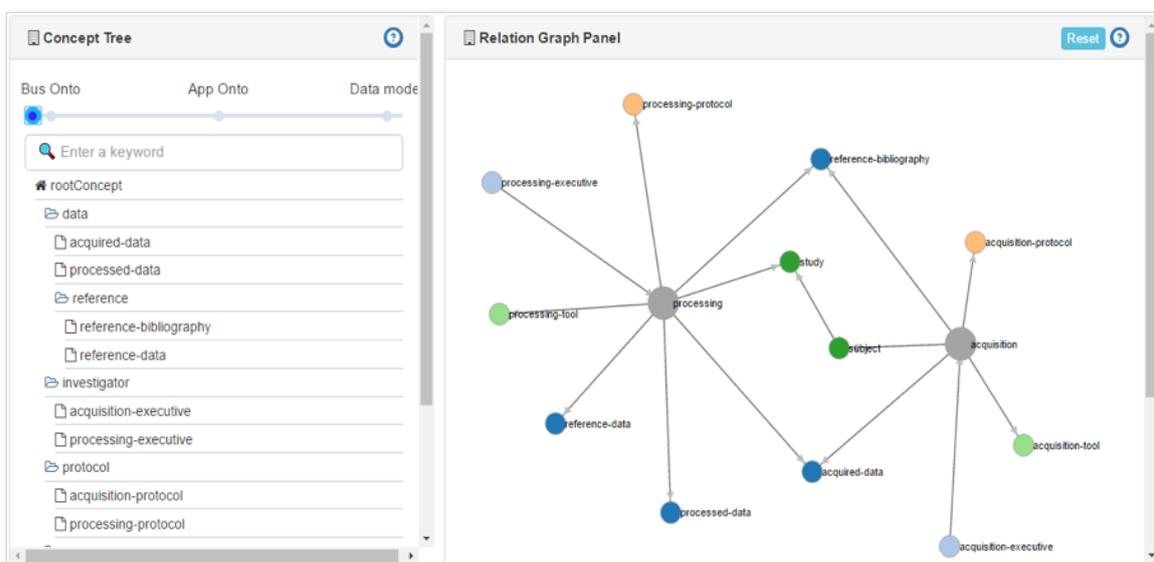


FIGURE 4.15 : La mise en page "Ontologie" de l'interface de requête.

4.2.3 Deux ontologies dédiées aux utilisateurs

Les chercheurs du GIN ont manipulé deux principaux types de données : les données d'imagerie obtenues à l'aide des techniques d'imagerie médicale (*PET, EEG, MEG, MRI*) et les données *comportementales / psychologiques* comme résultats du test de *comportement / psychologie*.

Les termes employés dans chaque ontologie proposée doivent aider chaque "profil" de scientifique à reconnaître les données même si elles sont nommées d'une manière spécifique dans la base de données et les relations définies entre les concepts doivent représenter les relations sémantiques entre les données auxquelles les concepts sont connectés. Suivant le résultat de l'entretien, nous avons décidé de proposer deux ontologies différentes, une pour les chercheurs qui travaillent avec les données d'imagerie et une pour les chercheurs qui travaillent avec les données de psychologie.

La construction des ontologies d'applications utilisées dans le système de requête VAQUERO a été présentée comme cas d'études dans la section 3.2. À partir de l'ontologie générale, nous avons spécifié l'arbre conceptuel présenté dans la Figure 3.24 dans un arbre de concepts plus détaillés : un pour les chercheurs travaillant avec des données d'imagerie (Figure 4.16) et un pour les chercheurs travaillant avec des données de psychologie (Figure 4.17). Tous les vocabulaires utilisés dans ces ontologies ont été définis en discutant avec les scientifiques du GIN pour s'assurer qu'ils sont adéquats et proches des utilisateurs correspondants. Les relations entre les sous-concepts sont héritées des concepts-parents qui sont illustrés ci-dessus (Figure 3.25).

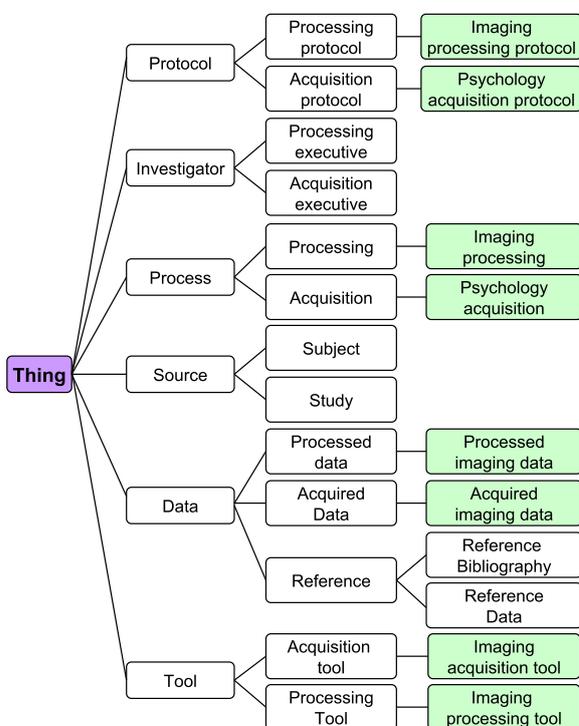


FIGURE 4.16 : L'ontologie proposée pour les chercheurs qui travaillent avec les données d'imagerie.

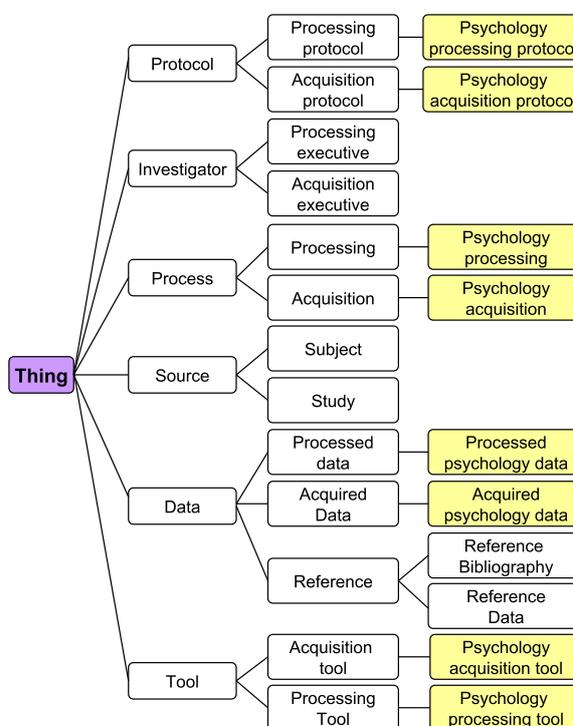


FIGURE 4.17 : L'ontologie proposée pour les chercheurs qui travaillent avec les données de psychologie.

4.3 EXEMPLES

Cette section fournit des exemples réels fréquemment utilisés par les chercheurs GIN pour montrer l'usage et la faisabilité du système VAQUERO.

Pour pouvoir formuler une requête, l'utilisateur commence par naviguer dans la hiérarchie des concepts pour sélectionner le concept associé aux données qu'il souhaite interroger. Dans un même temps, le graphe de relations met en évidence ce concept et toutes les relations qu'il a avec les autres concepts. L'utilisateur peut alors sélectionner un de ces concepts et l'ajouter dans la requête. Ce processus est répété jusqu'à ce que la requête de l'utilisateur soit formulée complètement. Au cours du processus de formulation de la requête, cette dernière est représentée sous la forme d'un graphe afin de fournir une visualisation illustrée de tous les concepts sélectionnés et des conditions définies. La requête de l'utilisateur (*originale* ou *enrichie*) doit être transformée en requête xQuery grâce aux mappings prédéfinis. Ces mappings sont définis après des discussions et échanges avec les chercheurs et les techniciens au GIN. Pour la première version, les mappings sont représentés par le langage XML.

Nous prenons des exemples concrets et démontrons comment elles sont formulées à l'aide de l'interface de requête. Cependant, nous les montrons sur la mise en page "*Ontologie*" où les ontologies sont implémentées pour fournir aux utilisateurs finaux un accès sémantique aux données.

4.3.1 Exemple 01

Un exemple de requête pourrait être utilisée par les chercheurs du GIN :

Chercher toutes les données d'imagerie obtenues en suivant le protocole d'acquisition "Echo-Plannar-Imaging-Sequence" sur tous les sujets "Homme" et "Gaucher"

Les utilisateurs ne connaissent pas l'organisation de données, pour cette requête ils peuvent rencontrer deux difficultés :

- i) Ils ne savent pas comment les données relatives à "*sujet*", "*protocole d'acquisition*" ou "*résultat d'acquisition*" sont appelées techniquement.
- ii) Ils ne savent pas que les données sont enregistrées dans des tables relationnelles et quelles sont les relations entre ces tables. Il est difficile pour les acteurs métiers d'identifier les éléments à mettre et les bons chemins à parcourir entre eux quand ils formulent une requête formelle (*xQuery*).

La mise en page "*Ontologie*" nous permet de surmonter ces problèmes. Pour la condition *i*, les concepts ontologiques implémentés dans l'interface de requête sont nommés par des termes/ des vocabulaires compréhensibles et propres aux acteurs métiers. Ces derniers définissent leurs requêtes en termes ontologiques au lieu des termes techniques. Pour la condition *ii*, les avantages de la visualisation et des mécanismes d'interactions ainsi que des moteurs d'inférence permettent aux acteurs métiers de pouvoir sélectionner des bons chemins à parcourir entre les concepts ontologiques.

Dans notre exemple, les "données d'imagerie" sont les données désirées et elles sont associées au concept "acquired-imaging-data" tandis que le "protocole d'acquisition" "Echo-Planar-Imaging-Sequence" est connecté au concept "image acquisition tool". Ce protocole est effectué sur le "sujet" qui appartient au concept du même nom "subject". Trois conditions de la requête :

- *Imaging-acquisition-protocol.name* = "Echo-Planar-Imaging-Sequence".
- *Subject.sex* = "H". (H pour Homme et F pour Femme)
- *Subject.handedness* = "G" (G pour Gaucher et F pour Femme).

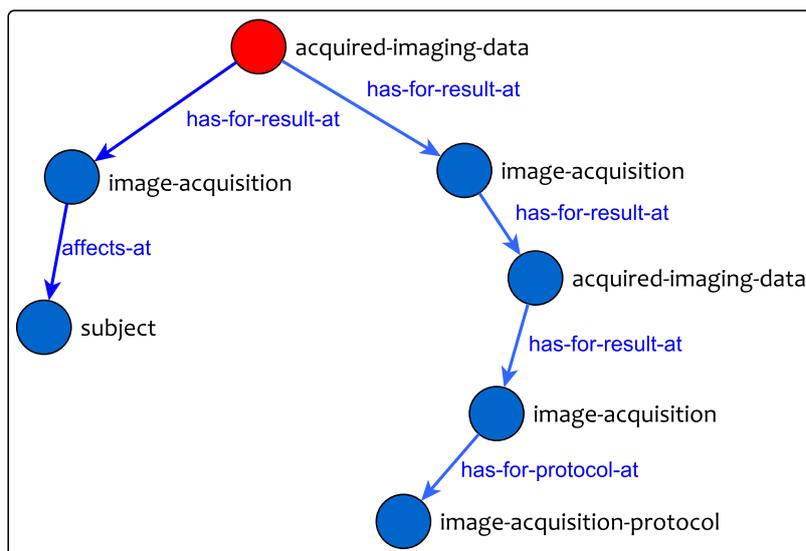


FIGURE 4.18 : Représentation graphique de la requête utilisateur définie par les chercheurs GIN.

