Accès sémantique aux données massives et hétérogènes en santé

Romain Lelong

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de l'Université de Rouen Normandie

Accès sémantique aux données massives et hétérogènes en santé

Présentée et soutenue par
Romain LELONG

Thèse soutenue publiquement le 17 juin 2019 devant le jury composé de		
M. Patrice BELLOT	PU, Université Aix-Marseille	Rapporteur
M. Stéfan J. DARMONI	PUPH, LIMICS U1142	Directeur de thèse
Mme. Catherine DUCLOS	PUPH, LIMICS U1142	Rapporteuse
M. Laurent MOUCHARD	MCU, HDR, Université Rouen Normandie	Examinateur
Mme. Aurélie NÉVÉOL	CR, HDR, LIMSI CNRS UPR3251	Examinatrice
Mme. Lina F. SOUALMIA	MCU, HDR, Université Rouen Normandie	Codirectrice de thèse

Thèse dirigée par :
Pr. Stéfan J. DARMONI, LIMICS U1142, Sorbonne Université
Dr. Lina F. SOUALMIA, LITIS EA4108, Université de Rouen Normandie

UNIVERSITÉ DE ROUEN NORMANDIE

MIIS

litis

Remerciements

Je souhaite tout d'abord remercier mon directeur de thèse, le Pr. Stéfan J. DARMONI, pour m'avoir encouragé à réaliser cette thèse et avoir répondu présent lorsque je l'ai sollicité. Je le remercie également pour le dynamisme et l'ambiance atypique, et vraisemblablement endémique, qu'il insuffle à l'ensemble de notre équipe de recherche.

Je souhaite également remercier ma codirectrice de thèse, le Dr. Lina F. SOUALMIA, pour son encadrement scientifique rigoureux, sa bienveillance, sa patience et ses relectures attentives non seulement de ce mémoire mais également des papiers scientifiques que j'ai été amené à rédiger.

Je tiens également à adresser de vifs remerciements à Badisse DAHAMNA avec qui les nombreux échanges scientifiques ont toujours été d'une grande aide et dont la pertinence et la philanthropie naturelle sont parvenues à susciter ma détermination à des moments clés de mon parcours.

Après près de dix ans passés au sein du D2IM (anciennement équipe CiSMeF), j'exprime tous mes remerciements à l'ensemble de ses membres et ex membres. Les travaux réalisés dans le cadre de cette thèse s'appuient sur de nombreux acquis de l'équipe et sont avant tout le résultat d'une collaboration et d'une transmission d'idées entre des personnes d'horizons et d'aptitudes divers. Je suis heureux d'avoir eu l'opportunité de travailler avec ces personnes et d'avoir eu la chance de m'appuyer sur leurs travaux, leurs expériences, leurs conseils et leurs contributions : Kévin BILLEY, Chloé CABOT, Emeric DYNOMANT, Nicolas GRIFFON, Julien GROSJEAN, Gaétan KERDELHUÉ, Ivan KERGOURLAY, Émeline LEJEUNE, Jean-Philippe LEROY, Catherine LETORD, Philippe MASSARI, Tayeb MERABTI, Adila MERABTI, Quentin POUSSIER, Matthieu SCHUERS, Saoussen SAKJI, Benoit THIRION, Sandrine VOURIOT.

Je suis également reconnaissant envers la famille BOTTÉ pour son soutien. J'adresse tout particulièrement mes remerciements à Cécile BOTTÉ pour son investissement organisé et minutieux dans les corrections de français de ce mémoire. L'ampleur de la tâche était pour le moins imposante et son aide suscite toute ma gratitude.

Un grand merci également à ma famille qui, malgré mon laconisme pathologique au sujet de ma thèse, a toujours su me soutenir et me conforter sur la bonne voie.

Enfin, je ne saurai terminer ces remerciements sans m'adresser à ma compagne, Clémentine FOURNIER, qui a été soumise à rudes épreuves ces dernières années. Son soutien quotidien, sa compréhension et ses attentions destinées à favoriser la réalisation de mes objectifs ont joué un rôle primordial que je tiens à souligner.

Résumé

Les données cliniques sont produites par différents professionnels de santé, dans divers lieux et sous diverses formes dans le cadre de la pratique de la médecine. Elles présentent par conséquent une hétérogénéité à la fois au niveau de leur nature et de leur structure mais également une volumétrie particulièrement importante et qualifiable de massive. Le travail réalisé dans le cadre de cette thèse s'attache à proposer une méthode de recherche d'information efficace au sein de ce type de données complexes et massives.

L'accès aux données cliniques se heurte en premier lieu à la nécessité de modéliser l'information clinique. Ceci peut notamment être réalisé au sein du dossier patient informatisé ou, dans une plus large mesure, au sein d'entrepôts de données. Je propose dans ce mémoire une preuve de concept d'un moteur de recherche permettant d'accéder à l'information contenue au sein de l'entrepôt de données de santé sémantique du Centre Hospitalier Universitaire de Rouen. Grâce à un modèle de données générique, cet entrepôt adopte une vision de l'information assimilable à un graphe de données rendant possible la modélisation de cette information tout en préservant sa complexité conceptuelle. Afin de fournir des fonctionnalités de recherche adaptées à cette représentation générique, un langage de requêtes permettant l'accès à l'information clinique par le biais des diverses entités qui la composent a été développé et implémenté dans le cadre de cette thèse.

En second lieu, la massivité des données cliniques constitue un défi technique majeur entravant la mise en œuvre d'une recherche d'information efficace. L'implémentation initiale de la preuve de concept sur un système de gestion de base de données relationnel a permis d'objectiver les limites de ces derniers en terme de performances. Une migration vers un système NoSQL orienté clé-valeur a été réalisée. Bien qu'offrant de bonnes performances d'accès atomique aux données, cette migration a également nécessité des développements annexes et la définition d'une architecture matérielle et applicative propice à la mise en œuvre des fonctionnalités de recherche et d'accès aux données.

Enfin, l'apport de ce travail dans le contexte plus général de l'entrepôt de données de santé sémantique du CHU de Rouen a été évalué. La preuve de concept proposée dans ce travail a ainsi été exploitée pour accéder aux descriptions sémantiques afin de répondre à des critères d'inclusion et d'exclusion de patients dans des études cliniques. Dans cette évaluation, une réponse totale ou partielle a pu être apportée à 72,97% des critères. De plus, la généricité de l'outil a également permis de l'exploiter dans d'autres contextes tels que la recherche d'information documentaire et bibliographique en santé.

Abstract

Clinical data are produced as part of the practice of medicine by different health professionals, in several places and in various formats. They therefore present an heterogeneity both in terms of their nature and structure and are furthermore of a particularly large volume, which make them considered as Big Data. The work carried out in this thesis aims at proposing an effective information retrieval method within the context of this type of complex and massive data.

First, the access to clinical data constrained by the need to model clinical information. This can be done within Electronic Health Records and, in a larger extent, within data Warehouses. In this thesis, I proposed a proof of concept of a search engine allowing the access to the information contained in the Semantic Health Data Warehouse of the Rouen University Hospital. A generic data model allows this data warehouse to view information as a graph of data, thus enabling to model the information while preserving its conceptual complexity. In order to provide search functionalities adapted to this generic representation of data, a query language allowing access to clinical information through the various entities of which it is composed has been developed and implemented as a part of this thesis's work.

Second, the massiveness of clinical data is also a major technical challenge that hinders the implementation of an efficient information retrieval. The initial implementation of the proof of concept highlighted the limits of a relational database management systems when used in the context of clinical data. A migration to a NoSQL key-value store has been then completed. Although offering good atomic data access performance, this migration nevertheless required additional developments and the design of a suitable hardware and applicative architecture to provide advanced search functionalities.

Finally, the contribution of this work within the general context of the Semantic Health Data Warehouse of the Rouen University Hospital was evaluated. The proof of concept proposed in this work was used to access semantic descriptions of information in order to meet the criteria for including and excluding patients in clinical studies. In this evaluation, a total or partial response is given to 72.97% of the criteria. In addition, the genericity of the tool has also made it possible to use it in other contexts such as documentary and bibliographic information retrieval in health.

Table des matières

Remerciements **3**

Résumé **5**

Abstract **7**

Table des matières **9**

Liste des figures **13**

Liste des tables **17**

Index des abréviations **19**

1 Introduction générale **25**
1.1 Introduction . . . 26
1.2 Objectifs des travaux de recherche . . . 29
1.3 Contexte des travaux . . . 31
1.4 Quelques réalisations . . . 32
1.4.1 Health Terminology/Ontology Portal (HeTOP) . . . 32
1.4.2 L'Extracteur de Concepts Multi-Terminologiques (ECMT) . . . 33
1.5 Les projets de recherche . . . 35
1.5.1 Le projet Retrieval And Visualization in ELectronic health records . . . 35
1.5.2 Le projet Saisie Informatique FAcile des DOnnées médicales . . . 36
1.6 Les données de santé au CHU de Rouen . . . 38
1.6.1 Le Dossier Patient Informatisé du CHU de Rouen . . . 38
1.6.2 L'Entrepôt de Données de Santé Sémantique (EDSS) . . . 38
1.7 Organisation du manuscrit . . . 41

2 Les données cliniques : nature, structure et enjeux **43**
2.1 Le Dossier Patient Informatisé (DPI) . . . 44
2.1.1 Les enjeux . . . 45
2.1.2 Les défis . . . 46
2.1.3 Les standards de représentation existants . . . 47
2.1.4 La structure des données du DPI . . . 51
2.2 Les Entrepôts de Données de Santé (EDS) . . . 56
2.2.1 La problématique des données massives (Big Data) . . . 56
2.2.2 Les objectifs de l'EDS . . . 57
2.2.3 Panorama des EDSs existants . . . 58

3 La recherche d'information en Santé **69**
3.1 Les fondements de la recherche d'information . . . 70
3.1.1 Le contexte . . . 70
3.1.2 Le rôle . . . 71
3.1.3 Le principe . . . 71
3.2 Les modèles de recherche d'information classiques . . . 74

3.2.1 Le modèle Booléen . . . 74
3.2.2 Le modèle Vectoriel . . . 76
3.2.3 Le modèle Probabiliste . . . 79
3.3 Le contexte de la santé . . . 82
3.3.1 La recherche d'information textuelle en Santé . . . 83
3.3.2 La recherche d'information au sein de données cliniques . . . 86

4 Les méthodes de stockage de graphes de données 89
4.1 Le Web de données . . . 91
4.1.1 Le Resource Description Framework . . . 91
4.1.2 Ontologies . . . 92
4.1.3 SPARQL Protocol And RDF Query Language . . . 94
4.1.4 Synthèse . . . 96
4.2 Les bases de données alternatives . . . 97
4.2.1 Le paysage des bases de données alternatives . . . 97
4.2.2 À chaque problématique son changement de paradigme . . . 100
4.2.3 Synthèse . . . 109

5 Modélisation et intégration des données de l'EDSS 111
5.1 À l'origine, un SGBDR . . . 113
5.1.1 Notre modèle générique . . . 113
5.1.2 Différents corpus . . . 117
5.1.3 La course « à la perf' » . . . 118
5.2 Le virage du NoSQL . . . 120
5.2.1 L'In Memory Data Grid qualifié . . . 120
5.2.2 Le changement de paradigme en action . . . 120
5.2.3 La recherche d'information . . . 123

6 Le langage de requête $\mathcal{L}_{\text{  }}$ 127
6.1 Description de la syntaxe . . . 130
6.1.1 Clause entité . . . 131
6.1.2 Les types de données . . . 132
6.1.3 Contraintes d'attributs . . . 133
6.1.4 Contraintes sémantiques . . . 138
6.1.5 Propriétés algébriques du langage . . . 146
6.2 Rappels sur les langages formels . . . 152
6.2.1 Alphabet et mots . . . 152
6.2.2 Langages . . . 154
6.2.3 Grammaire . . . 155
6.3 La grammaire $\mathcal{G}_{\text{  }}$ . . . 158
6.3.1 Le grammaire réduite $\mathcal{G}_{\text{  }}^{\text{}}$ . . . 158
6.3.2 $\mathcal{G}_{\text{  }}$, l'intégrale . . . 162
6.4 Implémentation . . . 165
6.4.1 Le générateur JavaCC™ . . . 165
6.4.2 Exploitation du parseur . . . 166

7 La recherche d'information par la pratique 169
7.1 Modes de requêtage . . . 171
7.1.1 Le langage $\mathcal{L}_{\text{DocCiSMeF}}$ . . . 173
7.1.2 Le langage $\mathcal{L}_{\mathbb{B}}$ . . . 176
7.1.3 Le Langage Naturel . . . 178
7.1.4 Le langage $\mathcal{L}_{\text{  }}$ . . . 179
7.2 Logique interne . . . 187
7.2.1 Structure arborescente . . . 187
7.2.2 Le Semantic Search Engine SQL (SSE_{SQL}) . . . 190

7.2.3 Le Semantic Search Engine NoSQL (SSE_{NoSQL}) 194

8 Résultats **199**
8.1 Accès Sémantique à l'Information de Santé (**ASIS**) 200
8.1.1 Première étape : la définition des contraintes 200
8.1.2 Deuxième étape : la constitution d'une requête Booléenne 202
8.1.3 Troisième étape : le choix du type de donnée de sortie 203
8.2 Évaluation 205
8.2.1 La recherche d'information au sein des textes Cliniques 205
8.2.2 Méthodologie 208
8.2.3 Résultats de l'évaluation 212

Conclusions et perspectives **221**

Bibliographie **225**

A Le thésaurus Le thésaurus Medical Subject Headings (MeSH) **243**

B Étude cliniques **245**
B.1 Étude clinique n° 1 245
B.2 Étude clinique n° 2 246
B.3 Étude clinique n° 3 246
B.4 Étude clinique n° 4 247
B.5 Étude clinique n° 5 248

C Annotation Sémantique des Documents Médicaux **251**

D Rappels sur les langages formels **253**
D.1 Alphabet et mots 254
D.2 Monoïde libre 257
D.3 Langages 260
D.4 Grammaire 262

Index des systèmes d'organisation des connaissances **269**

Liste des figures

1.1 Contexte des travaux réalisés. 30
1.2 Processus général de l'**E**xtracteur de **C**oncepts **M**ulti- **T**erminologiques (ECMT). Le texte fourni à l'ECMT ainsi que les différents libellés des concepts d'**H**ealth **T**erminology/**O**ntology **P**ortal (HeTOP) sont normalisés, tokénisés et racinisés. L'algorithme de sac recherche les correspondances entre les mots du texte et des concepts issus du portail HeTOP. 34
1.3 Architecture globale de l'**E**ntrepôt de **D**onnées de **S**anté **S**émantique (EDSS) du **C**entre **H**ospitalier **U**niversitaire **de Rouen** – Hôpital Charles Nicolle (CHU de Rouen) 39

2.2 Répartition des données du **D**ossier **P**atient **I**nformatisé (DPI) en quatre catégories. . . 52
2.5 Outil de requêtage de démonstration d'i2b2 (**I**nformatics for **I**ntegrating **B**iology and the **B**edside (i2b2) Workbench Query tool). Les patients ici recherchés sont les femmes ayant fait l'objet d'un diagnostic quelconque de la catégorie des maladies vasculaires artérielles. Les contraintes `Female` et `Arterial vascular disease` ont été glisser–déposer depuis le champ `Navigate Terms` vers les blocs correspondants du champ `Query Tool`. 61
2.6 Copie d'écran de l'outil de création de cohortes de patients basé sur **S**tanford **T**ranslational **R**esearch **I**ntegrated **D**atabase **E**nvironment (STRIDE) [1] 63

3.1 Pyramide DIKW (Data, Information, Knowledge, Wisdom) [2] 70
3.2 Processus de recherche d'information classique. 72
3.4 Représentation vectorielle d'un document e_i et d'une requête q dans un espace à trois dimensions (i.e. trois termes indexants) t_1, t_2 et t_3 77
3.5 Triade de la Médecine Fondée sur le Preuves (MFP) 82
3.6 Différence structurelle entre l'information documentaire et l'information clinique 87

4.1 Illustration de triplets **R**esource **D**escription **F**ramework (RDF). L'information représentée par ces triplets est « *Le modèle* RDF *est développé par la* ***W****orld* ***W****ide* ***W****eb* ***C****onsortium (*W3C*) dont le nom est* ***W****orld* ***W****ide* ***W****eb* ***C****onsortium* ». Cette information est représentée à l'aide de deux arcs et trois nœuds. Conformément à la convention, les nœuds de ressource/entité sont représentés sous forme d'ellipses et les donnée sous forme de rectangles. 92
4.2 Expréssivité des langages ontologiques Web Ontology Language (Seconde version, 2009) (OWL2) et **R**esource **D**escription **F**ramework **S**chema (RDFS) [3] 94
4.3 Protocole **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage (SPARQL) repose sur une architecture client serveur. La requête SPARQL est encapsulée dans une requête **H**ypertext **T**ransfer **P**rotocol (HTTP) transmise au Triplestore. Ce dernier peut alors renvoyer sa réponse sous différents formats. 95
4.4 Exemple simple d'une requête SPARQL basée sur le graphe RDF de la Figure 4.1. Cette requête permet de retourner les **U**niform **R**esource **I**dentifiers (URIs) des standards développés par le W3C. 95
4.6 Illustration du modèle de représentation des **S**ystèmes de **G**estion de **B**ase de **D**onnées (SGBDs) **N**ot **o**nly **SQL**s (NoSQLs) orientés clés-valeur. 101
4.7 Illustration du modèle de représentation des SGBDs NoSQLs orientés document. 102
4.8 Illustration du modèle de représentation des SGBDs NoSQLs orientés colonne. 104
4.9 Illustration du modèle de représentation des SGBDs NoSQLs orientés Graphe. 105

4.10 Triangle CAP illustrant de manière non exhaustive la politique de gestion de la concurrence de plusieurs SGBDs. 108

5.1 Modélisation relationnelle du **D**épartement d'**I**nformatique et d'**I**nformation **M**édicales (D2IM) et modélisation relationnelle classique. 114

5.2 Modèle logique de donnée du **S**ystème d'**I**nformation (SI). Les clés primaires sont soulignées (e.g. CLE_PRIMAIRE), les clés étrangères sont identifiées par l'annotation [FK], les champs textuels par **A**, les champs numériques par et les champs de type date par . 115

5.3 Principales données patient inclus dans le $_{2\ 000}$. 117

6.1 Positionnement et rôle du langage de requête dans la chaîne d'interrogation des données de l'EDSS. Les deux preuves de concept exploitant respectivement une base de données relationnelle et une base de données NoSQL Infinispan sont représentées. 130

6.2 Extrait du **M**odèle **C**onceptuel de **D**onnées (MCD) représenté sous forme d'un graphe de données orienté, étiqueté et attribué. Dans ce graphe, une simplification d'une partie des données de l'EDSS est représentée conceptuellement. L'entité patient est liée à des séjours au sein desquels peuvent être pratiqués des examens biologiques et des actes médicaux. Les actes médicaux et les séjours peuvent être effectués dans des unités médicales distinctes. Les actes sont rattachés à des codes de la **C**lassification **C**ommune des **A**ctes **M**édicaux (CCAM). 131

6.3 Illustration du rôle d'un analyseur lexical et d'un parseur (i.e. un analyseur syntaxique) dans le traitement d'une chaîne de caractères en entrée. La chaîne de caractères traitée est la $\mathcal{L}_{\text{}}$–requête `patient(sexe="F")`. L'analyseur syntaxique traite cette requête caractère par caractère afin d'identifier une séquence de tokens qui est ensuite analysée par le parseur à l'aide des règles de production de la grammaire $\mathcal{G}_{\text{}}$ pour en construire un objet structuré. 166

7.1 Organigramme de programmation représentant le processus de gestion d'une requête donnée en entrée du moteur de recherche. Les cinq modes d'interrogation possibles (viz. $\mathcal{L}_{\text{}}$–requête, $\mathcal{L}_{\text{Doc'CiSMeF}}$–requête, $\mathcal{L}_{\mathbb{B}}$–requête, $\mathcal{L}_{\#}$–requête et requête en langage naturel) sont représentés. Le format d'une requête est identifié afin de pouvoir se ramener de manière systématique à une $\mathcal{L}_{\text{}}$–requête. 172

7.2 Représentation des différents modes d'interrogation du **S**emantic **S**earch **E**ngine **NoSQL** (SSE_{NoSQL}) en fonction de leur complexité d'utilisation pour un utilisateur et du contexte de **R**echerche d'**I**nformation (RI) pour lequel ils sont adaptés. 173

7.3 Illustration « boîte noire » du convertisseur $\mathcal{L}_{\text{Doc'CiSMeF}} \rightarrow \mathcal{L}_{\text{}}$. La même requête `2018.an` est passée en entrée de Moteur de Recherche **Doc**umentaire basé sur le CiSMeF (Doc'CiSMeF) et **Li**ttérature **S**cientifique en **Sa**nté (LiSSa). Le convertisseur $\mathcal{L}_{\text{Doc'CiSMeF}} \rightarrow \mathcal{L}_{\text{}}$ est appelé avec la requête d'une part et le type d'entité attendu en sortie d'autre part (i.e. `DOC` pour Doc'CiSMeF et `NLM` pour LiSSa). La $\mathcal{L}_{\text{}}$–requête obtenue en sortie est alors différente. 175

7.4 Processus de transformation d'une $\mathcal{L}_{\mathbb{B}}$–requête en $\mathcal{L}_{\text{Doc'CiSMeF}}$–requête. La requête `guide de bonnes pratiques ET asthme ET (enfant OU nourrisson)` est d'abord transformée en un objet java à l'aide du $\text{parser}_{\mathbb{B}}$. Les mots clés sont analysés par le convertisseur $\mathcal{L}_{\mathbb{B}} \rightarrow \mathcal{L}_{\text{Doc'CiSMeF}}$ qui permet, à l'aide de l'annotateur sémantique ECMT de construire, pour chacun, une sous-$\mathcal{L}_{\text{Doc'CiSMeF}}$–requête. 177

7.5 Représentation graphique des différentes phases de réécriture de la requête « *Analyse biologique de polynucléaires neutrophiles supérieure à la normale du patient 71* » effectuée par le tagger. Chaque itération (hormis la première) consomme les tags identifiés par la ou les itérations précédentes afin d'identifier de nouveaux patrons plus complexes. . . . 183

7.6 Représentation de la génération récursive de la $\mathcal{L}_{\text{}}$–requête correspondant à la requête « *Analyse biologique de polynucléaires neutrophiles supérieure à la normale du patient 71* ». 184

7.7 Représentation de la structure arborescente de la $\mathcal{L}_{\text{arbre}}$–requête de l'exemple 32 utilisé en interne du moteur de recherche SSE_{NoSQL} et **S**emantic **S**earch **E**ngine **SQL** (SSE_{SQL}) pour représenter et exécuter cette dernière. Bien que contenus au sein des nœuds d'entité, les contraintes de jointure et de chemins ne sont pas représentés ici pour plus de lisibilité. 189
7.8 Illustration de l'utilisation et du rôle des trois outils mis à disposition par le SI NoSQL du **D2IM** dans le processus de l'exécution d'un $\mathcal{L}_{\text{arbre}}$–arbre. Le $\mathcal{L}_{\text{arbre}}$–arbre en question est repris de l'exemple 32. Le cache d'objet permet l'exécution des contraintes d'attribut basées sur l'identifiant d'une entité, les caches de jointure permettent l'exécution des contraintes sémantiques et les index inversés Apache **Lucene** (Lucene) l'exécution des contraintes d'attribut dans leurs ensembles. 196

8.1 Étape n° 1 de l'interface proposée par **A**ccès **S**émantique à l'**I**nformation de **S**anté (**ASIS**) permettant de définir des contraintes relative à différentes entités et différentes métadonnées. 200
8.2 Saisie avec auto-complétion de la valeur d'une contrainte de type *Médicaments*. 201
8.3 Options d'affinage d'une contrainte de type *Diagnostic* donnant la possibilité d'inclure les descendants du concept sélectionné, de rechercher ce concept au sein des comptes-rendus en plus des diagnostics et de préciser le type de diagnostic (principal ou relié). 201
8.4 Options d'affinage d'une contrainte de type *Analyse Biologique* donnant la possibilité de contraindre la valeur de celle-ci et de considérer également les concepts fils du concept d'analyse sélectionnée. 202
8.5 Composant graphique éditable permettant la composition d'une contrainte Booléenne globale à partir des contraintes unitaires définies à l'aide du formulaire de l'étape n° 1. . 202
8.6 Gestion d'un squelette de contrainte Booléenne globale depuis le formulaire de l'étape n° 1. 203
8.7 Formulaire permettant de choisir le ou les différents types d'entité désiré(s) en sortie de l'application. 203
8.8 Copie d'écran des résultats affichés par **ASIS**. 204
8.9 Annotation sémantique des documents médicaux du CHU de Rouen et post-filtrage manuel des annotations. 207
8.10 Représentation globale du processus de construction d'une $\mathcal{L}_{\text{arbre}}$–requête pour un critère d'étude clinique quelconque. 209
8.11 Pourcentages des critères (critères Non-Applicables exclus) par classe d'utilité de l'outil : outil de constitution de cohorte, outil de pre-screening et outil d'exploration d'information. 213
8.12 Illustration graphique de la répartition des sources d'information parmi les critères d'inclusion et d'exclusion. 214
8.13 Niveaux de prise en charge moyens des critères ciblant une seule source d'information, deux sources d'information, trois sources d'information ou une combinaison de sources d'information de manière générale. 215
8.14 Mise en correspondance des sources d'information ciblées et des limitations observées pour chaque niveau de prise en charge. 216
8.15 Affinage itératif des stratégies de recherches rendu possible par **ASIS** et la SSE_{NoSQL} . . 218
8.16 Mise en évidence des capacités de pré-filtrage du système dans le cadre de la recherche du critère « *insuffisance cardiaque sévère (classe III ou IV, NYHA)* » 219

Liste des tables

1.1 Cas d'usages du projet **R**etrieval **A**nd **V**isualization in **EL**ectronic health records (RAVEL [ANR–11–TECS–012]) et utilité/rôle potentiel de la RI pour chacun d'eux 36
1.2 Volumétrie de l'EDSS au 21 mars 2019 40

2.1 Principales catégories de données stockées dans l'**E**ntrepôt de **D**onnées de **S**anté (EDS) de STRIDE. [1] 62
2.2 Volumes de données gérés par **E**ntrepôt de données biomédicales de l'**HOP**ital (eHOP) au **C**entre **H**ospitalier **U**niversitaire (CHU) de Rennes. [4] 64