La mise en page "Ontologie" (avec l'ontologie illustrée dans la Figure 4.16 et Figure 4.17) aide les acteurs métiers à formuler une requête sous le forme d'un graphe illustré à la Figure 4.18. Cette requête initiale est transformée en requête xQuery à la Figure 4.19.

```

1 declare namespace Teamcenter="java:com.cadesis.spike.plm.teamcenter10.xquery.TCExtension";
2 declare namespace local="local";
3
4 declare variable $teamcenter := Teamcenter:instance();
5 declare $quote := "&#34;";
6
7 <elements>
8   {
9     let $query :=
10    <query query_name="GIN4_ExamRes-GIN4_StudySub_GIN4_AcqDef_20160725031642">
11      <param name="subject1.HumanSubject.PatientsSex" value="M"/>
12      <param name="subject1.HumanSubject.PatientComments" value="G"/>
13      <param name="imageacquisitionprotocol13.EchoPlanarImagingSequence.Comments" value="*" />
14    </query>
15    let $foundObjects := Teamcenter:Query($teamcenter, $query)
16    for $i in (1 to fn:count($foundObjects))
17      let $foundObject :=
18        <object KTCId="{ $foundObjects[$i]/@KTCId}" objectId="{ $foundObjects[$i]/@item_id}" >
19        </object>
20      return $foundObject
21    }
22  </elements>

```

FIGURE 4.19 : La requête utilisateur est transformée à la requête xQuery.

Les résultats de la requête sont retournés et représentés sous le forme d'un tableau ou un graphe dans la même interface de requête. Les acteurs métiers peuvent consulter cette visualisation de résultats pour comprendre les relations entre les données obtenues et, de plus,

pour trouver des informations cachées et des connaissances inférées. La Figure 4.20 illustre un extrait des résultats obtenus qui sont représentés sous la forme d'un graphe.



FIGURE 4.20 : Représentation graphique des résultats. Cliquer sur un noeud pour mettre en évidence tous les noeuds reliés.

4.3.2 Exemple 02

Utilisateur ne connaît pas son besoin avant de formuler une requête. Une requête utilisateur peut donc être formulée au fur et à mesure, d'une requête simple à une requête plus complexe contenant un nombre de conditions. Comme par exemple, la requête 02 :

Chercher tous les données d'acquisitions "T1-sag" des "sujets gauchers" dont "la date de naissances et entre 18 et 20 ans"

peut être réalisé en trois étapes :

- i) Chercher toutes les acquisitions du type **T1-sag**
- ii) Rajouter les sujets qui sont des **gauchers**
- iii) Rajouter les sujets dont la date de naissances est entre **18 et 20 ans**

Pour l'étape **i. "T1-sag"**, il s'agit une acquisition d'image, pour chercher tous les acquisitions de ce type, il faut sélectionner le concept "*acquisition-imaging-data*" dans l'arbre de concept, l'ajouter à la requête et définir la condition : *acquisition-imaging-data.name = "T1-sag"*.

Ensuite, à l'étape **ii.** en cliquant sur le concept "*acquisition-imaging-data*", nous sélectionnons le concept "*subject*" au travers du concept intermédiaire "*image-acquisition*", l'ajoutons à la requête actuelle et définissons la condition : *subject.handedness* = "G" (G pour *Gaucher* et *D* pour *Droitier*).

Enfin, à l'étape **iii.** nous ajoutons la deuxième condition sur le concept "*subject*" afin de compléter la requête initiale : *subject.Patients-Birth-Date* >= "18Y" et *subject.Patients-Birth-Date* <= "20Y".

Le processus de formulation de la requête est illustré à la Figure 4.21. Les résultats de la requête sont représentés sous la forme d'un graphe ou d'un fichier .csv. Le Tableau 4.1 illustre un extrait .csv de résultats.

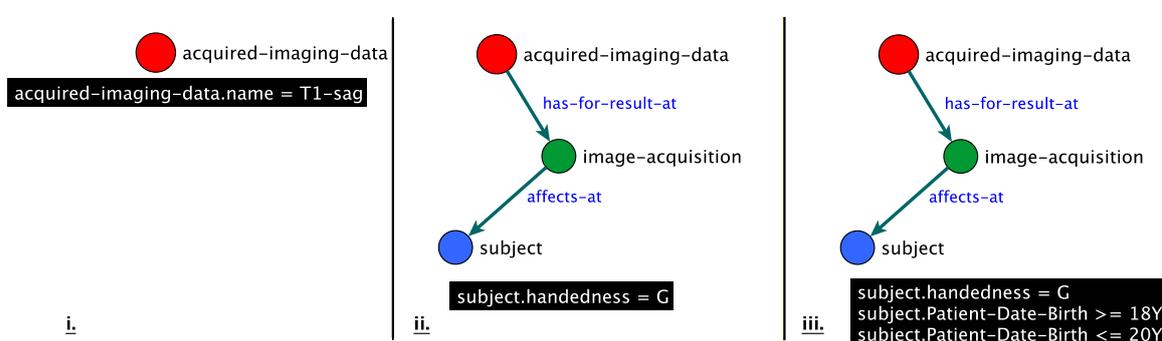


FIGURE 4.21 : La requête 02 est formulée au fur et à mesure en trois étapes.

objectName	Comments	Quality control	Body Part Examined	Repetition Time	Inversion Time	Pixel Spacing
ACQima_3D_T1_SAG_ISHARE_V01	20160421	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01	20160421	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01		2	BRAIN	2000	880	1 1
ACQima_3D_T1_SAG_ISHARE_V01	20160427	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01	20160427	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01	20160427	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01	20160428	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01	20160428	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01		2	BRAIN	2000	880	1 1
ACQima_3D_T1_SAG_ISHARE_V01	20160422	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01	20160211	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01	20160204	2	BRAIN	2000	880	1;1
ACQima_3D_T1_SAG_ISHARE_V01		2	BRAIN	2000	880	1 1

TABLE 4.1 : Extrait de résultats de la requête 02 sous la forme d'un tableau .csv.

4.4 DISCUSSION

Nous avons fait trois sessions de teste avec 08 chercheurs au GIN : 1 technicien, 3 chercheurs en psychologie, 3 chercheurs en imagerie et 1 directeur de recherche. Chaque participant était invité à sélectionner deux requêtes parmi des requêtes fréquemment utilisées. Les requêtes sélectionnées sont formulées sur VAQUERO par le participant lui-même. Les résultats montrent que la plupart des participants peuvent trouver eux-mêmes les données nécessaires

dans un temps pertinent. L'exemple présenté ci-dessus est réalisé en 5 minutes par un scientifique en imagerie après une courte formation de l'interface de requête.

Après trois sessions de tests, nous avons constaté que le système de requête VAQUERO améliore considérablement l'accès aux données de tous les chercheurs du laboratoire. Cependant, l'utilité et l'utilisabilité du système peuvent être améliorées afin d'accélérer le temps de formulation des requêtes, surtout des requêtes complexes et conjonctives.

Nous résumons les **points forts** du système :

- Un **système de requête visuel, dynamique et interactif**.
- Un système de requête **dédié aux différents groupes d'utilisateurs variés des niveaux de compétences techniques**.
- La **formulation des requêtes utilisateur est aisée et intuitive**.
- Les **données BMI sont accessibles sémantiquement** au travers de l'interface de requête (la mise en page "Ontologie"). Ce travail présenté dans cette thèse montre le succès des principes du Web sémantique dans la gestion de données complexes et hétérogènes.

Les **points à améliorer** :

- Les *interactions entre les différentes mises en page* doivent être développées et implémentées pour assurer une continuité du processus de formulation de données. C'est-à-dire, la requête en cours de formulation dans une mise en page peut être complétée et enrichie sur une autre mise en page.
- Le *moteur d'inférence doit être implémenté*.
- L'*ergonomie* de l'interface de requête doit être *améliorée*.

Les *ontologies proposées* doivent être *testées* et *validées* pour assurer une utilisation judicieuse des données stockées, éliminer la duplication des données et contrôler l'évolution des données complexes. Ce travail présenté dans cette thèse montre le succès des principes du Web sémantique dans la gestion de données complexes et hétérogènes.

CONCLUSION

5

SOMMAIRE

5.1 CONTRIBUTIONS	152
5.2 LIMITATIONS ET PERSPECTIVES	153
5.2.1 Limitations	153
5.2.2 Perspectives	154
5.3 LES CONTRIBUTIONS SCIENTIFIQUES	156

Depuis l'antiquité, les philosophes ont essayé d'expliquer le monde en répondant à des questions fondamentales telles que "*c'est quoi la vérité, le bien, le beau, le juste, la vie, le bonheur, etc.*". Ces questions divisent la philosophie en quatre branches : La **logique**, la **métaphysique**, l'**éthique**, l'**esthétique**⁷⁴. Etant la science de l'être en tant qu'être, la métaphysique (Heidegger 2014) recherche les principes de l'existence, la connaissance et comment faire pour avoir des connaissances.

Au fil du temps, la *connaissance* est toujours un des termes les plus abstraites des vocabulaires humains (Jakus et al. 2013), et jusqu'à maintenant, nous n'avons pas encore une définition de la connaissance qui soit largement acceptée. Chaque domaine a introduit sa propre définition de la connaissance qui est pilotée souvent par des caractéristiques du domaine. Dans la philosophie, Platon⁷⁵ a défini la connaissance comme une "**vraie croyance justifiée**" ("*justified true belief*"). Si "*une croyance est une idée que nous jugeons être vraie ou très probablement vraie indépendamment qu'elle soit effectivement vraie ou fausse*"⁷⁶, une connaissance est une croyance justifiée. C'est-à-dire, une personne connaît une proposition pour être vraie si (et seulement si) il ou elle croit en la vérité de la proposition et en même temps à la justification pour le faire (Jakus et al. 2013). En informatique, les scientifiques considèrent la connaissance sous un point de vue plus formel et plus programmatique. Les définitions introduites relient souvent aux autres termes comme "**donnée**", "**information**", "**sagesse**", "**savoir-faire**", et elles sont variées en fonction des disciplines d'application : *système d'information* (Guarino et others 1998), *système expert* (Matkar et Parab 2011), *intelligence artificielle* (Grüninger et al. 1995).

La connaissance joue un rôle très important dans les systèmes informatiques actuels. La représentation de connaissances désigne "*l'utilisation de symboles formels pour représenter une collection de propositions que croit un agent putatif*", elle joue comme une représentation interne de la réalité dans un système d'intelligence artificielle (Brachman et al. 1992). Fournissant un consensus sur des concepts représentant la réalité, l'ontologie est le coeur de la représentation de connaissances en ingénierie des connaissances (Studer et al. 1998). Cependant, une ontologie est limitée dans son nombre de concepts et relations, elle doit fournir des moyens pour produire des nouvelles connaissances à partir des connaissances qu'elle décrit explicitement (les mécanismes de raisonnement).

Par ailleurs, Le Web Sémantique (Berners-Lee et al. 2001), une extension du Web actuel, partage des objectifs communs avec la représentation de connaissances. Sa vision est de permettre l'interopérabilité sémantique et l'interprétation des ensembles de données à partir de diverses sources et de fournir les mécanismes permettant à ces données d'être utilisées comme support d'utilisation des données web d'une manière semi-automatisée et intelligente (Jakus et al. 2013).

Les données rendues disponibles dans les systèmes d'information actuels deviennent de plus en plus nombreuses, complexes et hétérogènes. Des nouvelles techniques de gestion de données telle que les BDD non-SQL, ou des nouvelles méthodes de recherche d'information (re-

74. <http://mapage.clg.qc.ca/danielfortier/partiel/partiel.htmv>.

75. <https://en.wikipedia.org/wiki/Plato>

76. <http://mapage.clg.qc.ca/danielfortier/partie5/partie5.htm>

cherche par mots clés ou *métadonnées*, *recherche sémantique*) sont proposées. Cependant, les données sont encore restées inexploitées et difficilement accessibles par les acteurs métiers de plusieurs domaines.

Les travaux de cette thèse répondent à certains aspects de cette interopérabilité sémantique en mettant en avant la problématique de la multi-conceptualisation et la multi-utilisation de données dans un contexte de collaboration.

5.1 CONTRIBUTIONS

Les travaux de recherche présentés dans le cadre de ce manuscrit comprennent des notions scientifiques différentes concernant l'ingénierie de connaissances, la représentation de connaissances, les technologies du Web Sémantique et la gestion de données. La question de recherche que nous avons adressée est de savoir comment faciliter l'accès et l'exploitation de données complexes et hétérogènes dans le contexte de travail collaboratif (*multi-utilisation de données*) des différents acteurs métiers (les *utilisateurs non-techniciens*). Les questions adressées sont :

- *Comment permettre une exploration liée au métier des utilisateurs de la base de données ?*
- *Comment amorcer un apprentissage et un partage des concepts considérant une évolution continue de la base de données ?*

A la suite d'une étude approfondie des éléments bibliographiques nous identifions que, malgré le fait que les outils de gestion de données actuels permettent de gérer des volumes importants de données, il reste néanmoins le problème de la structuration de ces données qui est proche d'une représentation technique permettant leur exploitation informatique. La représentation informatique de données impose une structuration spécifique qui rend les données difficilement accessibles et inexploitées par les acteurs métiers de plusieurs domaines, autrement dit, elle ne permet pas la multi-utilisation de données des utilisateurs non-techniciens.

Nous constatons que le problème d'accès aux données des utilisateurs vient du gap entre la représentation technique (le **point de vue informatique**) de données et les compétences métiers (le **point de vue métier**) des utilisateurs finaux. Pour surmonter ce gap, notre approche est d'utiliser une surcouche - une couche intermédiaire au niveau conceptuel faisant un lien entre ces deux points de vue.

- Sur la couche conceptuelle, nous proposons d'utiliser les ontologies d'application pour représenter explicitement les sémantiques et les logiques de données. Cette représentation doit être intuitive, visuelle et orientée utilisateur.
- Sur la couche conceptuelle, les acteurs métiers expriment leurs besoins d'information en termes de concepts ontologiques au lieu d'utiliser des langages de requête formels. Il faut avoir donc un ensemble de mappings permettant de transformer la requête utilisateur en une requête formelle.

Les contributions scientifiques de nos travaux peuvent se conclure comme suit :

- Nous proposons tout d'abord une démarche claire et concise pour accéder aux données selon une approche sémantique. Cette démarche est divisée en quatre fonctions principales : **Représentation des logiques de données par des ontologies d'application**,

Formulation graphique des requêtes utilisateurs, Transformation de requêtes utilisateurs en requête formelle et Transformation de résultats en instances visuelles.

- Sur la couche conceptuelle, nous mettons en évidence l'importance **d'avoir une représentation visuelle, graphique et intuitive des ontologies d'application** qui améliore nettement les capacités des utilisateurs en formulant des requêtes et en explorant des données.
- Les besoins d'information des utilisateurs sont exprimés en termes de concepts ontologiques, le langage de requête SPARQL est utilisé pour représenter les requêtes utilisateur initiale. Cependant, nous assumons que ces dernières devraient être formulées graphiquement. Nous proposons donc un **ensemble de notions graphiques** permettant d'exprimer les besoins d'information des utilisateurs en un **graphe "noeud-arc"**. Nous précisons ensuite des règles de transformations pour transformer le graphe en requête SPARQL.
- L'approche sémantique pour accéder aux données est basée principalement sur les ontologies d'application et les mappings entre eux et les schémas de données. **Nous proposons donc une démarche de construction d'ontologies d'application et de construction de mappings** qui est **concise, simple, reproductible et réutilisable**.

Nous impliquons qu'une représentation logique, sémantique et visuelle de données est nécessaire (par exemple pour aider les personnes à former leurs pensées et leurs besoins d'information, etc.). Cependant, pour assurer la collaboration des acteurs multi-discipline et la traçabilité des évolutions des données, les données doivent être organisées autour d'un objet d'étude principal. Par exemple, toutes les données BMI (les données d'acquisition ou de traitement, les données psychologies des sujets, etc.) sont organisées autour d'une étude. De même, l'ensemble des données gérées dans la conception de produits reflètent l'évolution du produit.

5.2 LIMITATIONS ET PERSPECTIVES

5.2.1 Limitations

Nous mettons en avant certaines limites des travaux de recherche présentés dans ce manuscrit selon trois points principaux à améliorer :

- Les modèles ontologies orientées-disciplines proposées sont simples. Elles contiennent seulement les concepts les plus généraux qui sont utilisés fréquemment par les acteurs métiers de la discipline. En réalité, ces ontologies doivent être plus détaillées et complètes pour que les utilisateurs finaux puissent identifier rapidement les concepts cibles correspondants à leurs besoins d'informations. Le domaine Biomédicale, Neuroimagerie en particulier, est complexe. L'ontologie du domaine peut contenir des centaines, voire des milliers de concepts qui sont interconnectés. Etant construite à base de l'ontologie du domaine et les modèles conceptuels de données, les ontologies d'applications utilisées doivent être suffisamment riches. Quand nous ajoutons des nouveaux sous-concepts, les relations entre concepts que nous avons proposées doivent être également

enrichies avec l'aide des experts du domaine. Plus les ontologies d'application sont complètes et cohérentes, plus l'utilisation de données est facile.

- Un point fort de l'utilisation de l'ontologie pour la représentation est le "**raisonnement**". Dans le cadre de notre thèse, nous n'avons pas bien profité des capacités de ce mécanisme. En fait, le raisonnement permet non seulement de vérifier la pertinence des connaissances déclarées dans l'ontologie mais aussi d'en produire de nouvelles. Des techniques de Data Mining et de Machine Learning peuvent être utilisés à ce propos.
- La méthodologie et la démarche proposées dans notre étude ont été appliquées seulement dans le contexte du PLM pour l'Imagerie Biomédicale. Pour tester la méthodologie, nous avons assumé que les données PLM peuvent être vues comme des contenus d'un document XML, le xQuery a donc été utilisé comme le langage de requête formelle sur les données. En fait, la méthodologie ne doit pas être restreinte à un domaine particulier, une représentation particulière de base de données ou un langage de requête prédéfini. Elle doit être applicable dans des domaines différents et variés de structuration de données. Les sources de données peuvent être distribuées, stockées et interrogées par des langages variés de requêtes formelles différentes. Dans ce cas-là, la démarche de formulation et de traitement de requête ne change pas, par contre, nous utilisons des ensembles différents de mappings sur chaque source de données.

5.2.2 Perspectives

Les limitations présentées ci-dessus nous permettent de proposer les perspectives suivantes :

A court terme :

- Nous envisageons tout d'abord d'améliorer la performance du système de requête VA-QUERO par l'implémentation de nouvelles techniques de visualisation où les ontologies et les résultats de requête seront représentés dans un espace 3D. Il faut enrichir les ontologies proposées et les tester avec des requêtes complexes différentes.
- Nous améliorerons ensuite l'aspect "**usage**" du système de requête pour s'assurer qu'il est non seulement intuitif et facile à utiliser mais aussi suffisamment performant en fonction des besoins variés des acteurs métiers.

A moyen terme

- La méthodologie proposée a été testée avec des données BMI gérées par les solutions PLM. Evidemment, les solutions PLM ne sont pas le seul moyen permettant de gérer des données complexes et hétérogènes. Les acteurs métiers dans des domaines multiples ont également besoin d'exploitation et de multi-utilisation de données. La méthodologie proposée devrait être testée avec des cas d'utilisation différents pour valider sa faisabilité.
- Les sources de données et les modèles ontologiques d'application peuvent régulièrement évoluer. L'ajout de nouvelles données ou la suppression d'un concept ontologique demande des modifications sur les mappings entre les ontologies d'application et les sources de données. Nous envisageons de développer un service permettant la mise à jour semi-automatique des mappings.

A long terme

- Une fois que les acteurs métiers peuvent accéder aux données et aux informations qu'ils souhaitent, il faut leur fournir des moyens support permettant d'explorer efficacement ces données. Bien que les outils de visualisation peuvent aider les acteurs métiers à mieux comprendre les données en profitant des avantages de la représentation graphique et des mécanismes d'interaction, il est nécessaire d'implémenter des techniques issues de l'intelligence artificielle comme les algorithmes de l'apprentissage supervisée en particulier.
- Dans le contexte de travail collaboratif entre des acteurs multi-métiers, multi-disciplines, les modifications sur les données ou l'ajout des nouvelles données délivrées (résultats d'un traitement de données) effectués par un individu devraient être notifiés à tous les autres. Un système d'alerte à ajouter dans le système VAQUERO serait utile pour assurer la traçabilité de l'ensemble de données.

5.3 LES CONTRIBUTIONS SCIENTIFIQUES

Les travaux de thèse ont fait l'objet des publications scientifiques suivantes :

- Pham, Cong Cuong, Nada Matta, Alexandre Durupt, Benoit Eynard, and Guillaume Ducellier. 2015. "*Knowledge Sharing in Heterogeneous Data Context : Application in PLM.*" DS 80-10 Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 10 : Design Information and Knowledge Management Milan, Italy, 27-30.07.15.
- Pham, Cong Cuong, Alexandre Durupt, Nada Matta, Benoît Eynard, and Guillaume Ducellier. 2015. "*Knowledge Sharing Using Product Life Cycle Management.*" In Advances in Production Management Systems : Innovative Production Management Towards Sustainable Growth - IFIP WG 5.7 International Conference, APMS 2015, Tokyo, Japan, September 7-9, 2015, Proceedings, Part I, 376-382. doi :[10.1007/978-3-319-22756-6_46](https://doi.org/10.1007/978-3-319-22756-6_46).
- Pham, Cong Cuong, Nada Matta, Alexandre Durupt, Benoit Eynard, Marianne Allanic, Guillaume Ducellier, Marc Joliot, and Philippe Boutinaud. 2015. "*Sharing Knowledge in Daily Activity : Application in Bio-Imaging.*" In Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 242-47. doi :[10.5220/0005633002420247](https://doi.org/10.5220/0005633002420247).
- Pham, Cong Cuong, Alexandre Durupt, Nada Matta, and Benoit Eynard. 2016. "*Knowledge Sharing Using Ontology Graph-Based : Application in PLM and Bio-Imaging Contexts.*" In Product Lifecycle Management in the Era of Internet of Things, edited by Abdelaziz Bouras, Benoit Eynard, Sebti Foufou, and Klaus-Dieter Thoben, 467 :238-47. Cham : Springer International Publishing. http://link.springer.com/10.1007/978-3-319-33111-9_22.
- Pham, Cong Cuong, Nada Matta, Alexandre Durupt, Benoit Eynard, Marianne Allanic, Guillaume Ducellier, Marc Joliot, and Philippe Boutinaud. 2016. "*Using Ontologies to Access Complex Data : Applications on Bio-Imaging.*" In Proceedings of AI4KM Workshop, IJCAI, 6.

ACRONYMES

BDD Base de Données.

BIOMIST BIO Medical research Imaging SemanTic data management.

BMI-LM Biomedical Imaging Lifecycle Management.

DBMS Database Management System.

GIN Imaging Group Neurofunctional, CEA CNRS Université de Bordeaux.

ICT Information and Comunication Technologies.

MDDR Modèle De Données Relationnelles.

OBDA Ontology-Based Data Access.

OWL Ontology Web Language.

PDM Product Data Management.

PLM Product Lifecycle Management.

R2RML RDF to RDF Mapping Language.

RDF Resource Description Framework.

SGDT Systèmes de Gestion des Données Techniques.

SPARQL SPARQL Protocol and RDF Query Language).

TIC Technologies de l'Information et de la Communication.

URI Uniform Ressource Identifier.

UTC Université de Technologie de Compiègne.

UTT Université de Technologie de Troyes.

VAQUERO VisuAlisation et QUERy based Ontology.

VQS Système d'Interrogation visuel.