3.1 Variantes des mesures $TF_{i,j}$ et IDF_i 79

5.1 Ensemble des cas d'usage ayant servi de base à la modélisation et au développement du SSE_{SQL} 119
5.2 Correspondance entre les tables du modèle physique de données générique et les Classes Java™ (Java™) qui en permettent leurs représentations au niveau applicatif. 120
5.3 Liste des caches de données du SI NoSQL du D2IM. Pour chaque $Map : Clé \rightarrow Valeur$: les rôles de la map, de la clé et de la valeur sont explicités. 122
5.4 Liste des saches système du SI NoSQL du D2IM. 123
5.5 Les deux maps de jointure utilisées dans le cadre des tâches de RI afin de parcourir les relations classiques et les relations d'indexations. L'utilité globale de chaque map et les rôles pris par la clé et la valeur sont explicités. 125

6.1 Opérateurs et comparateurs disponibles pour les différents types de données. 133
6.2 Correspondance entre les non-terminaux de $\mathcal{G}_{\text{<}}$ et les éléments syntaxiques qu'ils permettent de représenter au sein du langage de requête $\mathcal{L}_{\text{<}}$. 164

7.1 Fonction syntaxique jouée par chaque variable de la grammaire $\mathcal{G}_{\mathbb{B}}$ dans une $\mathcal{L}_{\mathbb{B}}$–requête. 176
7.2 Nombre de patterns (expression régulière) manuellement écrites et disponibles pour chacune des cinq itérations du tagger. Pas de tagging de concept (utilisation de l'ECMT). . 185
7.3 Requêtes et types de requêtes en langage naturel issus des cas d'usages du projet RAVEL [ANR–11–TECS–012]. Pour chaque requête le temps d'exécution moyen t de ces dernières sur le SSE_{SQL} est indiqué. 185
7.4 Correspondance entre les éléments de syntaxe des $\mathcal{L}_{\text{<}}$–requêtes et les éléments du modèle de données du D2IM permettant leur exécution. 190

8.1 Top 10 des annotations les plus fréquente avant filtrage 208
8.2 Nombre n et pourcentage p de critères d'inclusion et d'exclusion en fonction de leurs niveaux de prise en charge par le système. Pour chaque pourcentage p, il est également donné l'intervalle de confiance I_c à 95% de ce dernier. 212
8.3 Nombre de critères pour chaque niveau de prise en charge en fonction de la combinaison de source d'information nécessaire pour établir sa stratégie de recherche. 213
8.4 Taux t et pourcentage p d'implication de chaque source d'information dans les stratégies de recherche des 95 critères de l'étude. 214

C.1 Nombre d'annotations unitaire des documents médicaux du CHU de Rouen par terminologies. Seules les 20 premières terminologies ayant le plus d'annotations unitaires sont présentées. 251
C.2 Couverture terminologique des annotations des documents médicaux du CHU de Rouen. Seules les dix premières terminologies ayant le plus de concepts uniques identifiés sont présentées. 252

Index des abréviations

$\mathcal{A}Box$ ensemble de axiomes Assertionnels. 94, 115

ACID **A**tomicité, **C**ohérence, **I**solation et **D**urabilité. 97, 106, 108, 111

AGATE **A**pplication de **G**estion des **A**bsences et des **TE**mps. 38

ANR **A**gence **N**ationale de la **R**echerche. 31, 33, 35

ANSI **A**merican **N**ational **S**tandards **I**nstitute. 49

AOS **A**rchitecture **O**rientée **S**ervice. 59, 62

AP–HP **A**ssistance **P**ublique – **H**ôpitaux de **P**aris. 58, 66

API **A**pplication **P**rogramming **I**nterface. 102, 105, 123, 124, 169, 194

$\mathcal{L}_{\blacktriangleleft}$**–arbre** **Arbre** Binaire d'une $\mathcal{L}_{\blacktriangleleft}$–requête. 15, 187, 190, 191, 195–197

ASIP Santé **A**gence de **S**ystèmes d'**I**nformation **P**artagés de **Santé**. 44, 46, 118

ASIS **A**ccès **S**émantique à l'**I**nformation de **S**anté. 11, 15, 29, 188, 190, 197, 199–201, 204, 205, 208, 209, 211, 215, 218, 221, 222, 224

ATIH **A**gence **T**echnique de l'**I**nformation sur l'**H**ospitalisation. 53

BASE **B**asically **A**vailable, **S**oft–state, **E**ventually Consistent. 108

BLOB **B**inary **L**arge **OB**ject. 101, 104

BNF **B**ackus–**N**aur **F**orm. 157, 166, 263

BSD **B**erkeley **S**oftware **D**istribution. 165

BT–NT **T**erme générique – **T**erme spécifique. 32

caBIG **ca**ncer **B**iomedical **I**nformatics **G**rid. 61

CDM **C**ommon **D**ata **M**odel. 58, 66

CDP **C**Page **D**ossier **P**atient. 38, 117, 205, 206

CEN **C**omité **E**uropéen de **N**ormalisation. 48

CHRU **C**entre **H**ospitalier **R**égional **U**niversitaire. 58

CHU **C**entre **H**ospitalier **U**niversitaire. 17, 58, 64

CHU de Rouen **C**entre **H**ospitalier **U**niversitaire **de Rouen** – Hôpital Charles Nicolle. 13, 15, 18, 25, 26, 28, 29, 31, 32, 37–41, 46, 48, 52, 56, 58, 111, 117, 118, 168, 169, 171, 180, 188, 197, 200, 205, 207, 208, 211, 221, 222, 251, 252

CiSMeF **C**atalogue et **I**ndex des **S**ites **M**édicaux de langue **F**rançaise. 3, 31, 32, 125, 178, 251

CLCC **C**entre de **L**utte **C**ontre le **C**ancer. 64, 65

CLEF **C**onference and **L**abs of the **E**valuation **F**orum. 33

ConSoRe **Con**tinuum **So**ins-**Re**cherche. 58, 64–66

CSV **C**omma-**S**eparated **V**alues. 95

cTAKES **C**linical **T**ext **A**nalysis and **K**nowledge **E**xtraction **S**ystem™. 207

CUI **C**oncept **U**nique **I**dentifier. 32

D2IM **D**épartement d'**I**nformatique et d'**I**nformation **M**édicales. 3, 14, 15, 17, 25–29, 31, 32, 35–39, 41, 89, 109, 113, 114, 116, 121–123, 127, 130, 168, 173, 188, 190, 196, 197, 200, 205, 206, 208, 221–223

DIKW **D**ata, **I**nformation, **K**nowledge and **W**isdom. 70

DIM **D**épartement d'**I**nformation **M**édicale. 65

DMP **D**ossier **M**édical **P**artagé. 44, 45

DocCiSMeF Moteur de Recherche **Doc**umentaire basé sur le **CiSMeF**. 14, 29, 31, 35, 38, 84, 169, 171, 173, 175, 178, 200, 221

DPI **D**ossier **P**atient **I**nformatisé. 13, 26, 29, 31, 38, 41, 43–49, 51, 52, 55–57, 83, 130, 190, 221

DrWarehouse Dr. Warehouse. 58, 65–67, 205, 223

DSL **D**omain–**S**pecific **L**anguage. 123

DW4TR **D**ata **W**arehouse **for** **T**ranslational **R**esearch. 63, 66, 67

EAV **E**ntity–**A**ttribute–**V**alue. 58, 59, 62, 64, 66, 67, 111, 113, 114, 118, 128

ECMT **E**xtracteur de **C**oncepts **M**ulti- **T**erminologiques. 13, 14, 17, 31–34, 39, 118, 171, 177, 178, 182, 185, 206, 207, 221, 224

EDC **E**ntrepôt de **D**onnées Cliniques. 56, 61, 62

EDIFACT **E**lectronic **D**ata **I**nterchange **F**or **A**dministration, **C**ommerce and **T**ransport. 47

EDS **E**ntrepôt de **D**onnées de **S**anté. 17, 26–28, 37, 38, 41, 43, 54–58, 60–64, 66, 67, 74, 83, 87, 89, 90, 96, 97, 101–105, 111, 121, 128, 188, 205, 213, 221, 222

EDSS **E**ntrepôt de **D**onnées de **S**anté **S**émantique. 13, 14, 17, 29, 31, 32, 38–40, 52, 130, 131, 138, 168, 169, 171, 173, 175, 180, 190, 197, 200, 201, 205–208, 210–212, 215, 217, 218, 221–224, 251

EDT **E**nterprise **D**ata **T**rust. 63, 66

ehop **E**ntrepôt de données biomédicales de l'**HOP**ital. 17, 58, 64–66, 222, 223

EHR **E**lectronic **H**ealth **R**ecords. 45

EHRcom **E**lectronic **H**ealth **R**ecord **COM**munication. 48

EM Correspondance **E**xacte. 32

EMERSE **E**lectronic **ME**dical **R**ecord **S**earch **E**ngine. 66, 205, 223

ETEP **E**xtraction **T**ransformation **E**nrichissement **P**ublication. 65

ETL **E**xtract **T**ransform **L**oad. 117

HDF **H**L7 **D**evelopment **F**ramework. 49, 50

HEGP **H**ôpital **E**uropéen **G**eorges–**P**ompidou. 66

HEO **H**orizon **E**xpert **O**rders. 38

HeTOP **H**ealth **T**erminology/**O**ntology **P**ortal. 9, 13, 25, 31–34, 37, 39, 84, 90, 178, 201, 206, 207, 221, 224

HL7 **H**ealth **L**evel **Seven**. 47, 49, 50, 64

HL7 v3 HL7 version 3. 49, 50

HL7–CDA ® **HL7** **C**linical **D**ocument **A**rchitecture. 50

HL7–FHIR ® HL7 **F**ast **H**ealthcare **I**nteroperability **R**esources. 50

HL7–RIM **HL7** **R**eference **I**nformation **M**odel. 50, 62

HPRIM **H**armoniser et **PR**omouvoir l'**I**nformatique **M**édicale. 64

HTTP **H**ypertext **T**ransfer **P**rotocol. 13, 50, 94–96

i2b2 **I**nformatics for **I**ntegrating **B**iology and the **B**edside. 13, 58–62, 66, 128, 222

IMDG **I**n **M**emory **D**ata **G**rid. 109, 111, 120, 129, 195

Infinispan Infinispan. 14, 109, 111, 120, 121, 123, 124, 129, 130, 169, 221, 222

INIST **IN**stitut de l'**I**nformation **S**cientifique et **T**echnique. 270

INSA Institut National des Sciences Appliquées. 223

Inserm **I**nstitut **N**ational de la **S**anté et de la **R**echerche **M**édicale. 64

IRI **I**nternationalized **R**esource **I**dentifier. 91, 92

ISO **I**nternational **O**rganization for **S**tandardization. 45, 48, 49

Java™ Java™. 17, 120, 122, 124, 165, 166, 171, 206

JavaCC™ **Java** **C**ompiler **C**ompiler™. 10, 127, 165, 166, 173, 177

JSON **J**ava**S**cript **O**bject **N**otation. 50, 95, 102

LERUDI **L**ecture **R**apide en **U**rgence du **D**ossier **I**nformatique du patient. 46, 118

LESIM **L**aboratoire d'**É**pidemiologie, **S**tatistique et **I**nformatique **M**édicales. 35

LIMICS **[U1142, Paris 6 & Paris 13]** **L**aboratoire d'**I**nformatique **M**édicale et d'**I**ngénierie des **C**onnaissances en e-**S**anté. 31

LiSSa **Li**ttérature **S**cientifique en **Sa**nté. 14, 29, 38, 85, 169, 171, 173, 175, 178, 200, 221

Litis **[EA 41108]** **L**aboratoire d'**I**nformatique, du **T**raitement de l'**I**nformation et des **S**ystèmes. 25, 31

LTSI **L**aboratoire **T**raitement du **S**ignal et de l'**I**mage. 64

Lucene Apache **Lucene**. 15, 123, 124, 170, 194–196, 206

MCD **M**odèle **C**onceptuel de **D**onnées. 14, 113, 114, 130, 131, 180–182, 186, 188

MEDLINE® **MED**ical **LI**terature analysis and Retrieval System On**LINE**. 84

Microsoft Microsoft. 205, 206

MLD **M**odèle **L**ogique de **D**onnées. 113, 114, 120, 121, 124, 127, 130

MPD **M**odèle **P**hysique de **D**onnées. 113, 114

NINJAC **NINJAC** **I**s **N**ot **J**ust **A** **C**ache. 121, 123, 124, 169, 170, 194, 200, 206, 221, 222

NLM **N**ational **L**ibrary of **M**edicine. 32, 62, 84, 269

NoSQL **N**ot **o**nly **SQL**. 13–15, 17, 27, 29, 39, 41, 65, 96–98, 100–109, 111, 118, 120–124, 127, 130, 169, 194–196, 205–207, 221

NT–BT **T**erme spécifique – **T**erme générique. 32

OMOP **O**bservational **M**edical **O**utcomes **P**artnership. 58, 66

LAP Logiciel d'Aide à la Prescrition. 38

OMS **O**rganisation **M**ondiale de la **S**anté. 53

ORACLE Système de Gestion de Base de Données Relationnelles **ORACLE®**. 27, 38, 59, 63, 65, 66, 118, 120, 169, 205, 206, 221

Ovid Ovid®. 173, 174

OWL Web Ontology Language. 93, 115

OWL2 Web Ontology Language (Seconde version, 2009). 13, 93, 94

parser$_{\text{DocCiSMaF}}$ **Parser** de $\mathcal{L}_{\text{DocCiSMaF}}$–requête. 171, 173

parser$_{\mathbb{B}}$ **Parser** de $\mathcal{L}_{\mathbb{B}}$–requête. 14, 171, 177

parser$_{\text{share}}$ Parser du langage $\mathcal{L}_{\text{share}}$. 165–167, 171, 187, 189