BIBLIOGRAPHIE

- Russell L Ackoff. From data to wisdom. Journal of applied systems analysis, 16(1) :3–9, 1989. bibtex : ackoff1989data. (Cité page 19.)
- Sanjay Agrawal, Surajit Chaudhuri, et Gautam Das. DBXplorer : Enabling Keyword Search over Relational Databases. Dans Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, SIGMOD '02, pages 627–627, New York, NY, USA, 2002. ACM. ISBN 978-1-58113-497-1. URL <http://doi.acm.org/10.1145/564691.564782>. (Cité page 43.)
- A. M. Alashqur, S. Y. W. Su, et H. Lam. OQL : A Query Language for Manipulating Object-oriented Databases. Dans Proceedings of the 15th International Conference on Very Large Data Bases, VLDB '89, pages 433–442, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-101-7. URL <http://dl.acm.org/citation.cfm?id=88830.88909>. (Cité page 42.)
- Marianne Allanic. Gestion et visualisation de données hétérogènes multidimensionnelles : application PLM à la neuroimagerie. Theses, Université de Technologie de Compiègne, Décembre 2015. URL <https://hal.archives-ouvertes.fr/tel-01316896>. (Cité pages 3, 4, 6, 33, 34, 112, 133 et 134.)
- Marianne Allanic, Alexandre Durupt, Marc Joliot, Benoît Eynard, et Philippe Boutinaud. Application of PLM for Bio-Medical Imaging in Neuroscience. Dans Alain Bernard, Louis Rivest, et Debasish Dutta, éditeurs, Product Lifecycle Management for Society, numéro 409 dans IFIP Advances in Information and Communication Technology, pages 520–529. Springer Berlin Heidelberg, Juillet 2013. ISBN 978-3-642-41500-5 978-3-642-41501-2. URL http://link.springer.com/chapter/10.1007/978-3-642-41501-2_52. DOI : 10.1007/978-3-642-41501-2_52. (Cité pages 111, 133, 134 et 137.)
- Marianne Allanic, Pierre-Yves Hervé, Alexandre Durupt, Marc Joliot, Philippe Boutinaud, et Benoît Eynard. PLM As a Strategy for the Management of Heterogeneous Information in Bio-medical Imaging Field. Int. J. Inf. Technol. Manage., 16(1) :5–30, Janvier 2016. ISSN 1461-4111. URL <https://doi.org/10.1504/IJITM.2017.080950>. (Cité page 106.)
- Marianne Allanic, Pierre-Yves Hervé, Cong-Cuong Pham, Myriam Lekkal, Alexandre Durupt, Thierry Brial, Arthur Grioche, Nada Matta, Philippe Boutinaud, Benoit Eynard, et Marc Joliot. BIOMIST : A Platform for Biomedical Data Lifecycle Management of Neuroimaging Cohorts. Frontiers in ICT, 3 :35, 2017. ISSN 2297-198X. URL <http://journal.frontiersin.org/article/10.3389/fict.2016.00035>. (Cité pages 4 et 6.)

- Ion Androutsopoulos, Graeme D. Ritchie, et Peter Thanisch. Natural Language Interfaces to Databases - An Introduction. CoRR, cmp-lg/9503016, 1995. URL <http://arxiv.org/abs/cmp-lg/9503016>. (Cité page 43.)
- Ibrahim Assouroko, Guillaume Ducellier, Benoît Eynard, Philippe Boutinaud, et Julien Le Duigou. Amélioration de la gestion collaborative des connaissances du produit à partir de technologies du Web Sémantique. Dans 7ème Conférence Internationale Conception et Production Intégrée, Oujda, Morocco, Octobre 2011. URL <https://hal.archives-ouvertes.fr/hal-00797984>. (Cité page 21.)
- Irina Astrova et Ahto Kalja. Mapping of SQL relational schemata to OWL ontologies. Dans Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Greece, 2006. (Cité page 83.)
- Sören Auer, Sebastian Dietzold, Jens Lehmann, Sebastian Hellmann, et David Aumueller. Triplify : Light-weight Linked Data Publication from Relational Databases. Dans Proceedings of the 18th International Conference on World Wide Web, WWW '09, pages 621–630, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. URL <http://doi.acm.org/10.1145/1526709.1526793>. (Cité page 124.)
- Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, et Peter F. Patel-Schneider, éditeurs. The Description Logic Handbook : Theory, Implementation, and Applications. Cambridge University Press, New York, NY, USA, 2003. ISBN 978-0-521-78176-3. (Cité pages 49 et 61.)
- Bartosz Balis, Marian Bubak, et Tomasz Grabiec. Graphical Query Construction over Scientific Data Sets Using Semantic Technologies. 2012. (Cité pages 74 et 76.)
- Jesus Barrasa, Oscar Corcho, et Asunción Gómez-Pérez. Fund Finder : A case study of database-to-ontology mapping. Dans Semantic Integration Workshop (SI-2003), page 9, 2003. (Cité pages 105, 119, 120 et 122.)
- Jesús Barrasa Rodríguez, Oscar Corcho, et A. Gómez-Pérez. Fund Finder : A case study of database-to-ontology mapping. Dans Proceedings of the 2nd International Semantic Web Conference, ISWC2003, volume 82, Florida, USA, Octobre 2003. Facultad de Informática (UPM). URL <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-82/>. (Cité page 118.)
- Ori Ben-Yitzhak, Nadav Golbandi, Nadav Har'El, Ronny Lempel, Andreas Neumann, Shila Ofek-Koifman, Dafna Sheinwald, Eugene Shekita, Benjamin Sznajder, et Sivan Yogev. Beyond Basic Faceted Search. Dans Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08, pages 33–44, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-927-2. URL <http://doi.acm.org/10.1145/1341531.1341539>. (Cité page 45.)
- Amaia Bernaras, Iñaki Laresgoiti, et Jose Manuel Corera. Building and Reusing Ontologies for Electrical Network Applications. Dans Wolfgang Wahlster, éditeur, 12th

- European Conference on Artificial Intelligence, Budapest, Hungary, August 11-16, 1996, Proceedings, pages 298–302. John Wiley and Sons, Chichester, 1996. bibtex : DBLP :conf/ecai/BernarasLC96. (Cité pages 53 et 63.)
- Tim Berners-Lee. Relational Databases and the Semantic Web (in Design Issues). Dans in Design Issues of the WWW, 1998. URL <https://www.w3.org/DesignIssues/RDB-RDF.html>. (Cité page 123.)
- Tim Berners-Lee, James Hendler, Ora Lassila, et others. The semantic web. Scientific american, 284(5) :28–37, 2001. (Cité pages 3, 6, 77 et 151.)
- Stefan Bertschi, Sabrina Bresciani, Tom Crawford, Randy Goebel, Wolfgang Kienreich, Martin Lindner, Vedran Sabol, et Andrew Vande Moere. What Is Knowledge Visualization? Eight Reflections on an Evolving Discipline. Dans Francis T. Marchese et Ebad Banissi, éditeurs, Knowledge Visualization Currents : From Text to Art to Culture, pages 13–32. Springer London, London, 2013. ISBN 978-1-4471-4303-1. URL http://dx.doi.org/10.1007/978-1-4471-4303-1_2. (Cité pages 47, 66 et 68.)
- Christian Bizer. D2R MAP - A Database to RDF Mapping Language. 2003. (Cité pages 78, 82, 83, 121 et 128.)
- Christian Bizer et Andy Seaborne. D2rq-treating non-RDF databases as virtual RDF graphs. Dans Proceedings of the 3rd international semantic web conference (ISWC2004), volume 2004. Springer, 2004. (Cité pages 121 et 122.)
- Jethro Borsje et Hanno Embregts. Graphical query composition and natural language processing in an RDF visualization interface. Erasmus School of Economics and Business Economics, Vol. Bachelor. Erasmus University, Rotterdam, 2006. (Cité page 76.)
- Willem Nico Borst. Construction of Engineering Ontologies for Knowledge Sharing and Reuse. phdthesis, Universiteit Twente, Septembre 1997. URL <http://doc.utwente.nl/17864/>. bibtex : Borst1997. (Cité pages 3, 6 et 53.)
- Ronald Brachman et Hector Levesque. Knowledge Representation and Reasoning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. ISBN 978-0-08-048932-2. (Cité page 51.)
- Ronald J Brachman, Hector J Levesque, et Raymond Reiter. Knowledge representation. MIT press, 1992. (Cité page 151.)
- Matthieu Bricogne-Cuignières. Méthode agile pour la conception collaborative multidisciplinaire de systèmes intégrés : application à la mécatronique. PhD thesis, Université de Technologie de Compiègne, 2015. (Cité page 29.)
- Andrew J. Buckler, Tiffany Ting Liu, Erica Savig, Baris E. Suzek, M. Ouellette, J. Danagoulian, G. Wernsing, Daniel L. Rubin, et David Paik. Quantitative imaging biomarker ontology (QIBO) for knowledge representation of biomedical imaging biomarkers. Journal of Digital Imaging, 26(4) :630–641, Août 2013. ISSN 1618-727X. (Cité page 113.)

- R. A. Burkhard. Learning from architects : the difference between knowledge visualization and information visualization. Dans Eighth International Conference on Information Visualisation, 2004. IV 2004. Proceedings, pages 519–524, Juillet 2004. (Cité page 67.)
- Remo Aslak Burkhard. Knowledge visualization - The Use of Complementary Visual Representations for the Transfer of Knowledge. A Model, a Framework, and Four New Approaches. PhD thesis, 2005. (Cité page 67.)
- Margaret M. Burnett. Visual Programming. Dans Wiley Encyclopedia of Electrical and Electronics Engineering. John Wiley & Sons, Inc., 2001. ISBN 978-0-471-34608-1. URL <http://onlinelibrary.wiley.com/doi/10.1002/047134608X.W1707/abstract>. (Cité page 45.)
- Margaret M. Burnett et Marla J. Baker. A Classification System for Visual Programming Languages. Journal of Visual Languages & Computing, 5(3) :287–300, Septembre 1994. ISSN 1045-926X. URL <http://www.sciencedirect.com/science/article/pii/S1045926X84710159>. (Cité page 45.)
- Diego Calvanese. Ontology-Based Data Access : From Theory to Practice, Décembre 2012. URL http://bda2012.isima.fr/keynote_calvanese. (Cité page 43.)
- Diego Calvanese, Pietro Liuzzo, Alessandro Mosca, José Remesal, Martin Rezk, et Guillem Rull. Ontology-based data integration in EPNet : Production and distribution of food during the Roman Empire. Engineering Applications of Artificial Intelligence, 51 :212 – 229, 2016. ISSN 0952-1976. URL <http://www.sciencedirect.com/science/article/pii/S0952197616000099>. Mining the Humanities : Technologies and Applications bibtex : Calvanese2016212. (Cité page 86.)
- Stuart K. Card, Jock Mackinlay, et Ben Shneiderman. Readings in Information Visualization : Using Vision to Think. Morgan Kaufmann, San Francisco, Calif., 1999. ISBN 978-1-55860-533-6. (Cité page 67.)
- Tiziana Catarci. What Happened When Database Researchers Met Usability. 2000. (Cité pages 43 et 45.)
- Tiziana Catarci, Maria F. Costabile, Stefano Levialdi, et Carlo Batini. Visual Query Systems for Databases : A Survey. Journal of Visual Languages & Computing, 8(2) :215–260, Avril 1997. ISSN 1045-926X. URL <http://www.sciencedirect.com/science/article/pii/S1045926X97900379>. (Cité pages 38, 42 et 45.)
- Christopher F. Chabris et Stephen M. Kosslyn. Representational Correspondence as a Basic Principle of Diagram Design. Dans Sigmar-Olaf Tergan et Tanja Keller, éditeurs, Knowledge and Information Visualization, numéro 3426 dans Lecture Notes in Computer Science, pages 36–57. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26921-2 978-3-540-31962-7. URL http://link.springer.com/chapter/10.1007/11510154_3. DOI : 10.1007/11510154_3. (Cité page 95.)