PHR **P**ersonal **H**ealth **R**ecords. 45

PME Petite ou Moyenne Entreprise. 31, 224

PMSI **P**rogramme de **M**édicalisation des **S**ystèmes d'**I**nformation. 26, 43, 47, 52–54, 57, 64, 117, 211

POJO **P**lain **O**ld **J**ava **O**bject. 120, 121, 124, 166

Postgre SQL PostgreSQL. 59, 111, 121

PubMed PubMed. 84, 85, 174

R&D **R**echerche et **D**éveloppement. 61

RAM **R**andom– **A**ccess **M**emory. 97, 101, 102, 106, 109, 120, 207, 222

RAVEL [**ANR–11–TECS–012**] **R**etrieval **A**nd **V**isualization in **EL**ectronic health records. 17, 29, 31, 35, 36, 117, 118, 130, 184, 185, 190, 221

RDF **R**esource **D**escription **F**ramework. 13, 27, 41, 86, 87, 91–96, 106, 115–117

RDFS **R**esource **D**escription **F**ramework **S**chema. 13, 93, 94, 115

$\mathcal{L}_{\text{DocCiSMeF}}$**–requête** **R**equêtes écrites dans le langage $\mathcal{L}_{\text{DocCiSMeF}}$. 14, 21, 171–173, 175, 177, 178

$\mathcal{L}_{\mathbb{B}}$**–requête** **R**equêtes écrites dans le langage $\mathcal{L}_{\mathbb{B}}$. 14, 17, 21, 171–173, 176, 177

$\mathcal{L}_{\#}$**–requête** **R**equêtes écrites dans le langage $\mathcal{L}_{\#}$. 14, 171–173

$\mathcal{L}_{\text{share}}$**–requête** **requête** écrite dans le langage $\mathcal{L}_{\text{share}}$. 14, 15, 17, 19, 129, 147, 148, 165–167, 169–173, 175, 177, 179, 181, 184, 186–191, 195, 200, 201, 203, 206, 208, 209, 222, 224

REST **RE**presentational **S**tate **T**ransfer. 33, 50

RI **R**echerche d'**I**nformation. 14, 17, 25–27, 29, 31, 35–39, 41, 43, 47, 51, 55, 56, 63, 65–67, 69–74, 76, 78, 79, 81–87, 89, 90, 96, 97, 102–105, 111, 113, 118, 123–125, 127–130, 147, 168–171, 173–175, 177–179, 194, 200, 201, 205, 206, 221–223

RIDoPI **R**echerche d'**I**nformation dans le **Do**ssier **P**atient **I**nformatisé. 190, 221

SA Voir–**A**ussi. 32

SGBD **S**ystème de **G**estion de **B**ase de **D**onnées. 13, 14, 27, 29, 41, 66, 90, 96–98, 100–109, 112, 121, 124, 169, 170, 187, 194, 195, 200, 206, 221, 222

SGBDR **S**ystème de **G**estion de **B**ase de **D**onnées **R**elationnelles. 27, 29, 38, 39, 41, 59, 62, 63, 65, 66, 90, 94–98, 100, 103, 106, 109, 111, 113, 114, 118, 120, 121, 123, 124, 128, 169, 190, 195, 205–207, 221

SI **S**ystème d'**I**nformation. 14, 15, 17, 25–29, 38, 41, 45–47, 64, 66, 109, 113–118, 121–124, 128, 130, 146, 148, 168, 169, 173, 177, 194, 196, 197, 205–207, 221

SIFaDo [**ANR–11–TECS–0014**] **S**aisie **I**nformatique **Fa**cile des **Do**nnées médicales. 31, 35–37, 221

SIH **S**ystème d'**I**nformation **H**ospitalier. 43, 46, 49, 58, 64, 205, 210, 211, 217

SMEYEDAT **SM**art **EYE** **DAT**abase. 64, 66, 128

SOAP **S**imple Object **A**ccess **P**rotocol. 33, 94

SOC **S**ystème d'**O**rganisation des **C**onnaissances. 26, 32, 37, 38, 60, 83, 84, 116, 205, 207

SPARQL **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage. 13, 90, 94–96, 106

SQL **S**tructured **Q**uery **L**anguage. 62, 66, 94, 95, 97, 98, 106, 111, 118, 123, 128, 130, 131, 133, 169, 170, 190, 191, 193–195

SRI **S**ystème de **R**echerche d'**I**nformation. 26, 27, 29, 41, 46, 47, 56, 71–73, 76, 78, 80, 83, 85, 86, 97, 111, 128, 165

SSE$_{NoSQL}$ **S**emantic **S**earch **E**ngine **NoSQL**. 11, 14, 15, 29, 38, 39, 41, 111, 112, 130, 131, 148, 152, 165, 166, 168–171, 173, 175, 178, 180, 186–190, 193–195, 197, 200, 205, 206, 211, 215, 218, 221–224, 253

SSE$_{SQL}$ **S**emantic **S**earch **E**ngine **SQL**. 10, 15, 17, 29, 38, 41, 111, 117–119, 130, 131, 148, 152, 165, 166, 168–171, 180, 184, 185, 187, 189, 190, 193–195, 197, 221, 223, 224, 253

STRIDE **S**tanford **T**ranslational **R**esearch **I**ntegrated **D**atabase **E**nvironment. 13, 17, 58, 61–63, 66, 67, 128, 222

T2A **T**arification à l'**A**ctivité. 53, 57

TAL **T**raitement **A**utomatique du **L**angage. 178–180, 182, 206, 207, 224

TALN **T**raitement **A**utomatique du **L**angage **N**aturel. 31, 57, 71

$\mathcal{T}Box$ ensemble de axiomes Terminologiques. 115

TecSan **Tec**hnologies pour la **San**té et l'autonomie. 35

TF–IDF **T**erm **F**requency–**I**nverse **D**ocument **F**requency. 78

TIBS **T**raitement de l'**I**nformation en **B**iologie et **S**anté. 31

TO **T**erminologie/**O**ntologie. 32, 33, 35, 37–39, 62, 65, 84–86, 127, 174, 178, 201, 206, 207, 251

UMLS **U**nified **M**edical **L**anguage **S**ystem. 32, 33, 84, 207

URI **U**niform **R**esource **I**dentifier. 13, 91, 92, 95

URL **U**niform **R**esource **L**ocator. 50

USA **É**tats **U**nis d'**A**mérique. 205

W3C **W**orld **W**ide **W**eb **C**onsortium. 13, 89, 91, 92, 94, 95, 116

XML **E**xtensible **M**arkup **L**anguage. 33, 50, 95, 102

Chapitre 1

Introduction générale

Sommaire

1.1 Introduction 26
1.2 Objectifs des travaux de recherche 29
1.3 Contexte des travaux 31
1.4 Quelques réalisations 32
1.4.1 Health Terminology/Ontology Portal (HeTOP) 32
1.4.2 L'Extracteur de Concepts Multi-Terminologiques (ECMT) 33
1.5 Les projets de recherche 35
1.5.1 Le projet Retrieval And Visualization in ELectronic health records . . . 35
1.5.2 Le projet Saisie Informatique FAcile des DOnnées médicales 36
1.6 Les données de santé au CHU de Rouen 38
1.6.1 Le Dossier Patient Informatisé du CHU de Rouen 38
1.6.2 L'Entrepôt de Données de Santé Sémantique (EDSS) 38
1.7 Organisation du manuscrit 41

Dans ce mémoire je présente mes travaux de recherche effectués au sein du **D**épartement d'**I**nformatique et d'**I**nformation **M**édicales (D2IM) du **C**entre **H**ospitalier **U**niversitaire **de Rouen** – Hôpital Charles Nicolle (CHU de Rouen) [1] dans le cadre de la réalisation d'une thèse d'informatique au sein du **L**aboratoire d'**I**nformatique, du **T**raitement de l'**I**nformation et des **S**ystèmes (LITIS [EA 41108]) [2]. Cette dernière s'articule autour de la problématique de **R**echerche d'**I**nformation (RI) au sein de données cliniques. J'ai, dans le cadre de ce travail, conçu un langage de requête spécifique et en ai réalisé l'implémentation sous forme de deux preuves de concepts de moteurs de recherches. Une première approche générale de ces deux outils explicitant synthétiquement leurs rôles au sein du **S**ystème d'**I**nformation (SI) est donnée dans la section suivante. Ces rôles s'appuient sur de nombreux acquis du D2IM aussi bien sur le plan théorique que pratique et notamment sur une modélisation générique des informations dont ils en fournissent un accès tout aussi générique. L'un de ces deux moteurs est par ailleurs aujourd'hui pleinement exploité au sein du SI du D2IM y compris en dehors du contexte d'information clinique.

1. url : `http://www3.chu-rouen.fr/internet`
2. url : `http://www.litislab.eu/`

1.1 Introduction

L'informatisation des sciences d'une manière générale a donné naissance à de nouvelles disciplines à part entière et notamment à l'informatique médicale en ce qui concerne la Santé. Depuis les premières utilisations d'ordinateurs en médecine au début des années 1950 [5] et l'augmentation progressive de leur utilisation dans les années 1960, le format papier a graduellement cédé du terrain au profit de formats de stockage électroniques divers et variés. Les données cliniques acquises lors des différentes prises en charge médicales des patients sont aujourd'hui en partie regroupées au sein de **D**ossiers **P**atients **I**nformatisés (DPIs). Ceux-ci correspondent à la version électronique du traditionnel dossier patient utilisé par les professionnels de Santé [6–8].

Dans une plus large mesure, les **E**ntrepôts de **D**onnées de Santé (EDSs) permettent une centralisation encore plus importante de données de santé. Ils incluent notamment les DPIs mais également d'autres données telles que celles issues des dossiers infirmiers par exemple (la nature, la complexité et les enjeux liés aux données cliniques sont abordés dans le chapitre 2 p. 43). Un état de l'art des **S**ystèmes de **R**echerche d'**I**nformation (SRIs) permettant l'accès aux informations contenues dans un EDS sera également réalisé.

Les données d'un EDS peuvent être utilisées à des fins de soins. Cependant, l'exhaustivité relative et la variété des données contenues au sein de ces EDSs permettent un usage secondaire des données cliniques qu'il contiennent. On peut notamment citer à ce titre les systèmes d'aide à la décision clinique [9], la recherche clinique et la constitution de cohortes de patients [10], l'éducation [11, 12], l'étude et la définition d'indicateurs, etc. Au CHU de Rouen, depuis 2017, le **D2IM** est en charge du développement d'un EDS destiné à l'optimisation du **P**rogramme de **M**édicalisation des **S**ystèmes d'**I**nformation (PMSI). Mon travail de thèse s'intègre à ce contexte et vise à proposer des méthodes de RI au sein des données de cet EDS. Il est, à ce titre, assujetti à plusieurs contraintes :

— des contraintes généricité des fonctionnalités de RI qu'il propose. Celles-ci doivent permettre de couvrir un large panel d'accès à des données cliniques variées en préservant la cohérence clinique de l'information qu'elles définissent et de répondre à des cas d'usages concrets ;

— des contraintes de performances compte-tenu de la volumétrie des données des EDSs de manière générale qui impose une optimisation et une réflexion sur l'architecture technique des outils dévellopés qui en découlent ;

— des contraintes d'intégration dans le SI pré-existant.

L'objectif de la RI est de fournir un accès à des données spécifiques d'un domaine d'application particulier et peut servir, dans une vision plus large, diverses applications vis à vis de cette discipline : consultation d'informations, de recherches d'informations spécifiques, gestions, de pilotages ou encore prises de décisions. La RI au sein des données de santé n'est pas un domaine de recherche nouveau. De nombreux travaux ont été effectués dans ce domaine notamment dans le cadre de la RI documentaire et bibliographique [13–17]. Cette dernière présente néanmoins des spécificités par rapport à la RI classique. Elle repose sur de nombreux **S**ystèmes d'**O**rganisation des **C**onnaissances (SOCs) [18–20] qui permettent de représenter la sémantique de l'information médicale. Les SOCs et la RI sémantique constituent un axe de recherche du **D2IM**. La RI documentaire et bibliographique permet un accès à de la littérature scientifique ou à des informations et des connaissances théoriques ayant déjà été éprouvées dans la pratique [21, 22]. La médecine est cependant une « science du vivant » ou plutôt un « art du vivant » dont la mise en pratique et surtout la progression requiert la mise en correspondance de ces connaissances théoriques avec des données cliniques. La RI au sein des données cliniques joue donc un rôle stratégique en santé. Cette dernière fait néanmoins l'objet de défis supplémentaires principalement dus à la variété des types d'informations cliniques et leur complexité structurelle et sémantique. Bien qu'une grande partie de cette information clinique réside encore aujourd'hui dans les textes cliniques non structurés, l'utilisation conjointe et la mise en relation d'informations structurées et non structurées s'avèrent indispensables pour une RI pleinement efficace [23, 24]. Dans le chapitre 3

p. 69 de ce mémoire, un état de l'art des principaux types de RIs est réalisé. Les spécificités de la RI sur les données cliniques sont également exposées afin de tenter d'expliquer les raisons pour lesquelles les méthodes de RI classiques peuvent s'avérer inefficaces ou inadaptées à ce cas spécifique. D'un point de vue conceptuel, l'information clinique est composée d'entités multiples entretenant des liens logiques entre elles. Ainsi, la RI au sein de ces données, necessite avant tout un modèle de représentation de l'information plus générique que celui utilisé pour la RI documentaire et bibliographique.

La structure de graphe permet notamment une expression de l'information à la fois intuitive et granulaire. Ce type de structure est particulièrement adaptée au domaine de la santé dont l'information s'articule autour de multiples notions, mesures quantitatives et informations fondamentalement différentes entretenant des liens lourds de sens pour les professionnels de santé. Ces éléments peuvent, en effet, être modélisés comme des entités d'un graphe. L'utilisation de graphes dans le domaine de la santé est de plus en plus répandu notamment pour leurs capacités analytiques [25, 26] et beaucoup d'initiatives ont vu le jour suite aux travaux du Web sémantique [27]. Le Web sémantique propose, par exemple, le modèle de graphe **R**esource **D**escription **F**ramework (**RDF**) comme modèle de représentation de base de l'information. Ce modèle ainsi que les concepts entourant le Web sémantique seront traités dans le chapitre 4 p. 89. Ce modèle apporte de grandes possibilités sur le plan théorique notamment en terme de modélisation de l'information. Des accès inférentiels à ces données peuvent en outre être fournis par l'intermédiaire d'Ontologies. Le Web sémantique repose néanmoins technologiquement sur le Web, ce qui le rend peu performant au regard de la volumétrie des données d'un EDS. La problématique de la volumétrie de données est centrale dans de nombreux domaines scientifiques. En géologie, par exemple, les séquenceurs sismiques génèrent 2 *To* octets de données par run. Il en est de même en ce qui concerne le milieu médical où chaque expérimentation sur l'ADN génère environ 1 *To* de données. En termes de quantité de données, les EDS s'intègrent à cette problématique de gestion de données massives (Big Data). Depuis le début des années 2000, une multitude de nouveaux **S**ystèmes de **G**estion de **B**ase de **D**onnées (SGBDs) alternatifs ont fait leur apparition afin de répondre à ces problématiques. Ces derniers permettent le stockage de données massives tout en leur garantissant un accès basique performant. Dans le chapitre 4 p. 89, un état de l'art de ces bases de données alternatives sera effectué afin d'expliquer dans quelle mesure ces dernières peuvent être, ou ne pas être, utiles à la problématique de RI au sein d'un EDS.

Le SI du **D2IM** reposait sur le Système de Gestion de Base de Données Relationnelles **ORACLE**® (ORACLE®) [3]. Depuis plusieurs années, notre équipe développe et exploite un modèle de données relationnel inspiré des technologies du Web sémantique. Ce dernier permet une modélisation générique des données. Face à des besoins de montée en charge, un SGBD **N**ot **o**nly **SQL** (NoSQL) [4] est aujourd'hui utilisé pour interfacer l'accès aux données entre les applications et le **S**ystème de **G**estion de **B**ase de **D**onnées **R**elationnelles (SGBDR). Dans le cadre de cette thèse, j'ai, dans un premier temps, réalisé une preuve de concept de SRI sur différents jeux de données patients stockés dans le SGBDR ORACLE® historique. J'ai notamment réalisé des tests de montée en charge qui m'ont permis de remettre en cause l'exploitation de celui-ci dans le contexte de RI sur des données cliniques. J'ai ensuite réalisé, dans un second temps, une deuxième preuve de concept basée cette fois sur une technologie NoSQL. Le chapitre 5 p. 111 s'attache à décrire le modèle de données générique du **D2IM** ainsi que la modélisation et l'intégration des données cliniques au sein des différents SGBDs.

La méthode de RI que je propose dans ce travail repose sur la modélisation d'un langage de requêtes Booléen « augmenté ». Il permet d'accèder à l'ensemble des entités conceptuelles composant l'information de santé. Dans le chapitre 6 p. 127, je donne une description concrète et accompagnée d'exemples de ce dernier. La grammaire formelle permettant d'engendrer ce langage est également explicitée de manière rigoureuse. Dans le chapitre 7 p. 169, je précise la

3. url : `https://www.oracle.com/index.html`
4. : « Pas Seulement **SQL** »

manière dont j'ai implémenté les preuves de concepts afin qu'elles puissent d'une part, s'intégrer au SI du **D2IM** et d'autre part, exécuter des requêtes écrites dans le langage que j'ai défini.

Enfin, le chapitre 8 p. 199 décrit l'intégration de mes travaux dans le cadre de l'EDS du CHU de Rouen. Une étude menée en 2018 visant à évaluer la capacité de l'EDS dans son ensemble à répondre à des critères d'inclusion et d'exclusion d'études cliniques y est notamment présentée.

1.2 Objectifs des travaux de recherche

La problématique brièvement présentée dans la section précédente permet de mettre en évidence quelques éléments susceptibles de rendre les méthodes de la RI classiques inadéquates pour la problématique des données cliniques. Ainsi, il existe une réelle nécessité de trouver des méthodes adaptées pour modéliser ces données et les rendre exploitables.

Mes travaux de thèse ont été réalisés au sein du **D**épartement d'**I**nformatique et d'**I**nformation **M**édicales (D2IM) du CHU de Rouen. L'essentiel de mon travail a consisté à proposer une méthode d'interrogation de données cliniques et, dans une plus large mesure, de données interconnectées. J'ai ainsi proposé un langage de requêtes Booléen spécifique permettant d'accéder à des données modélisées sous la forme d'un graphe : le langage $\mathcal{L}_{\bullet}$. Ce travail a abouti opérationnellement au développement de deux preuves de concepts de SRIs :

— le **S**emantic **S**earch **E**ngine **SQL** (SSE_{SQL}) ;

— le **S**emantic **S**earch **E**ngine **NoSQL** (SSE_{NoSQL}).

Le SSE_{SQL} m'a permis d'élaborer les fondements de la méthode et d'en évaluer empiriquement son utilité dans le cadre du projet de recherche **R**etrieval **A**nd **V**isualization in **EL**ectronic health records (RAVEL [ANR–11–TECS–012]) [5] auquel le D2IM a participé. Ce dernier était alors destiné à fournir un accès sémantique à l'information contenue au sein du DPI. Compte tenu de son incapacité à assurer des performances « satisfaisantes » avec la montée en charge des données, j'ai également modélisé et implémenté une seconde preuve de concept : le SSE_{NoSQL}.

Le SSE_{SQL} et le SSE_{NoSQL} diffèrent principalement de part le type de SGBD qu'ils exploitent et qui correspondent respectivement, à un SGBDR et un SGBD NoSQL. En terme de périmètre fonctionnel, le SSE_{SQL} propose des fonctionnalités plus étendues que le SSE_{NoSQL}. C'est, néanmoins, ce dernier qui est aujourd'hui utilisé comme moteur de recherche interne de plusieurs outils de RI du D2IM y compris en dehors du cas d'usage de données cliniques.

L'un des aspects de mon travail de thèse a également consisté à intégrer le SSE_{NoSQL} au sein du SI du D2IM. La généricité de ce dernier a permis de l'utiliser dans des contextes de RI variés. Il constitue aujourd'hui le moteur de recherche interne de divers outils :

— Moteur de Recherche **Doc**umentaire basé sur le CiSMeF (Doc'CiSMeF) [6] qui permet une RI documentaire en santé ;

— **Li**ttérature **S**cientifique en **Sa**nté (LiSSa) [7] qui permet une RI bibliographique en santé ;

— **A**ccès **S**émantique à l'**I**nformation de **S**anté (ASIS) qui permet un accès sémantique à l'information de l'**E**ntrepôt de **D**onnées de **S**anté **S**émantique (EDSS) du CHU de Rouen.

Néanmoins, l'ambition première du SSE_{NoSQL} reste de fournir des fonctionnalités d'accès aux informations cliniques de l'EDSS du CHU de Rouen. Outre le langage de requête $\mathcal{L}_{\bullet}$, j'ai également proposé d'autres modes d'interrogation qui se sont concrétisés par d'autres langages tel que $\mathcal{L}_{\#}$, $\mathcal{L}_{\mathbb{B}}$. Ces derniers possèdent une expressivité propice à différents type de recherche d'information. Une partie importante de mon travail s'est attachée à rendre accessible ces modes d'interrogation de manière cohérente au sein du SSE_{NoSQL}. L'ensemble de mes apports et de mes travaux sont synthétisés dans la Figure 1.1. Les éléments colorés en rouge correspondent aux outils et méthodes que je propose dans le cadre de la réalisation de ma thèse.

5. : « **R**echerche Et **V**isualisation des informations dans le dossier du patient **EL**ectronique »
6. url : `http://doccismef.chu-rouen.fr/`
7. url : `https://www.lissa.fr/dc/#env=lissa`

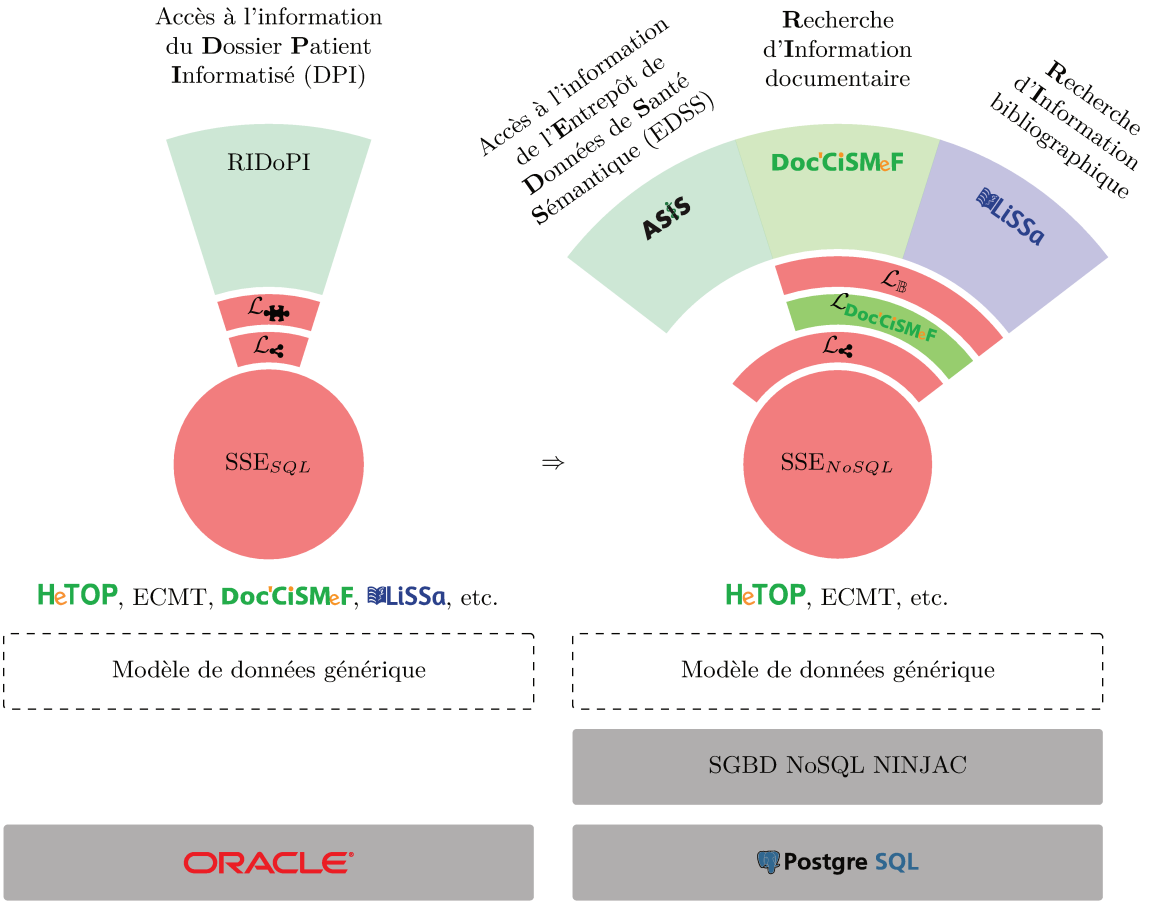


FIGURE 1.1 – Contexte des travaux réalisés.

1.3 Contexte des travaux

Le **C**atalogue et **I**ndex des **S**ites **M**édicaux de langue **F**rançaise (CiSMeF) [8] est un projet initié en février 1995 au CHU de Rouen par Stéfan J. DARMONI (Professeur d'Informatique Médicale) et Benoit THIRION (Bibliothécaire). Ce projet a débuté dès la création du site Web du CHU de Rouen et visait à mettre à disposition des sites et des documents Francophones de santé de qualité.

Sur le plan hospitalier, l'équipe est rattachée au D2IM du CHU de Rouen.

> ⚠ **Remarque 1 :**
> *Dans la suite de ce mémoire, on emploiera parfois improprement l'expression « équipe CiSMeF » pour désigner l'ensemble des membres de l'équipe de recherche initiée avec le projet CiSMeF. Dans la majorité des cas on parlera cependant du « D2IM » pour désigner cette dernière.*

Le D2IM est une équipe pluridisciplinaire actuellement composée de 4 médecins, 5 ingénieurs, 3 documentalistes et d'un maître de conférence en médecine générale. Le D2IM comporte également deux internes de santé publique, un interne de médecine générale et quatre doctorants.