- Kulkarni Chaitali, Divate Prachi, Kale Ashwini, et Kate Aakanksha. Natural Language Interface To Database. International Research Journal of Engineering and Technology (IRJET), 03(10), Octobre 2016. URL <https://www.irjet.net/volume3-issue10>. (Cité page 44.)
- Soumen Chakrabarti, Sunita Sarawagi, et S Sudarshan. Enhancing Search with Structure. 2010. (Cité pages 43 et 44.)
- Donald D. Chamberlin et Raymond F. Boyce. SEQUEL : A Structured English Query Language. Dans Proceedings of the 1974 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control, SIGFIDET '74, pages 249–264, New York, NY, USA, 1974. ACM. URL <http://doi.acm.org/10.1145/800296.811515>. (Cité page 42.)
- Michel Chein, Marie-Laure Mugnier, et Madalina Croitoru. Visual reasoning with graph-based mechanisms : the good, the better and the best. The Knowledge Engineering Review, 28(03) :249–271, Septembre 2013. ISSN 0269-8889, 1469-8005. URL http://www.journals.cambridge.org/abstract_S0269888913000234. (Cité page 47.)
- Chaomei Chen. Information Visualization : Beyond the Horizon. Springer, New York ; London, 2nd edition édition, Février 2010. ISBN 978-1-84628-340-6. (Cité page 44.)
- Peter Chen. Entity-Relationship Modeling : Historical Events, Future Trends, and Lessons Learned. Dans Prof Dr Manfred Broy et Prof Dr Ernst Denert, éditeurs, Software Pioneers, pages 296–310. Springer Berlin Heidelberg, 2002. ISBN 978-3-642-63970-8 978-3-642-59412-0. URL http://link.springer.com/chapter/10.1007/978-3-642-59412-0_17. DOI : 10.1007/978-3-642-59412-0_17. (Cité page 39.)
- Peter Pin-Shan Chen. The Entity-relationship Model? Toward a Unified View of Data. ACM Trans. Database Syst., 1(1) :9–36, Mars 1976. ISSN 0362-5915. URL <http://doi.acm.org/10.1145/320434.320440>. (Cité pages 39 et 111.)
- Y. Chen, W. Wang, et Z. Liu. Keyword-based search and exploration on databases. Dans 2011 IEEE 27th International Conference on Data Engineering, pages 1380–1383, Avril 2011. (Cité page 43.)
- Yi Chen, Wei Wang, Ziyang Liu, et Xuemin Lin. Keyword Search on Structured and Semi-structured Data. Dans Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, SIGMOD '09, pages 1005–1010, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-551-2. URL <http://doi.acm.org/10.1145/1559845.1559966>. (Cité page 44.)
- Pasapitch Chujai, Nittaya Kerdprasop, et Kittisak Kerdprasop. On Transforming the ER Model to Ontology Using Protégé OWL Tool. International Journal of Computer Theory and Engineering, 6(6) :484–489, Décembre 2014. ISSN 17938201. (Cité page 111.)
- Edgar F Codd. A relational model of data for large shared data banks. Communications of the ACM, 13(6) :377–387, 1970. (Cité pages 38 et 40.)

- Richard Cox. Representation construction, externalised cognition and individual differences. Learning and instruction, 9(4) :343–363, 1999. (Cité page 93.)
- Edward Craig. Ontology. 1998. URL <https://www.rep.routledge.com/articles/thematic/ontology/v-1>. (Cité page 51.)
- Madalina Croitoru. Conceptual graphs at work : efficient reasoning and applications. PhD thesis, Citeseer, 2006. (Cité page 66.)
- Souripriya Das, Seema Sundara, et Richard Cyganiak. R2RML : RDB to RDF Mapping Language. Septembre 2012. URL <http://www.w3.org/TR/r2rml>. (Cité pages 125 et 126.)
- C. J. Date. An Introduction to Database Systems. Pearson, Boston, 8 edition édition, Août 2003. ISBN 978-0-321-19784-9. (Cité page 42.)
- C.J. J. Date. An Introduction to Database Systems. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd édition, 1981. ISBN 978-0-201-14471-0. (Cité page 42.)
- Randall Davis. Knowledge-based systems. Science, 231(4741) :957–963, 1986. (Cité pages 11 et 54.)
- Randall Davis, Howard Shrobe, et Peter Szolovits. What Is a Knowledge Representation ? AI Magazine, 14(1) :17, Mars 1993. ISSN 0738-4602. URL <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1029>. (Cité pages 47 et 51.)
- Richard De Courcy. Les systèmes d'information en réadaptation. Québec, Réseau International CIDIH et facteurs environnementaux, 1(5) :7–10, 1992. (Cité page 20.)
- Cristian Pérez de Laborda et Stefan Conrad. Relational.OWL : A Data and Schema Representation Format Based on OWL. Dans Proceedings of the 2Nd Asia-Pacific Conference on Conceptual Modelling - Volume 43, APCCM '05, pages 89–96, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc. ISBN 1-920682-25-2. URL <http://dl.acm.org/citation.cfm?id=1082276.1082287>. (Cité pages 123 et 124.)
- Koenraad De Smedt. Knowledge Representation Techniques in Artificial Intelligence : An Overview. Dans Gerrit C. van der Veer et Gijsbertus Mulder, éditeurs, Human-Computer Interaction : Psychonomic Aspects, pages 207–222. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988. ISBN 978-3-642-73402-1. URL http://dx.doi.org/10.1007/978-3-642-73402-1_13. (Cité page 92.)
- Mike Dean, Guus Schreiber, Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L McGuinness, Peter F Patel-Schneider, et L Andrea Stein. OWL web ontology language reference. W3C Recommendation February, 10, 2004. (Cité pages 61 et 62.)
- Denis Debaecker. PLM, la gestion collaborative du cycle de vie des produits. Hermes Science Publications, Paris, 2004. ISBN 978-2-7462-0884-1. (Cité pages 29 et 30.)

- Tharam Dillon, Elizabeth Chang, Maja Hadzic, et Pornpit Wongthongtham. Differentiating conceptual modelling from data modelling, knowledge modelling and ontology modelling and a notation for ontology modelling. Dans Conferences in Research and Practice in Information Technology Series, volume 79, pages 7–17. Australian Computer Society, Inc., 2008. ISBN 978-1-920682-60-6. bibtex : dillon2008differentiating. (Cité page 55.)
- Patrick Doherty, Witold Lukaszewicz, Andrzej Skowron, et Andrzej Szalas. Knowledge Representation Techniques. : a rough set approach. Springer, 2006. (Cité page 92.)
- Catherine Dolbear et John Goodwin. Position paper on expressing relational data as RDF. Dans W3C Workshop on RDF Access to Relational Databases, pages 25–26. Citeseer, 2007. (Cité page 119.)
- John Domingue, Dieter Fensel, et James A Hendler. Handbook of semantic web technologies. Springer Science & Business Media, 2011. (Cité page 38.)
- Marek Dudáží, Ondřej Zamazal, et Vojtěch Svátek. Roadmapping and Navigating in the Ontology Visualization Landscape. Dans Krzysztof Janowicz, Stefan Schlobach, Patrick Lambrix, et Eero Hyvönen, éditeurs, Knowledge Engineering and Knowledge Management : 19th International Conference, EKAW 2014, Linköping, Sweden, November 24-28, 2014. Proceedings, pages 137–152. Springer International Publishing, Cham, 2014. ISBN 978-3-319-13704-9. URL http://dx.doi.org/10.1007/978-3-319-13704-9_11. DOI : 10.1007/978-3-319-13704-9_11. (Cité page 72.)
- Syeda Sana e Zainab, Ali Hasnain, Muhammad Saleem, Qaiser Mehmood, Durre Zehra, et Stefan Decker. FedViz : A Visual Interface for SPARQL Queries Formulation and Execution. 2015. (Cité page 76.)
- Marc Ehrig. Ontology alignment : bridging the semantic gap, volume 4. Springer Science & Business Media, 2006. (Cité page 106.)
- P. Eklund, N. Roberts, et S. Green. OntoRama : Browsing RDF ontologies using a hyperbolic-style browser. Dans First International Symposium on Cyber Worlds, 2002. Proceedings., pages 405–411, 2002. (Cité page 71.)
- Shady Elbassuoni, Maya Ramanath, Ralf Schenkel, et Gerhard Weikum. Searching RDF Graphs with SPARQL and Keywords. IEEE Data Eng. Bull., 33(1) :16–24, 2010. (Cité page 44.)
- Ramez Elmasri et Shamkant B. Navathe. Fundamentals of Database Systems. Pearson, Boston, Mass. ; Munich u.a., 6 edition édition, Avril 2010. ISBN 978-0-13-608620-8. (Cité pages 40 et 41.)
- Dr Martin J. Eppler. What Is an Effective Knowledge Visualization? Insights from a Review of Seminal Concepts. Dans Francis T. Marchese et Ebad Banissi, éditeurs, Knowledge Visualization Currents, pages 3–12. Springer London, 2013. ISBN 978-1-4471-4302-4 978-1-4471-4303-1. URL http://link.springer.com/chapter/10.1007/978-1-4471-4303-1_1. DOI : 10.1007/978-1-4471-4303-1_1. (Cité page 68.)

- Richard G. Epstein. The TableTalk query language. Journal of Visual Languages & Computing, 2(2) :115–141, Juin 1991. ISSN 1045-926X. URL <http://www.sciencedirect.com/science/article/pii/S1045926X05800266>. (Cité page 45.)
- Orri Erling et Ivan Mikhailov. RDF Support in the Virtuoso DBMS. Dans Tassilo Pellegrini, Søren Auer, Klaus Tochtermann, et Sebastian Schaffert, éditeurs, Networked Knowledge - Networked Media : Integrating Knowledge Management, New Media Technologies and Semantic Systems, pages 7–24. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-02184-8. URL <http://dx.doi.org/10.1007/978-3-642-02184-8₂>. bibtex : Erling2009. (Cité pages 105 et 124.)
- Jean-Louis Ermine. La gestion des connaissances. Hermes Lavoisier, 2003. URL <https://hal.archives-ouvertes.fr/hal-00997696>. (Cité page 108.)
- Muhammad Fahad. ER2owl : Generating OWL Ontology from ER Diagram. Dans Zhongzhi Shi, E. Mercier-Laurent, et D. Leake, éditeurs, Intelligent Information Processing IV : 5th IFIP International Conference on Intelligent Information Processing, October 19-22, 2008, Beijing, China, pages 28–37. Springer US, Boston, MA, 2008. ISBN 978-0-387-87685-6. URL http://dx.doi.org/10.1007/978-0-387-87685-6_6. DOI : 10.1007/978-0-387-87685-6_6. (Cité page 111.)
- Adam Farquhar, Richard Fikes, et James Rice. The Ontolingua Server : a Tool for Collaborative Ontology Construction. Dans International Journal of Human-Computer Studies, 1996. (Cité pages 61 et 62.)
- D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, et P. F. Patel-Schneider. OIL : an ontology infrastructure for the Semantic Web. IEEE Intelligent Systems, 16(2) :38–45, Mars 2001. ISSN 1541-1672. (Cité pages 61 et 116.)
- Dieter Fensel. Ontologies : A Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 édition, 2003. ISBN 978-3-540-00302-1. (Cité pages 56 et 78.)
- Dieter Fensel. Ontologies : A Silver Bullet for Knowledge Management and Electronic Commerce, volume 7. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-642-05558-4. URL <http://link.springer.com/10.1007/978-3-662-09083-1>. bibtex : Fensel2004b. (Cité pages 77 et 78.)
- M. Fernández-López, A. Gómez-Pérez, et N. Juristo. METHONTOLOGY : From Ontological Art Towards Ontological Engineering. Dans Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series, Stanford University, EEUU, Mars 1997. Facultad de Informática (UPM). URL <http://oa.upm.es/5484/>. (Cité page 63.)
- M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, et P. Yanker. Query by image and video content : the QBIC system. Computer, 28(9) :23–32, Septembre 1995. ISSN 0018-9162. (Cité page 42.)