Sur le plan universitaire, l'équipe est rattachée depuis 2016 au **L**aboratoire d'**I**nformatique **M**édicale et d'**I**ngénierie des **C**onnaissances en e-**S**anté (LIMICS [U1142, Paris 6 & Paris 13]) [9] ainsi qu'à l'équipe **T**raitement de l'**I**nformation en **B**iologie et **S**anté (TIBS) [10] du Ulitis [EA 41108] faisant lui même partie de l'Université de Rouen.[11]

Aujourd'hui les thématiques de recherche du D2IM ont évolué et ne se limitent plus simplement à la mise à disposition de ressources en santé et à leurs indexations. De nombreux travaux ont été entrepris autour de l'informatique médicale et plus particulièrement des problématiques de RI (e.g. création en 2000 du DocCiSMeF), de terminologie de santé (e.g. création du portail terminologique **H**ealth **T**erminology/**O**ntology **P**ortal (HeTOP) [12, 13]) et des problématiques de **T**raitement **A**utomatique du **L**angage **N**aturel (TALN) (e.g. création de l'**E**xtracteur de **C**oncepts **M**ulti- **T**erminologiques (ECMT)).

L'équipe a participé à de nombreux projets de l'**A**gence **N**ationale de la **R**echerche (ANR) [14] dans le cadre du programme Technologies pour la Santé, ainsi qu'à des projets en partenariat avec des entreprises privées telles que le Vidal ou omicX. De plus, nous avons valorisé certains de nos outils (i.e. HeTOP, ECMT) via la PME Alicante des Hauts de France.

En 2017, le CHU de Rouen a officiellement confié au D2IM le projet de création de l'EDSS qui constitue le contexte majeur dans lequel s'inscrit le travail de thèse décrit dans ce mémoire. Néanmoins, sur le plan scientifique, ce projet fait suite à divers travaux initiés sur des problématiques de RI, d'ergonomie et de saisie du DPI notamment dans le cadre de la participation aux projets RAVEL [ANR–11–TECS–012] (cf. sous-section 1.5.1 p. 35) et **S**aisie **I**nformatique **Fa**cile des **Do**nnées médicales (SIFaDo [ANR–11–TECS–0014]) (cf. sous-section 1.5.2 p. 36) ainsi que des travaux de thèses de Ahmed-Diouf DIRIEH DIBAD [28] ou encore plus récemment de Chloé CABOT [29] portant respectivement sur la RI multi-terminologique et la RI clinomique au sein du DPI .

8. url : `http://www.chu-rouen.fr/cismef/`
9. url : `http://www.limics.fr/`
10. url : `http://www.chu-rouen.fr/tibs/`
11. url : `http://www.univ-rouen.fr/`
12. url : `http://www.hetop.eu/`
13. : « **P**ortail **T**erminologique/**O**ntologique en Santé »
14. url : `http://www.agence-nationale-recherche.fr/`

1.4 Quelques réalisations

Depuis sa création en 1995, l'équipe CiSMeF, et plus récemment le D2IM, a initié plusieurs projets qui ont donné lieu à la création de plusieurs outils utilisés, pour certains, quotidiennement par une communauté de divers profils. L'EDSS du CHU de Rouen repose sur une architecture permettant l'exploitation conjointe de plusieurs de ces outils et notamment de l'HeTOP et de l'ECMT. Une description de ces derniers ainsi que de leurs principales caractéristiques sont données dans les sections suivantes.

1.4.1 Health Terminology/Ontology Portal (HeTOP)

Le **H**ealth **T**erminology/**O**ntology **P**ortal (HeTOP) [15, 16] est un portail terminologique développé dans le cadre du travail de thèse de Julien GROSJEAN [30]. Il donne un accès à plusieurs SOCs et permet d'accéder à 2 639 620 concepts et 10 735 905 termes ainsi qu'à leurs structures sémantiques (i.e. leurs libellés préférés, leurs termes alternatifs, leurs descriptions textuelles, leurs hiérarchies, leurs relations sémantiques avec d'autres concepts, etc.). Ces concepts proviennent, en 2017, de 75 **T**erminologies/**O**ntologies (TOs) diverses de santé. HeTOP propose, en plus des relations sémantiques nativement présentes au sein des TOs, diverses relations intra-terminologiques et inter-terminologiques qui peuvent être définies manuellement ou automatiquement mais supervisées manuellement. Ces dernières correspondent à :

- des Correspondance **E**xacte (EM) [17].
- des relations **T**erme spécifique – **T**erme générique (NT–BT) [18].
- des relations **T**erme générique – **T**erme spécifique (BT–NT) [19].
- des relations de type Voir–**A**ussi (SA) [20].
- des relations vers les types sémantiques du Métathésaurus de l'**U**nified **M**edical **L**anguage **S**ystem (UMLS) [21, 22][31].
- etc.

D'un point de vue standardisation, le portail HeTOP est a priori compatible avec la norme ISO 25964-1 [32] et suit les recommandations de Tao et al. [33] telles que la définition d'un libellé préféré, de termes alternatifs, etc.

L'UMLS comporte un Métathésaurus édité et maintenu par la **N**ational **L**ibrary of **M**edicine (NLM) [23]. Tout comme HeTOP, ce dernier regroupe un grand nombre de TOs. Il agrège, au sein d'une structure unifiante, les différents concept issus de ces divers vocabulaires contrôlés en regroupant les concepts sémantiquement identiques sous un même « Concept » muni d'un **C**oncept **U**nique **I**dentifier (CUI) [24].

Bien que l'UMLS soit largement exploité dans la littérature et qu'il intègre un nombre plus important de SOCs, HeTOP propose une couverture française plus étendue que ce dernier. Cet outil interlingue [25] propose un accès aux différents concepts et termes dans 32 langues différentes. Il faut cependant noter que toutes ces langues ne sont pas disponibles pour chaque concept.

Dans sa version de 2017, l'UMLS fournit un accès en français pour 11 de ses SOCs. HeTOP intègre au total 17 des SOCs de l'UMLS. Parmi les 978 233 CUIs que ces deux derniers ont en commun, seulement 143 762 (i.e. 14,7%) concepts en français sont nativement fournis par

15. url : `http://www.hetop.eu/`
16. : « **P**ortail **T**erminologique/**O**ntologique en Santé »
17. : « **E**xact–**M**atch »
18. : « **N**arrower **T**erm – **B**roader **T**erm »
19. : « **B**roader **T**erm – **N**arrower **T**erm »
20. : « **S**ee–**A**lso »
21. url : `https://www.nlm.nih.gov/research/umls/`
22. : « **S**ystème **U**nifié de la **L**angue **M**édicale »
23. : « Bibliothèque **N**ationale américaine de **M**édecine »
24. : « **I**dentifiant **U**nique de **C**oncept »
25. : « cross–lingual »

l'UMLS contre 428 854 (i.e. 43,8%) pour HeTOP. Une partie des TOs ont, en effet, été traduites partiellement ou totalement au sein du HeTOP (e.g. **S**ystematized **N**omenclature **O**f **MED**icine (version 3.5) (SNOMED 3.5) (52,3%), **M**edical **S**ubject **H**eadings (MeSH) descriptors (100%), **N**ational **C**ancer **I**nstitute **T**hesaurus (NCIt) (53,35%), Online Mendelian Inheritance in Man (OMIM) (94,90%), **H**uman **P**henotype **O**ntology (HPO) (80,11%), Radlex (22,1%), etc.). Plus généralement, 50% des 2,64 millions de concepts accessibles par HeTOP sont fournis en français ainsi que 19,1% des 10,74 millions de termes.

Sur le plan applicatif, HeTOP se présente sous la forme d'un service Web reposant sur les protocoles **S**imple **O**bject **A**ccess **P**rotocol (SOAP) [26] ou **RE**presentational **S**tate **T**ransfer (REST). Une application web lui est également dédiée.

1.4.2 L'Extracteur de Concepts Multi-Terminologiques (ECMT)

L'**E**xtracteur de **C**oncepts **M**ulti- **T**erminologiques (ECMT) est un outil un d'annotation sémantique. Il permet d'identifier et de mettre en correspondance les concepts Terminologiques et/ou Ontologiques du portail HeTOP avec les mots et/ou expressions des textes exprimés en langage naturel. Cet outil permet de plus de tirer parti de la sémantique des concepts non seulement dans le processus d'extraction, en exploitant les termes alternatifs (synonymes) des concepts, mais également fonctionnellement en proposant diverses options :

- récupération, pour chaque terme extrait, des termes alternatifs, des ancêtres et descendants hiérarchiques, des concepts alignés ou encore des types sémantiques issus de l'UMLS ;
- filtrage et exclusion basés sur les types de concept ;
- priorisation basée sur la terminologie source des concepts ;
- affinage de l'ensemble des concepts renvoyés afin de ne conserver que ceux couvrant un maximum de termes du texte.

L'algorithme interne de l'ECMT repose sur celui du **Sac de Mot**. La correspondance entre les concepts et les expressions textuelles est essentiellement établie en recherchant au sein du vocabulaire contrôlé les différentes combinaisons des mots du texte. L'ECMT applique différents traitements à la fois sur les textes en langage naturel fournis en entrée et sur le vocabulaire contrôlé afin de réaliser ces combinaisons. La Figure 1.2 p. 34 synthétise ce processus. Les combinaisons sont formées au sein d'une fenêtre glissante paramétrable et restreinte à chaque phrase du texte. En plus de l'extraction de concepts, des mécanismes de reconnaissance de motifs [27] permettent la détection des négations mais aussi des données symboliques, numériques ou chronologiques particulièrement présentes dans les comptes-rendus hospitaliers.

Depuis sa création, l'ECMT a été exploité dans plusieurs projets financés par l'ANR et notamment dans [34, 35]. Une version adaptée de l'ECMT [29, 36] a, de plus, été évaluée dans le cadre des compétitions **C**onference and **L**abs of the **E**valuation **F**orum (CLEF) [28] (2015, 2016 et 2017) et obtenu les meilleurs résultats en français pour certaines tâches en 2017.

Enfin, l'ECMT peut être interrogé par un service web reposant sur les protocoles SOAP ou REST. Les données en sortie sont fournies au format **E**xtensible **M**arkup **L**anguage (XML). Une interface Web [29] est également disponible.

26. : « **P**rotocole d'**A**ccés **S**imple à des **O**bjets (distants) ». La notion d'objet est obselète depuis la version 1.2 du protocole SOAP et sa réécriture en fonction d'infoset XML. SOAP n'est actuellement plus un acronyme.
27. l'expression anglaise correspondante « pattern–matching » étant plus largement utilisée
28. : « **C**onférence et **L**aboratoires du **F**orum d'**É**valuation »
29. url : `http://ecmt.chu-rouen.fr/`

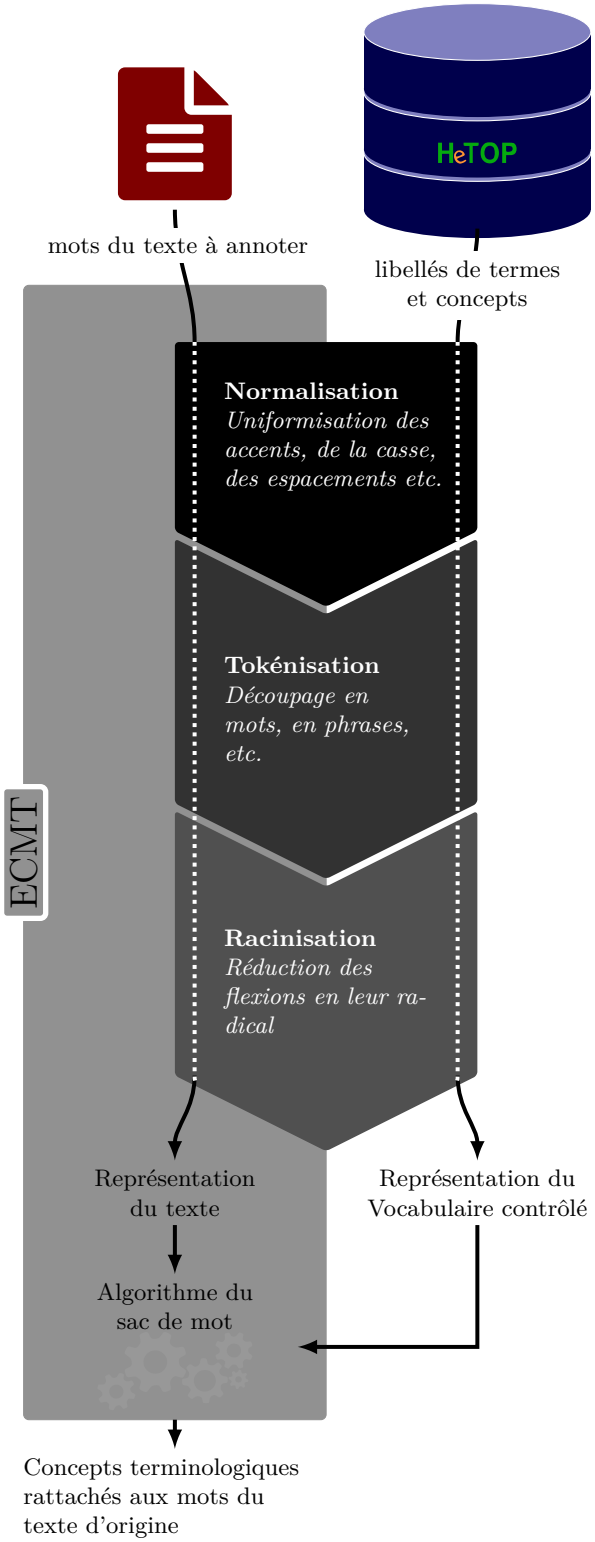


FIGURE 1.2 – Processus général de l'ECMT. Le texte fourni à l'ECMT ainsi que les différents libellés des concepts d'HeTOP sont normalisés, tokénisés et racinisés. L'algorithme de sac recherche les correspondances entre les mots du texte et des concepts issus du portail HeTOP.

1.5 Les projets de recherche

Entre 1995 et le début des années 2010, la RI documentaire et bibliographique ainsi que les TOs de Santé ont constitué les deux axes principaux sur lesquels les travaux de recherche du D2IM se sont portés. C'est majoritairement par l'intermédiaire de la participation à deux projets financés par l'**A**gence **N**ationale de la **R**echerche (ANR) [30] que le D2IM a abordé des problématiques liées à l'information clinique :

— le projet **R**etrieval **A**nd **V**isualization in **EL**ectronic health records (RAVEL [ANR–11–TECS–012]) [31] ;

— le projet **S**aisie **I**nformatique **Fa**cile des **Do**nnées médicales (SIFaDo [ANR–11–TECS–0014]).

Ces deux projets font partie de l'édition 2011 du programme de **Tec**hnologies pour la **San**té et l'autonomie (TecSan) [32] de l'ANR. Le projet RAVEL [ANR–11–TECS–012] constitue plus notoirement un point de départ historique ayant motivé les réflexions initiales relatives au langage de requête et plus généralement aux moteurs de recherche sémantique développés dans le cadre de cette thèse.

Dans cette section, ces deux projets sont brièvement décrits.

1.5.1 Le projet Retrieval And Visualization in ELectronic health records

Le D2IM a participé au projet RAVEL [ANR–11–TECS–012] en partenariat avec trois autres structures de recherche [33] ainsi que deux entreprises industrielles [34].

Celui-ci visait à fournir des outils performants permettant d'une part, de retrouver en temps réel des éléments de données médicales pertinents et d'autre part, de visualiser ces données selon des modèles de présentation intuitifs et synthétiques. Ce projet s'articulait autour de trois axes de recherche :

— l'indexation sémantique ;

— la **R**echerche d'**I**nformation ;

— la visualisation du dossier médical électronique.

Le D2IM a principalement été impliqué dans deux tâches de ce projet se rapportant essentiellement à l'axe de RI. La première d'entre elles consistait à fournir un modèle de données générique apte à contenir l'ensemble des données cliniques relatives aux dossiers patients. En ce qui me concerne, j'ai essentiellement été impacté par la deuxième tâche qui avait pour objectif de fournir un moteur de recherche en mesure de répondre à un certain nombre de cas d'usages précis définis dans le cadre du projet. Ces cas d'usages sont résumés dans la Table 1.1 p. 36.

Ces cas d'usages m'ont fourni une base de réflexion importante pour définir les méthodes de RI proposées dans le cadre de cette thèse. Le D2IM avait, dans un premier temps, envisagé d'adapter le moteur de recherche DocCiSMeF (incluant au passage une « augmentation » de son langage de requêtes) afin qu'il puisse répondre aux cas d'usages de la Table 1.1 p. 36. Étant en charge de ce développement, j'ai pu mettre concrètement en évidence le manque d'expressivité de ce moteur relativement aux besoins d'informations exprimés par de tels cas d'usages. Le projet RAVEL [ANR–11–TECS–012] m'a donc permis d'aborder la thématique de la RI au sein des

30. url : `http://www.agence-nationale-recherche.fr/`

31. : « **R**echerche Et **V**isualisation des informations dans le dossier du patient **EL**ectronique »

32. url : `http://www.agence-nationale-recherche.fr/projets-finances/?tx_lwmsuivibilan_pi1[Programme]=481`

33. Le **L**aboratoire d'**É**pidemiologie, **S**tatistique et **I**nformatique **M**édicales (LESIM) de l'université Victor Segalen (Bordeaux 2), la **M**aison **E**uropéenne des **S**ciences de l'**H**omme et de la **S**ociété (MESHS [USR 3185]) (Lille) et L'Université de Rennes 1 [Unité Inserm U936]

34. MEDASYS (éditeur et intégrateur de solutions santé) et le VIDAL

Cas d'usage transversal	
Objectif :	Élaboration d'une interface graphique permettant une visualisation d'ensemble du dossier patient et notamment par l'intermédiaire de visualisations chronologiques (e.g. lignes de vie (Time-line), tables ou listes chronologiques), graphiques (e.g. diagrammes) ou schémas (e.g. arbres de décision, relations causales)
rôle de la RI :	RI peu impactée, à part, éventuellement, dans le cadre d'une sélection des informations relatives à un problème médical particulier.
Cas d'usage de Cancérologie	
Objectif :	Fournir une aide à la préparation des dossiers médicaux pour les passages en Réunions de Concertations Pluridisciplinaires (RCP) de cancérologie.
rôle de la RI :	Intérêts multiples d'une RI sur les données cliniques notamment pour accéder aux documents (e.g. « dernier compte-rendu d'imagerie », « Traitements réalisés », « Imagerie avec notion de métastases » etc.
Cas d'usage de la polyarthrite rhumatoïde	
Objectif :	Vérifier la réalité du diagnostic de polyarthrite rhumatoïde par exemple dans le cadre d'une remise en cause du diagnostic suite à plusieurs échecs thérapeutiques.
rôle de la RI :	Ce cas d'usage requiert la mise en relation de « paramètres dont les formats diffèrent (biologie, traitement, score, etc.) ». Plus particulièrement, il nécessite la sélection et l'observation d'analyses biologiques spécifiques (e.g. protéine C réactive, vitesse de sédimentation, facteurs rhumatoïde et anticorps anti-peptides cycliques citrullinés) mais aussi de permettre l'application de contraintes particulières sur leurs valeurs (e.g. Polynucléaires neutrophiles inférieurs à 1 500/mm^3). De même, le requêtage des diagnostics (notamment associés) peut s'avérer très utile pour la détermination du score DAS–28 (Disease Activity Score).
Cas d'usage des anticoagulants	
Objectif :	Fournir une aide à la surveillance des patients traités par anticoagulants (risques d'hémorragies ou de thromboses, évolution des doses prescrites, des marqueurs de la coagulation).
rôle de la RI :	Dans le cadre de ce cas d'usage, la RI est davantage « exploratoire ». Elle peut, en effet, s'avérer utile afin d'effectuer un suivi du traitement, d'étudier les facteurs de risque de complication du traitement, les causes de surdosage possibles et les effets indésirables des traitements anticoagulants, etc. Elle se doit donc de permettre la navigation entre ces différents types d'informations.

TABLE 1.1 – Cas d'usages du projet RAVEL [ANR–11–TECS–012] et utilité/rôle potentiel de la RI pour chacun d'eux

données cliniques qui diffère de la RI classique de part la complexité et surtout l'hétérogénéité des informations qu'elle implique de manipuler et de mettre en relation. J'ai ainsi proposé un langage de requête permettant l'expression de certaines questions cliniques et implémenté ce dernier au sein d'un nouveau moteur de recherche. Des travaux portant sur la thématique des données cliniques et plus généralement des données complexes avaient auparavant déjà été entrepris au sein du D2IM notamment en ce qui concerne la conception d'un modèle de données générique [28, 30]. En s'appuyant sur ces derniers, le projet RAVEL [ANR–11–TECS–012] a initié, à travers la réalisation de mon travail de thèse, la mise en application concrète d'une RI spécifique.

1.5.2 Le projet Saisie Informatique FAcile des DOnnées médicales

Le projet SIFaDo [ANR–11–TECS–0014] visait la conception et l'évaluation de méthodes et d'outils ergonomiques pour faciliter la saisie et le codage de données textuelles et graphiques

dans les dossiers médicaux électroniques. Il s'agissait, dans ce projet, de favoriser la saisie de données structurées au sein des dossiers patients électroniques. La principale tâche du D2IM dans le cadre de ce projet était de fournir un accès, par l'intermédiaire du HeTOP, aux SOCs nécessaires au codage des informations du dossier patient. Bien que le projet SIFaDo [ANR–11–TECS–0014] ait joué un rôle moins impactant dans la modélisation des outils créés dans le cadre de ce travail de thèse, ce projet a néanmoins permis de valoriser l'apport sémantique des TOs du HeTOP pour la représentation de l'information clinique. Il apparaît ainsi pertinent vis à vis de la problématique plus générale de RI au sein d'un EDS. Comme il le sera précisé dans la section 1.6 p. 38, le travail décrit dans ce mémoire s'intègre d'un point de vue opérationnel dans le contexte plus large de développement d'un EDS sémantique au CHU de Rouen. En d'autres termes, SIFaDo [ANR–11–TECS–0014] constitue un projet fondateur de nombreux travaux de recherche du D2IM et notamment de ceux sur lesquels s'appuient les méthodes décrites dans ce mémoire.

1.6 Les données de santé au CHU de Rouen

1.6.1 Le Dossier Patient Informatisé du CHU de Rouen

Au CHU de Rouen, la gestion administrative informatisée des patients a débuté dès 1982. L'informatisation du Dossier Patient [37] a, quant à elle, débuté en 1986. Cependant, les règles d'accès à ce DPI prenant en compte la notion de prise en charge et de droit applicatif de l'utilisateur n'ont été formalisées qu'en 1992. Depuis cette date, le DPI du CHU de Rouen collecte et maintient des informations démographiques, cliniques et biologiques concernant les patients admis dans l'établissement.

D'un point de vue « applicatif », le SI dédié au DPI est géré depuis le début des années 2000 par le logiciel **C**Page **D**ossier **P**atient (CDP) développé par l'éditeur GIP CPage[35]. Au cours des années d'utilisation, CDP a été enrichi et complété par de nouveaux outils tel que Gestime (outil de gestion de rendez-vous) ou encore **A**pplication de **G**estion des **A**bsences et des **TE**mps (AGATE) (outil de gestion des droits) ainsi que certains dossiers de spécialités (e.g. néonat, urgences, etc.).
D'un point de vue « données », CDP repose sur le SGBDR ORACLE® muni d'un modèle de données complexes contenant une cinquantaine de tables.

Dans un souci de modernisation et d'harmonisation à la fois fonctionnelle et technologique du SI dédié au DPI jusque là riche mais vieillissant et « disparate », le CHU de Rouen a lancé en 2008 un « dialogue compétitif ». L'entreprise privée McKesson[36] a été choisie pour mettre en place une nouvelle solution pour le DPI du CHU de Rouen.
Cette dernière, aujourd'hui en cours d'implantation, reposera notamment sur CrossWay Hôpital[37] incluant différents modules et applications (e.g. CORA pour le codage de l'activité, son contrôle qualité et la gestion du Dossier Patient, **H**orizon **E**xpert **O**rders (HEO) pour la prescription multimodale (Logiciel d'Aide à la Prescrition (LAP)[38]), Pharma[39] de Computer Engineering pour la gestion du circuit du médicament et des dispositifs médicaux, UrQual[40] (McKesson) pour la gestion des urgences). La mise en place de cette nouvelle solution semble cependant actuellement compromise d'un point de vue à la fois technique et politique.

Parallèlement, le CHU de Rouen a confié au D2IM la création de l'EDS du CHU de Rouen. Les EDS, bien qu'incluant le DPI en terme de données, ne se destinent pas à une utilisation à visée de soins (cf. section 2.2 p. 56). Fort de son expérience en informatique médicale et de ses acquis en terme de TOs de santé, le D2IM entreprend depuis 2017 la création de l'EDSS du CHU de Rouen.

1.6.2 L'Entrepôt de Données de Santé Sémantique (EDSS)

Le travail réalisé dans le cadre de cette thèse s'inscrit pleinement dans le contexte de création de l'**E**ntrepôt de **D**onnées de **S**anté **S**émantique (EDSS). Bien qu'également utilisé aujourd'hui comme moteur de recherche sous-jacent aux applications de recherches documentaires et bibliographiques DocCiSMeF et LiSSa, les moteurs SSE_{NoSQL} et SSE_{SQL} ont été imaginés dans un objectif de RI au sein de données patient. Dans le cadre ce projet, le SSE_{NoSQL} constitue une des trois briques fonctionnelles majeures de l'EDSS.

L'EDSS permet une RI sémantique de ces données basée sur plusieurs SOCs. Il repose sur deux ensembles de données :

35. url : `http://www.cpage.fr/`
36. url : `http://www.mckesson.com/`
37. url : `http://www.cegedim-logiciels.com/nos-produits/nos-logiciels-cabinet-et-exercice-mixte/6-crossway.html`
38. : « Computerized Physician Order Entry (CPOE) »
39. url : `http://www.computer-engineering.fr/emodule-pharma`
40. url : `https://gammem.maincare.com/production-de-soins/m-urqual/m-urqual,35,41.html`

une base de connaissances : qui fournit plus de 75 TOs ;

une base de données de santé : qui gère les diverses données patient, les documents médicaux, les données cliniques et administratives, etc.

Les fonctionnalités de l'EDSS exploitent conjointement trois outils majeurs :

le portail terminologique HeTOP : dont le rôle majeur est de permettre un accès au différents concepts et relations sémantiques de la base de connaissance ;

l'annotateur sémantique ECMT : qui permet de mettre en correspondance les mots et expressions en langage naturel et les concepts termino-ontologiques de la base de connaissances ;

le moteur de recherche sémantique SSE$_{NoSQL}$: qui est dédié à la RI sémantique de l'information de santé.