- Frederico Fonseca et James Martin. Learning the differences between ontologies and conceptual schemas through ontology-driven information systems. Journal of the Association for Information Systems, 8(2) :129, 2007. bibtex : fonseca2007learning. (Cité pages 58 et 59.)
- Virginie Fortineau. Contribution à une modélisation ontologique des informations tout au long du cycle de vie du produit. phdthesis, Ecole nationale supérieure d'arts et métiers - ENSAM, Novembre 2013. URL <https://pastel.archives-ouvertes.fr/pastel-01064598/document>. (Cité page 30.)
- John Gantz et David Reinsel. The digital universe in 2020 : Big data, bigger digital shadows, and biggest growth in the far east. IDC iView : IDC Analyze the future, 2007 :1–16, 2012. (Cité page 22.)
- Raji Ghawi et Nadine Cullot. Database-to-ontology mapping generation for semantic interoperability. Dans VDBL'07 conference, VLDB Endowment ACM, pages 1–8, 2007. (Cité page 119.)
- Bernard Gibaud, Germain Forestier, Hugues Benoit-Cattin, Frédéric Cervenansky, Patrick Clarisse, Denis Friboulet, Alban Gaignard, Patrick Hugonnard, Carole Lartizien, Hervé Liebgott, Johan Montagnat, Joachim Tabary, et Tristan Glatard. OntoVIP : An ontology for the annotation of object models used for medical image simulation. Journal of Biomedical Informatics, 52 :279–292, 2014. ISSN 15320464. (Cité page 113.)
- Martin Giese, Diego Calvanese, Peter Haase, Ian Horrocks, Yannis Ioannidis, Herald Kllapi, Manolis Koubarakis, Maurizio Lenzerini, Ralf Möller, Mariano Muro, Özgür Özçep, Riccardo Rosati, Rudolf Schlatte, Michael Schmidt, Ahmet Soylu, et Arild Waaler. Scalable end-user access to big data. Dans Rajendra Akerkar, éditeur, Big Data Computing, pages 205–244. Chapman and Hall/CRC, 2013. ISBN 978-1-4665-7837-1 978-1-4665-7838-8. URL <http://www.crcnetbase.com/doi/abs/10.1201/b16014-9>. (Cité pages 25, 26 et 43.)
- Seth Gilbert et Nancy Lynch. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services. SIGACT News, 33(2) :51–59, Juin 2002. ISSN 0163-5700. URL <http://doi.acm.org/10.1145/564585.564601>. (Cité page 38.)
- Jean-Pierre Giraudin. Complexité des systèmes d'information et de leur ingénierie. e-TI - la revue électronique des technologies d'information, 3, 2007. URL <http://www.revueeti.net/document.php?id=1180>. (Cité page 23.)
- Stephan Grimm, Dr Andreas Abecker, Johanna Völker, et Rudi Studer. Ontologies and the Semantic Web. Dans John Domingue, Dieter Fensel, et James A. Hendler, éditeurs, Handbook of Semantic Web Technologies, pages 507–579. Springer Berlin Heidelberg, 2011. ISBN 978-3-540-92912-3 978-3-540-92913-0. URL http://link.springer.com/referenceworkentry/10.1007/978-3-540-92913-0_13. DOI : 10.1007/978-3-540-92913-0_13. (Cité pages 52, 53, 54, 55, 56, 57, 58, 59, 79, 80, 81, 82 et 84.)

- Stephan Grimm, Pascal Hitzler, et Andreas Abecker. Knowledge Representation and Ontologies ? Logic, Ontologies and Semantic Web Languages. 2004. (Cité page 51.)
- Thomas R Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition - Special issue : Current issues in knowledge modeling, 5(2) :199–220, Juin 1993. ISSN 10428143. URL <http://www.sciencedirect.com/science/article/pii/S1042814383710083><http://linkinghub.elsevier.com/retrieve/pii/S1042814383710083>. (Cité pages 53 et 57.)
- Michael Grüninger, Michael Grüninger, et Mark S. Fox. Methodology for the Design and Evaluation of Ontologies. 1995. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.8723>. (Cité pages 3, 6, 63 et 151.)
- Nicola Guarino, Daniel Oberle, et Steffen Staab. What Is an Ontology? Dans Steffen Staab et Rudi Studer, éditeurs, Handbook on Ontologies, International Handbooks on Information Systems, pages 1–17. Springer Berlin Heidelberg, 2009. ISBN 978-3-540-70999-2 978-3-540-92673-3. URL http://link.springer.com/chapter/10.1007/978-3-540-92673-3_0. DOI : 10.1007/978-3-540-92673-3_0. (Cité page 51.)
- Nicola Guarino et others. Formal ontology and information systems. Dans Proceedings of FOIS, volume 98, pages 81–97, 1998. (Cité page 151.)
- Ramanathan Guha, Rob McCool, et Eric Miller. Semantic search. Dans Proceedings of the 12th international conference on World Wide Web, pages 700–709. ACM, 2003. (Cité page 80.)
- Asunción Gómez-Pérez. Knowledge Sharing and Reuse. Dans The Handbook of Applied Expert Systems. Edited by J. Liebowitz. CRC Press LLC, 1998. bibtex : Gomez-Perez1998. (Cité page 64.)
- Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho, et a Gomez-Perez. Ontological Engineering. 2004. ISBN 978-1-85233-551-9. URL [http://delicias.dia.fi.upm.es/wiki/images/a/aa/1._Intro-Sweb.pdf\\$delimiter"026E30F\\$nhhttp://books.google.com/books?id=UjS0N1W7GSEC](http://delicias.dia.fi.upm.es/wiki/images/a/aa/1._Intro-Sweb.pdf$delimiter). (Cité pages 53, 55, 61, 63 et 64.)
- Asunción Gómez-Pérez, Mariano Fernández-López, et A.J DE VINCENTE. Towards a Method to Conceptualize Domain Ontologies. Dans ECAI-96 Workshop on Ontological Engineering, ECAI-96 Workshop Proceedings, 1996. (Cité page 63.)
- Florian Haag, Steffen Lohmann, Stephan Siek, et Thomas Ertl. Visual Querying of Linked Data with QueryVOWL. Dans SumPre-HSWI@ ESWC, 2015. (Cité page 100.)
- Theo Haerder et Andreas Reuter. Principles of Transaction-oriented Database Recovery. ACM Comput. Surv., 15(4) :287–317, Décembre 1983. ISSN 0360-0300. URL <http://doi.acm.org/10.1145/289.291>. (Cité page 38.)

- R. Hecht et S. Jablonski. NoSQL evaluation : A use case oriented survey. Dans 2011 International Conference on Cloud and Service Computing, pages 336–341, Décembre 2011. (Cité page 39.)
- Martin Heidegger. Introduction to metaphysics. Yale University Press, 2014. (Cité page 151.)
- Martin G. Helander, Thomas K. Landauer, et Prasad V. Prabhau, éditeurs. Handbook of Human-Computer Interaction. Elsevier Science Inc., New York, NY, USA, 2nd édition, 1997. ISBN 978-0-444-81862-1. (Cité page 43.)
- Gary G. Hendrix, Earl D. Sacerdoti, Daniel Sagalowicz, et Jonathan Slocum. Developing a Natural Language Interface to Complex Data. ACM Trans. Database Syst., 3(2) :105–147, Juin 1978. ISSN 0362-5915. URL <http://doi.acm.org/10.1145/320251.320253>. (Cité page 43.)
- Matthias Hert, Gerald Reif, et Harald C. Gall. Updating Relational Data via SPARQL/Update. Dans Proceedings of the 2010 EDBT/ICDT Workshops, EDBT '10, pages 24 :1–24 :8, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-990-9. URL <http://doi.acm.org/10.1145/1754239.1754266>. (Cité pages 124 et 125.)
- Matthias Hert, Gerald Reif, et Harald C. Gall. A Comparison of RDB-to-RDF Mapping Languages. Dans Proceedings of the 7th International Conference on Semantic Systems, I-Semantics '11, pages 25–32, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0621-8. URL <http://doi.acm.org/10.1145/2063518.2063522>. (Cité pages 120, 121, 124 et 127.)
- Frederik Hogenboom, Viorel Milea, Flavius Frasinca, et Uzay Kaymak. RDF-GL : A SPARQL-Based Graphical Query Language for RDF. Dans Richard Chbeir, Youakim Badr, Ajith Abraham, et Aboul-Ella Hassanien, éditeurs, Emergent Web Intelligence : Advanced Information Retrieval, pages 87–116. Springer London, London, 2010. ISBN 978-1-84996-074-8. URL http://dx.doi.org/10.1007/978-1-84996-074-8_4. (Cité pages 74 et 75.)
- Ian Horrocks. Ontologies and the semantic web. Communications of the ACM, 51(12) : 58–67, 2008. bibtex : horrocks2008ontologies. (Cité page 77.)
- Vagelis Hristidis et Yannis Papakonstantinou. Discover : Keyword Search in Relational Databases. Dans Proceedings of the 28th International Conference on Very Large Data Bases, VLDB '02, pages 670–681, Hong Kong, China, 2002. VLDB Endowment. URL <http://dl.acm.org/citation.cfm?id=1287369.1287427>. (Cité page 43.)
- Wei Hu et Yuzhong Qu. Discovering Simple Mappings Between Relational Database Schemas and Ontologies. Dans Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, et Philippe Cudré-Mauroux, éditeurs, The Semantic Web : 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings, pages 225–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-76298-0. URL http://dx.doi.org/10.1007/978-3-540-76298-0_17. (Cité pages 83 et 123.)

- IDC-EMC. The Digital Universe of Opportunities. Rapport technique, 2014. URL <http://emc2.co/leadership/digital-universe/2014iview/index.htm>. bibtex : IDC/EMC2014. (Cité page 22.)
- Sameem Abdul Kareem Ismail M. A, Mashkuri Yaakob. Ontology Construction : An Overview. National Convension of Educational Technology, 9th-11th September 2006, Langkawi, Malaysia, 2006. bibtex : IsmailM.AMashkuriYaakob2006. (Cité pages 64 et 65.)
- Grega Jakus, Veljko Milutinovi?, Sanida Omerovi?, et Sa?o Toma?i? Concepts, Ontologies, and Knowledge Representation. SpringerBriefs in Computer Science. Springer New York, New York, NY, 2013. ISBN 978-1-4614-7821-8 978-1-4614-7822-5. URL <http://link.springer.com/10.1007/978-1-4614-7822-5>. (Cité pages 48, 50, 51 et 151.)
- T. J. Jankun-Kelly et Kwan-Liu Ma. MoireGraphs : radial focus+context visualization and interaction for graphs with visual nodes. Dans IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714), pages 59–66, Octobre 2003. (Cité page 71.)
- Robert Jasper, Mike Uschold, et others. A framework for understanding and classifying ontology applications. Dans Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW, volume 99, pages 16–21, 1999. (Cité page 56.)
- John Stark. Product Lifecycle Management Volume 1 : 21st Century Paradigm for Product Realisation. Decision Engineering. Springer International Publishing, Cham, 2015. ISBN 978-3-319-17439-6 978-3-319-17440-2. URL <http://link.springer.com/10.1007/978-3-319-17440-2>. (Cité pages 4, 6, 11 et 32.)
- Marc Joliot, Nicolas Delcroix, Laure Zago, M Vigneau, Fabrice Crivello, G Simon, Laurent Petit, MR Turbelin, Mickaël Naveau, F Lambertson, et others. GINdb : portable database for the storage and processing of human functional brain imaging data. Dans Proceedings of the 16th Annual Meeting of the Organization for Human Brain Mapping Barcelona, Spain, 2010. (Cité page 106.)
- Jun Juan et David H. Jones. Enabling Semantic Access to Enterprise RDB Data, 2007. URL <https://www.w3.org/2007/03/RdfRDB/papers/yuan.html>. (Cité pages 39 et 42.)
- KACTUS. The KACTUS Booklet version 1.0. Esprit Project 8145 KACTUS. Rapport technique, 1996. URL <http://www.swi.psy.uva.nl/prjects/NewKACTUS/Reports.html>. bibtex : kactus :1996. (Cité page 63.)
- Gilles Kassel. Integration of the DOLCE top-level ontology into the OntoSpec methodology. Rapport technique, Octobre 2005. bibtex : Kassel2005b. (Cité page 113.)
- Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, et Eugenia Giannopoulou. Ontology Visualization Methods? a Survey. ACM Comput. Surv., 39(4), Novembre 2007. ISSN 0360-0300. URL <http://doi.acm.org/10.1145/1287620.1287621>. (Cité pages 69 et 71.)

- Esther Kaufmann et Abraham Bernstein. Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases. Web Semantics : Science, Services and Agents on the World Wide Web, 8(4) :377–393, Novembre 2010. ISSN 1570-8268. URL <http://www.sciencedirect.com/science/article/pii/S1570826810000582>. (Cité page 43.)
- Daniel A Keim. Visual Techniques for Exploring Databases. Dans Knowledge Discovery in Databases (KDD'97), 1997. (Cité page 66.)
- Abdel Kader Keita. Conception Coopérative d'Ontologies Pré-Consensuelles : Application au domaine de l'Urbanisme. PhD thesis, L'Institut National des Sciences Appliquées de Lyon, 2007. (Cité page 108.)
- Tanja Keller et Sigmar-Olaf Tergan. Visualizing Knowledge and Information : An Introduction. Dans Sigmar-Olaf Tergan et Tanja Keller, éditeurs, Knowledge and Information Visualization, numéro 3426 dans Lecture Notes in Computer Science, pages 1–23. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26921-2 978-3-540-31962-7. URL http://link.springer.com/chapter/10.1007/11510154_1. DOI : 10.1007/11510154_1. (Cité pages 3, 6, 19, 38, 66, 67 et 139.)
- Christopher S. G. Khoo et Jin-Cheon Na. Semantic relations in information science. Annual Review of Information Science and Technology, 40(1) :157–228, 2006. ISSN 1550-8382. URL <http://dx.doi.org/10.1002/aris.1440400112>. (Cité page 107.)
- Nikolaos Konstantinou, Dimitrios-Emmanuel Spanos, et Nikolas Mitrou. Ontology and Database Mapping : A Survey of Current Implementations and Future Directions. J. Web Eng., 7(1) :1–24, Mars 2008. ISSN 1540-9589. URL <http://dl.acm.org/citation.cfm?id=2011262.2011263>. (Cité page 83.)
- S. Kriglstein et G. Wallner. Knoocks - A Visualization Approach for OWL Lite Ontologies. Dans 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pages 950–955, Février 2010. (Cité page 70.)
- D.M. Lamberti, J.M. Prager, et M. Nappari. Constrained natural language interface for a computer system. Google Patents, Novembre 1994. URL <https://www.google.com/patents/US5369575>. US Patent 5,369,575. (Cité page 43.)
- Jens Lehmann et Johanna Voelker. An introduction to ontology learning. Perspectives on ontology learning. AKA/IOS Press, Heidelberg, pages 9–16, 2014. (Cité page 87.)
- Yuanguai Lei, Victoria Uren, et Enrico Motta. Semsearch : A search engine for the semantic web. Dans International Conference on Knowledge Engineering and Knowledge Management, pages 238–245. Springer, 2006. (Cité page 80.)
- Douglas B. Lenat et R. V. Guha. Building Large Knowledge-Based Systems ; Representation and Inference in the Cyc Project. Addison-Wesley Longman Publishing Co., Inc., Décembre 1989. ISBN 0-201-51752-3. URL <http://dl.acm.org/citation.cfm?id=575523>. (Cité pages 61 et 63.)

- Maurizio Lenzerini. Data integration : a theoretical perspective. page 233. ACM Press, 2002. ISBN 978-1-58113-507-7. URL <http://portal.acm.org/citation.cfm?doid=543613.543644>. (Cité pages 81 et 85.)
- Ling Liu et M. Tamer Zsu. Encyclopedia of Database Systems. Springer Publishing Company, Incorporated, 1st édition, 2009. ISBN 978-0-387-35544-3. (Cité page 51.)
- M. F. Lopez, A. Gomez-Perez, J. P. Sierra, et A. P. Sierra. Building a chemical ontology using Methontology and the Ontology Design Environment. IEEE Intelligent Systems and their Applications, 14(1) :37–46, Janvier 1999. ISSN 1094-7167. (Cité page 64.)
- Y. Lv et Z. M. Ma. Transformation of relational model to RDF model. Dans 2008 IEEE International Conference on Systems, Man and Cybernetics, pages 506–511, Octobre 2008. (Cité pages 78 et 82.)
- Meenakshi Malhotra et T. R. Gopalakrishnan Nair. Evolution of Knowledge Representation and Retrieval Techniques. I.J. Intelligent Systems and Applications, Juillet 2015. URL <http://www.mecs-press.org/ijisa/ijisa-v7-n7/IJISA-V7-N7-3.pdf>. (Cité page 48.)
- Roland Maranzana. 3d data mining for part and information re-use in a PLM context. 2013. (Cité page 134.)
- Francis T. Marchese et Ebad Banissi, éditeurs. Knowledge Visualization Currents, From Text to Art to Culture. Springer London, London, 2013. ISBN 978-1-4471-4302-4 978-1-4471-4303-1. URL <http://link.springer.com/10.1007/978-1-4471-4303-1>. (Cité pages 66 et 94.)
- Rodriguez-Muro Mariano. Stanford’12 Intro to Ontology Based Data Access for RDBMS through query rewriting, 2012. URL <http://www.slideshare.net/marianomx/seminar-obdaintrostanfordbmir>. (Cité page 104.)
- Rahul Matkar et Ajit Parab. Ontology based expert systems? replication of human learning. Dans Thinkquest~2010, pages 43–47. Springer, New Delhi, 2011. URL https://link.springer.com/chapter/10.1007/978-81-8489-989-4_7. DOI : 10.1007/978-81-8489-989-4_7. (Cité page 151.)
- Andrew McAfee, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, et Dominic Barton. Big data. The management revolution. Harvard Bus Rev, 90(10) :61–67, 2012. bibtex : mcafee2012big. (Cité page 15.)
- John McCarthy. What is Artificial Intelligence? 2007. URL <http://www-formal.stanford.edu/jmc/whatisai/>. (Cité page 52.)
- Franck Michel, Johan Montagnat, et Catherine Faron-Zucker. A survey of RDB to RDF translation approaches and tools. Research Report, I3S, Mai 2014. URL <https://hal.archives-ouvertes.fr/hal-00903568>. ISRN I3S/RR 2013-04-FR 24 pages. (Cité pages 119, 120 et 125.)

- Marvin Minsky. A framework for representing knowledge. 1974. (Cité page 49.)
- Yassaman Zand Moghaddam et Joe D Horton. Relational database schema to ontology mapping approaches, 2010. (Cité page 57.)
- Lynn Nadel. Encyclopedia of cognitive science, volume 3. nature publishing group London,, United Kingdom, 2003. (Cité page 47.)
- Robert Neches, Richard E Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, et William R Swartout. Enabling technology for knowledge sharing. AI magazine, 12(3) : 36–56, 1991. ISSN 07384602. bibtex : Neches1991. (Cité page 52.)
- Chokri Ben Necib. Ontology-based semantic query processing in database systems. phdthesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2008. bibtex : necib2008ontology. (Cité page 43.)
- Thomas Paviot. Méthodologie de résolution des problèmes d'interopérabilité dans le domaine du Product Lifecycle Management. phdthesis, Ecole Centrale Paris, Juillet 2010. URL <https://tel.archives-ouvertes.fr/tel-00504680/document>. (Cité pages 24 et 30.)
- Cong Cuong Pham, Nada Matta, Alexandre Durupt, Benoit Eynard, Marianne Allanic, Guillaume Ducellier, Marc Joliot, et Philippe Boutinaud. How to share complex data and knowledge : Application in Bio-Imaging. IFAC-PapersOnLine, 49(12) :1098–1103, 2016a. ISSN 2405-8963. URL <http://www.sciencedirect.com/science/article/pii/S2405896316308497>. (Cité pages 4, 6 et 137.)
- Cong Cuong Pham, Nada Matta, Alexandre Durupt, Benoit Eynard, Marianne Allanic, Guillaume Ducellier, Marc Joliot, et Philippe Boutinaud. Using Ontologies to access complex Data : Applications on Bio-Imaging. Dans proceedings of AI4KM Workshop, IJCAI, page 6, 2016b. (Cité pages 4, 6 et 137.)
- M. Ross Quillian. Semantic Networks. Dans Marvin L. Minsky, éditeur, Semantic Information Processing. MIT Press, 1968. (Cité page 48.)
- Robert Reix, Bernard Fallery, Michel Kalika, et Frantz Rowe. Systèmes d'information et management. Vuibert, 2016. (Cité page 20.)
- N.D. Rishe. Database querying system and method. Google Patents, Septembre 2004. URL <https://www.google.com/patents/US6795825>. US Patent 6,795,825. (Cité page 42.)
- Mariano Rodriguez-Muro, Roman Kontchakov, et Michael Zakharyashev. Ontology-based data access : Ontop of databases. Dans The Semantic Web–ISWC 2013, pages 558–573. Springer, 2013. bibtex : rodriguez2013ontology. (Cité page 43.)
- Catherine Roussey, Francois Pinet, Myoung Ah Kang, et Oscar Corcho. An Introduction to Ontologies and Ontology Engineering. Dans Ontologies in Urban Development Projects,

- pages 9–38. Springer London, London, 2011. ISBN 978-0-85729-724-2. URL <http://dx.doi.org/10.1007/978-0-85729-724-2>. bibtex : Roussey2011. (Cité page 58.)
- Jennifer Rowley. The wisdom hierarchy : representations of the DIKW hierarchy. Journal of Information Science, 33(2) :163–180, Avril 2007. ISSN 0165-5515, 1741-6485. URL <http://jis.sagepub.com/content/33/2/163>. (Cité page 47.)
- Alistair Russell, Paul R. Smart, Dave Braines, et Nigel R. Shadbolt. NITELIGHT : A Graphical Tool for Semantic Query Construction, Avril 2008. URL <http://eprints.soton.ac.uk/264975/>. (Cité pages 75, 76 et 100.)
- Leila Safari et Jon D. Patrick. Restricted natural language based querying of clinical databases. Journal of Biomedical Informatics, 52 :338–353, Décembre 2014. ISSN 1532-0464. URL <http://www.sciencedirect.com/science/article/pii/S1532046414001592>. (Cité page 43.)
- Anuj Saini. Data Warehousing, 2014. URL <http://www.slideshare.net/AnujSaini5/data-warehousing-44050107>. (Cité page 83.)
- Sherif Sark. Supply cloud-level data scalability with NoSQL databases. Rapport technique, Mars 2013. URL <http://www.ibm.com/developerworks/cloud/library/cl-nosqldatabase/index.html>. (Cité page 39.)
- Mike Scaife et Yvonne Rogers. External cognition : how do graphical representations work? International Journal of Human-Computer Studies, 45(2) :185–213, Août 1996. ISSN 1071-5819. URL <http://www.sciencedirect.com/science/article/pii/S1071581996900488>. (Cité page 66.)
- Juan F Sequeda, Syed H Tirmizi, Oscar Corcho, et Daniel P Miranker. Direct mapping SQL databases to the semantic web : A survey. Univeristy of Texas, Department of Computer Sciecnecs Technical Report TR-09-04, 2009. (Cité page 83.)
- Ben Shneiderman. The future of interactive systems and the emergence of direct manipulation. Behaviour & Information Technology, 1(3) :237–256, 1982. URL <http://dx.doi.org/10.1080/01449298208914450>. (Cité page 43.)
- Ben Shneiderman. The Eyes Have It : A Task by Data Type Taxonomy for Information Visualizations. Dans Proceedings of the 1996 IEEE Symposium on Visual Languages, VL '96, pages 336–, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 978-0-8186-7508-9. URL <http://dl.acm.org/citation.cfm?id=832277.834354>. (Cité page 42.)
- Siemens. Teamcenter 10.1 - Query Builder Guide. Rapport technique, 2013. (Cité page 135.)
- Mark Silver, M. Markus, et Cynthia Beath. The Information Technology Interaction Model : A Foundation for the MBA Core Course. Management Information Systems Quarterly, 19 (3), Décembre 1995. URL <http://aisel.aisnet.org/misq/vol19/iss3/5>. (Cité page 20.)

- Paul R. Smart, Alistair Russell, Dave Braines, Yannis Kalfoglou, Jie Bao, et Nigel R. Shadbolt. A Visual Approach to Semantic Query Design Using a Web-Based Graphical Query Designer. Dans Aldo Gangemi et Jérôme Euzenat, éditeurs, Knowledge Engineering : Practice and Patterns : 16th International Conference, EKAW 2008, Acitrezza, Italy, September 29 - October 2, 2008. Proceedings, pages 275–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-87696-0. URL http://dx.doi.org/10.1007/978-3-540-87696-0_25. (Cité pages 74 et 75.)
- John F Sowa. Conceptual graphs : Information processing in mind and machine. Reading, Addison Wesley, 234, 1984. (Cité pages 49 et 50.)
- John F Sowa. Knowledge representation : logical, philosophical, and computational foundations. 1999. (Cité page 94.)
- Ahmet Soylu, Martin Giese, Ernesto Jimenez-Ruiz, Evgeny Kharlamov, Dmitriy Zheleznyakov, et Ian Horrocks. Ontology-based end-user visual query formulation : Why, what, who, how, and which? Universal Access in the Information Society, pages 1–33, Avril 2016. ISSN 1615-5289, 1615-5297. URL <http://link.springer.com/article/10.1007/s10209-016-0465-0>. (Cité pages 42, 45 et 92.)
- Ahmet Soylu, Martin G. Skjæveland, Martin Giese, Ian Horrocks, Ernesto Jimenez-Ruiz, Evgeny Kharlamov, et Dmitriy Zheleznyakov. A Preliminary Approach on Ontology-Based Visual Query Formulation for Big Data. Dans Emmanouel Garoufallou et Jane Greenberg, éditeurs, Metadata and Semantics Research, numéro 390 dans Communications in Computer and Information Science, pages 201–212. Springer International Publishing, Novembre 2013. ISBN 978-3-319-03436-2 978-3-319-03437-9. URL http://link.springer.com/chapter/10.1007/978-3-319-03437-9_21. DOI : 10.1007/978-3-319-03437-9_21. (Cité pages 2, 3, 5, 6, 20, 25 et 27.)
- Dimitrios-Emmanuel Spanos, Periklis Stavrou, et Nikolas Mitrou. Bringing Relational Databases into the Semantic Web : A Survey. Semant. web, 3(2) :169–209, Avril 2012. ISSN 1570-0844. URL <http://dx.doi.org/10.3233/SW-2011-0055>. (Cité pages 39, 42, 82, 83, 84, 85 et 128.)
- Mohamed-Foued Sriti et Philippe Boutinaud. PLMXQuery : Towards a Standard PLM Querying Approach. Dans Louis Rivest, Abdelaziz Bouras, et Borhen Louhichi, éditeurs, Product Lifecycle Management. Towards Knowledge-Rich Enterprises, numéro 388 dans IFIP Advances in Information and Communication Technology, pages 379–388. Springer Berlin Heidelberg, Juillet 2012. ISBN 978-3-642-35757-2 978-3-642-35758-9. URL http://link.springer.com/chapter/10.1007/978-3-642-35758-9_34. DOI : 10.1007/978-3-642-35758-9_34. (Cité pages 137, 139 et 142.)
- S. Staab et R. Studer. Handbook on Ontologies. International Handbooks on Information Systems. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-40834-5. URL <https://books.google.fr/books?id=0EIGz95mM8QC>. (Cité page 55.)

- Steffen Staab, Rudi Studer, Hans-Peter Schnurr, et York Sure. Knowledge Processes and Ontologies. IEEE Intelligent Systems, 16(1) :26–34, Janvier 2001. ISSN 1541-1672. URL <http://dx.doi.org/10.1109/5254.912382>. (Cité page 64.)
- John Stark. Product Lifecycle Management - 21st Century Paradigm for Product Realisation. Numéro 1619-5736 dans Decision Engineering. Springer London, London, 2005. ISBN 978-1-85233-810-7. URL <http://link.springer.com/10.1007/b138157>. (Cité page 31.)
- Rudi Studer, V.Richard Benjamins, et Dieter Fensel. Knowledge engineering : Principles and methods. Data & Knowledge Engineering, 25(1-2) :161–197, 1998. ISSN 0169023X. (Cité pages 3, 6, 53, 62, 93 et 151.)
- Qi Su et J. Widom. Indexing relational database content offline for efficient keyword-based search. Dans 9th International Database Engineering Application Symposium (IDEAS'05), pages 297–306, Juillet 2005. (Cité page 43.)
- Osma Suominen, Kim Viljanen, et Eero Hyvänen. User-Centric Faceted Search for Semantic Portals. Dans Proceedings of the 4th European Conference on The Semantic Web : Research and Applications, ESWC '07, pages 356–370, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-72666-1. URL http://dx.doi.org/10.1007/978-3-540-72667-8_26. (Cité page 45.)
- B. Swartout, R. Patil, K. Knight, et T. Russ. Toward Distributed Use of Large-Scale Ontologies. Dans Ontological Engineering, AAI-97 Spring Symposium Series, pages 138–148, 1997. (Cité pages 53, 54 et 64.)
- Lynda Temal, Michel Dojat, Gilles Kassel, et Bernard Gibaud. Towards an ontology for sharing medical images and regions of interest in neuroimaging. Journal of Biomedical Informatics, 41(5) :766–778, 2008. ISSN 15320464. (Cité page 113.)
- Sergio Terzi, Abdelaziz Bouras, Debashi Dutta, Marco Garetti, et Dimitris Kiritsis. Product lifecycle management-from its history to its new role. International Journal of Product Lifecycle Management, 4(4) :360–389, 2010. (Cité pages 4, 6, 21, 29, 33 et 133.)
- Thanh Tran, Philipp Cimiano, Sebastian Rudolph, et Rudi Studer. Ontology-based interpretation of keywords for semantic search. Dans The Semantic Web, pages 523–536. Springer, 2007. URL http://link.springer.com/chapter/10.1007/978-3-540-76298-0_38. (Cité page 80.)
- Van Long Tran. Visualizing High-density Clusters in Multidimensional Data. PhD thesis, Jacobs University Bremen, Janvier 2010. URL <https://opus.jacobs-university.de/frontdoor/index/index/docId/292>. (Cité page 66.)
- Christophe Tricot. Cartographie sémantique : des connaissances à la carte. PhD thesis, 2006. URL <http://www.theses.fr/2006CHAMS032>. (Cité page 93.)

- Christophe Tricot et Christophe Roche. Visualisation of ontology : a focus and context approach. Dans International Conference on Multidisciplinary Information Sciences and Technologies, volume 282, pages 287–288, 2006. (Cité page 72.)
- Daniel Tunkelang. Faceted Search. Synthesis Lectures on Information Concepts, Retrieval, and Services, 1(1) :1–80, Janvier 2009. ISSN 1947-945X, 1947-9468. URL <http://www.morganclaypool.com/doi/abs/10.2200/S00190ED1V01Y200904ICR005>. (Cité pages 44 et 45.)
- Vernon Turner, John F Gantz, David Reinsel, et Stephen Minton. The digital universe of opportunities : Rich data and the increasing value of the internet of things. IDC Analyze the Future, 2014. (Cité page 15.)
- Angela Tursi. Ontology-Based approach for Product-Driven interoperability of enterprise production systems. phdthesis, Université Henri Poincaré - Nancy I; Politecnico di Bari, Novembre 2009. URL <https://tel.archives-ouvertes.fr/tel-00432303/document>. (Cité page 55.)
- Michael Uschold et Michael Gruninger. Ontologies and semantics for seamless connectivity. ACM SIGMod Record, 33(4) :58–64, 2004. (Cité page 60.)
- Mike Uschold. Building Ontologies : Towards a Unified Methodology. Dans In 16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems, pages 16–18, 1996. (Cité page 61.)
- Mike Uschold et Robert Jasper. A Framework for Understanding and Classifying Ontology Applications. 1999. (Cité page 54.)
- Mike Uschold et Martin King. Towards a Methodology for Building Ontologies. Dans In Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95, 1995. (Cité page 63.)
- Raphael Volz, Siegfried Handschuh, Steffen Staab, Ljiljana Stojanovic, et Nenad Stojanovic. Unveiling the hidden bride : deep annotation for mapping and migrating legacy data to the Semantic Web. Web Semantics : Science, Services and Agents on the World Wide Web, 1(2) :187–206, Février 2004. ISSN 1570-8268. URL <http://www.sciencedirect.com/science/article/pii/S1570826803000295>. (Cité pages 83 et 85.)
- Holger Wache, Thomas Voegelé, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, et Sebastian Hübner. Ontology-based integration of information-a survey of existing approaches. Dans IJCAI-01 workshop : ontologies and information sharing, volume 2001, pages 108–117. Citeseer, 2001. (Cité pages 63 et 84.)
- Ulli Waltinger, Dan Tecuci, Mihaela Olteanu, Vlad Mocanu, et Sean Sullivan. Natural Language Access to Enterprise Data. AI Magazine, 35 :38–52, 2014. (Cité page 43.)
- Taowei David Wang et Bijan Parsia. CropCircles : topology sensitive visualization of OWL class hierarchies. Dans International Semantic Web Conference, pages 695–708. Springer, 2006. (Cité page 70.)

- Colin Ware. Information Visualization : Perception for Design, 2nd edition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. ISBN 978-1-55860-819-1. (Cité page 66.)
- Donald A. Waterman. A Guide to Expert Systems. Addison-Wesley Longman Publishing Co., Inc., 1986. bibtex : Waterman1986. (Cité page 64.)
- Shuxin Zhao et Elizabeth Chang. From Database to Semantic Web Ontology : An Overview. Dans Robert Meersman, Zahir Tari, et Pilar Herrero, éditeurs, On the Move to Meaningful Internet Systems 2007 : OTM 2007 Workshops, numéro 4806 dans Lecture Notes in Computer Science, pages 1205–1214. Springer Berlin Heidelberg, Novembre 2007. ISBN 978-3-540-76889-0 978-3-540-76890-6. URL http://link.springer.com/chapter/10.1007/978-3-540-76890-6_48. DOI : 10.1007/978-3-540-76890-6_48. (Cité pages 81 et 86.)
- Jiwei Zhong, Haiping Zhu, Jianming Li, et Yong Yu. Conceptual graph matching for semantic search. Dans International Conference on Conceptual Structures, pages 92–106. Springer, 2002. (Cité page 80.)
- Linhua Zhou, Huajun Chen, Yu Zhang, et Chunying Zhou. A Semantic Mapping System for Bridging the Gap between Relational Database and Semantic Web. Dans AAAI Spring Symposium : Semantic Scientific Knowledge Integration, volume 122, 2008. (Cité page 83.)
- Leyla Zhuhadar, Olfa Nasraoui, Robert Wyatt, et Rong Yang. Visual knowledge representation of conceptual semantic networks. Social Network Analysis and Mining, 1(3) :219–229, Novembre 2010. ISSN 1869-5450, 1869-5469. URL <http://link.springer.com/article/10.1007/s13278-010-0008-2>. (Cité page 66.)