La Figure 1.3 illustre l'architecture de cet entrepôt :

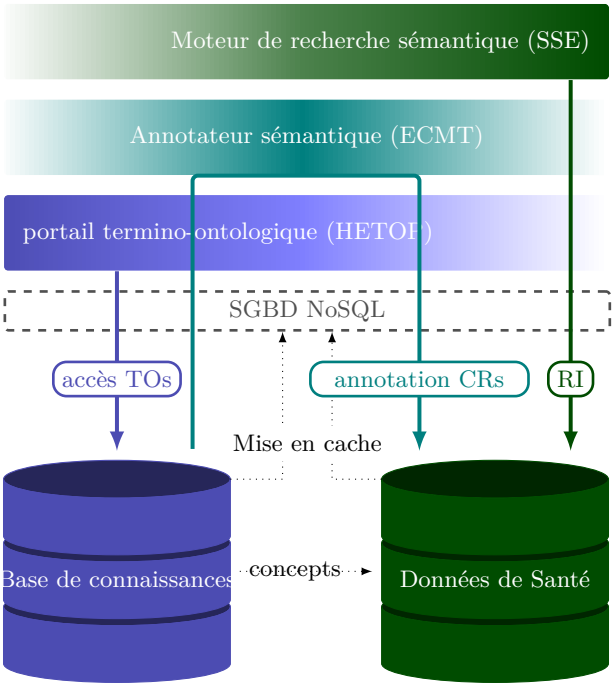


FIGURE 1.3 – Architecture globale de l'EDSS du CHU de Rouen

Les trois outils (viz. HeTOP, ECMT et SSE$_{NoSQL}$) sont répartis en trois couches distinctes. Chacune d'entre elles consomme les fonctionnalités des couches qui lui sont inférieures et fournit des fonctionnalités reposant sur un périmètre de données différent : le périmètre du portail HeTOP se limite à la base de connaissances tandis que celui de l'ECMT se trouve à l'intersection des données termino-ontologiques et des données de santé. Le SSE$_{NoSQL}$ quant à lui s'articule principalement autour des données de santé.

Ensemble HeTOP et l'ECMT permettent de donner une description sémantique de l'information de santé. Cette dernière est alors exploitée par le SSE$_{NoSQL}$ afin de fournir une RI sémantique de cette information.

Les données de la base de connaissances tout comme celles de santé sont stockées au sein d'un SGBDR muni d'un modèle de données générique (cf. chapitre 5 p. 111). Compte tenu de la volumétrie importante des données de Santé, un cache NoSQL a cependant été implémenté par l'équipe du D2IM afin d'interfacer les données et d'améliorer les performances des applications. Ces dernières accèdent et manipulent donc les données par l'intermédiaire de ce cache.

L'EDSS du CHU de Rouen est un projet actuellement en plein évolution. De nombreux efforts axés sur la modélisation des données cliniques, les outils en permettant un accès ou encore leur intégration sont en effet en cours aujourd'hui. La Table 1.2 donne, néanmoins, les volumétries de l'EDSS du CHU de Rouen à titre informatif :

Entité	Nombre
Patients	1 874 578
Hospitalisations	2 448 799
Séjours	13 468 460
Prises en charges	3 399 512
Diagnostics	9 627 103
Actes médicaux et chirurgicaux	8 653 193
Analyses biologiques unitaires	121 129 957
Textes cliniques	13 603 661
Annotations sémantiques	1 681 511 333

TABLE 1.2 – Volumétrie de l'EDSS au 21 mars 2019

1.7 Organisation du manuscrit

Dans ce mémoire, la nature, la complexité et les enjeux liés aux données cliniques sont, dans un premier temps, abordés dans le chapitre 2 p. 43. Ce dernier définit également la notion de DPI et d'EDS et un état de l'art des SRIs fournissant un accès aux informations contenues dans divers EDSs y est également réalisé.

Dans le chapitre 3 p. 69, les principaux types de RIs sont présentés. Les spécificités de la RI sur les données cliniques sont également exposées afin de tenter d'expliquer les raisons pour lesquelles les méthodes de RI classiques peuvent s'avérer inefficaces ou inadaptées à ce cas spécifique. Enfin, la nécessité, pour les données cliniques, d'un modèle de représentation de l'information plus générique que celui utilisé pour la RI documentaire et bibliographique est mis en évidence.

Les travaux effectués dans cette thèse reposent sur un modèle de représentation de l'information générique inspiré du Web sémantique et plus particulièrement du modèle **RDF**. Ce dernier, ainsi que les concepts entourant le Web sémantique sont traités dans le chapitre 4 p. 89.

Les preuves de concept que j'ai développé dans le cadre de cette thèse ont permis de mettre en évidence les limites des SGBDRs dans le cadre de la RI au sein de données cliniques. Dans le chapitre 4 p. 89, un état de l'art des bases de données alternatives est effectué afin d'expliquer dans quelle mesure ces dernières peuvent être, ou ne pas être, utiles à la problématique de RI au sein d'un EDS.

Le modèle de données générique du **D2IM** utilisé dans le cadre de mes travaux est concrètement décrit dans le chapitre 5 p. 111 de même que la modélisation et l'intégration des données cliniques au sein de ce modèle et dans les différents SGBDs (relationnel et NoSQL) sur lesquels reposent respectivement le SSE_{SQL} et le SSE_{NoSQL}.

Le cœur de mes réalisations est ensuite abordé à partir du chapitre 6 p. 127. Je donne, ainsi, une description concrète et accompagnée d'exemples du langage de requête spécifique que j'ai défini. La grammaire formelle permettant d'engendrer ce langage y est également explicitée de manière rigoureuse.

Dans le chapitre 7 p. 169, je précise la manière dont j'ai implémenté les preuves de concepts afin qu'elles puissent d'une part, s'intégrer au SI du **D2IM** et d'autre part, exécuter des requêtes écrites dans le langage précédent.

Enfin, le chapitre 8 p. 199 décrit l'intégration de mes travaux dans le cadre de l'EDS du CHU de Rouen. Une étude menée en 2018 visant à évaluer la capacité de l'EDS dans son ensemble à répondre à des critères d'inclusion et d'exclusion d'études cliniques y est notamment présentée.

Chapitre 2

Les données cliniques : nature, structure et enjeux

Sommaire

2.1 Le Dossier Patient Informatisé (DPI) 44
2.1.1 Les enjeux 45
2.1.2 Les défis 46
2.1.3 Les standards de représentation existants 47
2.1.4 La structure des données du DPI 51
2.2 Les Entrepôts de Données de Santé (EDS) 56
2.2.1 La problématique des données massives (Big Data) 56
2.2.2 Les objectifs de l'EDS 57
2.2.3 Panorama des EDSs existants 58

Le **D**ossier **P**atient **I**nformatisé (DPI), et dans un contexte plus générique et ambitieux, les EDSs, constituent deux éléments des **S**ystèmes d'**I**nformation **H**ospitaliers (SIHs) au sein desquels sont regroupées les informations cliniques relatives aux divers patients et celles des établissements de santé. Ces deux derniers sont donc au centre de multiples problématiques. Si le DPI a été initialement conçu dans un objectif de soins, les EDSs quant à eux, sont destinés à permettre un usage secondaire des données de Santé dans des contextes variés. Les données que contiennent ces derniers sont donc au centre de multiples problématiques de RI en Santé (e.g. recherche d'informations cliniques, acquisition de données agrégées, données massives, optimisation du PMSI, etc.).

La structure des données cliniques dérive majoritairement du processus de prise en charge des patients au sein des établissements de santé. Celle-ci recouvre ainsi intrinsèquement une information de valeur pour les divers professionnels de santé susceptibles de les exploiter. Dans le cadre de la RI, il est par conséquent indispensable de trouver une modélisation adéquate de ces données ainsi qu'une solution de stockage propice à un requêtage cohérent de ces dernières.

Dans ce chapitre, une définition du DPI et des EDSs sera donnée. La particularité structurelle des données qu'ils agrègent est également abordée ainsi que leurs différents enjeux et défis.

2.1 Le Dossier Patient Informatisé (DPI)

Comme précisé dans [38], il est actuellement difficile de donner une définition précise du **D**ossier **P**atient **I**nformatisé (DPI). Beaucoup de bases de données destinées aux DPIs ont vu le jour ces dernières années avec des objectifs, des modes d'utilisation et des visions différentes. Pour illustrer cette diversité, on peut d'ailleurs trouver dans la littérature trois grandes classes d'organisation de l'information au sein des DPIs :

Les DPI orientés source : ils organisent les informations relatives aux patients en fonction de la manière dont elles ont été obtenues ;

Les DPI orientés problème : ils agrègent les informations pour chaque problème du patient et chaque problème est détaillé selon le plan classique :

$$\text{Anamnèse} \rightarrow \text{Statut} \rightarrow \text{Synthèse} \rightarrow \text{Traitement}$$

Les DPI orientés chronologie : ils représentent les informations dans l'ordre chronologique où elles sont apparues.

Aujourd'hui, les DPIs « mêlent » ces trois types d'organisations de l'information.

En France, on peut trouver une définition du DPI dans la loi du 4 mars 2002 (davantage de précisions sur son contenu sont également données dans le décret du 29 avril 2002) :

> **Définition 1 (extrait de la loi du 4 mars 2002) :**
> *L'ensemble des informations concernant sa santé détenues par des professionnels et établissements de santé, qui sont formalisées et ont contribué à l'élaboration et au suivi du diagnostic et du traitement ou d'une action de prévention, ou ont fait l'objet d'échanges écrits entre professionnels de santé, notamment des résultats d'examen, comptes rendus de consultation, d'intervention, d'exploration ou d'hospitalisation, des protocoles et prescriptions thérapeutiques mis en œuvre, feuilles de surveillance, correspondances entre professionnels de santé, à l'exception des informations mentionnant qu'elles ont été recueillies auprès de tiers n'intervenant pas dans la prise en charge thérapeutique ou concernant un tel tiers.*

Le concept même de DPI est très général bien que ce terme soit aujourd'hui largement employé dans la littérature. En France, ce dernier n'impose pas nécessairement une notion de **partage sécurisé de l'information** contrairement au **D**ossier **M**édical **P**artagé (DMP).

Le DMP est un projet mis en place par l'**A**gence de **S**ystèmes d'**I**nformation **P**artagés de **Santé** (ASIP Santé) dans le cadre de la loi n° 2004–810 du 13 août 2004 relative à l'assurance maladie. Le DMP a pour but premier d'améliorer la coordination des soins par l'intermédiaire d'un partage des données de santé entre les divers professionnels de santé de manière sécurisée. À la différence du DPI, le patient est directement impliqué dans la gestion de ce dossier partagé car :

- il en est le propriétaire ;
- il doit donner son consentement pour sa création ;
- il en gère lui même les droits d'accès pour chaque professionnels de santé (e.g. médecins, infirmiers, pharmaciens, biologistes etc.) ;
- il peut y ajouter des informations concernant sa propre santé.

Le DMP contient donc des données qui sont de nature différente de celles du DPI puisque les DPIs sont maintenus par des institutions de santé alors que le patient lui même renseigne des informations sur sa propre santé au sein de son DMP. Néanmoins, dans une vision plus large et ambitieuse, le DPI est parfois vu comme une notion à l'intersection du DPI et du DMP

permettant d'agréger des données de santé provenant de sources variées et de les partager de manière sécurisée entre des professionnels de santé préalablement autorisés.

Le terme d'**E**lectronic **H**ealth **R**ecords (EHR), largement employé dans la littérature internationale pour désigner les **D**ossiers **P**atients **I**nformatisés, fait l'objet d'une confusion similaire. Bien qu'en toute rigueur l'Organisation internationale de normalisation (**I**nternational **O**rganization for **S**tandardization (ISO)) définit le concept d'EHR comme un système partagé [39] semblable à la notion de partage du DMP français, il est régulièrement employé pour désigner des SIs basés sur le DPI n'incluant pas nécessairement cette notion de partage. Néanmoins, la notion de **P**ersonal **H**ealth **R**ecords (PHR) existe par ailleurs et est équivalente à celle du DMP français.

Définition 2 (Définition d'un EHR selon l'Organisation Internationale de Normalisation [39]) :
« repository of information regarding the health status of a subject of care, in computer processable form, stored and transmitted securely and accessible by multiple authorized users, having a standardized or commonly agreed logical information model that is independent of EHR systems and whose primary purpose is the support of continuing, efficient and quality integrated health care. It contains information which is retrospective, concurrent and prospective. »
Dépôt d'informations relatives à la santé d'un sujet de soins sous forme digitale, stockées et transmises de manière sécurisée et accessible par de multiples utilisateurs autorisés. Il repose sur un modèle d'information logique normalisé ou communément admis qui est indépendant des systèmes de DPI. L'objectif principal de ce DPI est de supporter la continuité, l'efficacité et la qualité des soins. Il contient des informations à la fois rétrospectives, actuelles et prospectives.

L'ISO précise également tout un ensemble de termes régulièrement utilisés pour désigner les différents types de DPI et précise qu'il existe encore aujourd'hui un grand nombre de DPIs qui ne sont pas totalement conformes à cette définition notamment en ce qui concerne les dépôts de données cliniques qui centralisent et gèrent les données cliniques recueillies dans différents points de services.

En somme, le DPI peut être vu comme la version électronique du traditionnel dossier patient papier utilisé par les professionnels de santé [6–8]. Bien qu'il soit encore aujourd'hui difficile de donner une définition précise du DPI, et que le champ lexical du concept reste encore mal défini, l'idée sous jacente à ces systèmes depuis les années 90 a toujours été d'agréger et de centraliser l'ensemble des informations relatives à la santé des patients à la fois de manière transversale et longitudinale [40]. Le DPI, regroupe ainsi un ensemble d'informations provenant potentiellement d'établissements et de professionnels de santé différents et collecté sur l'ensemble de son histoire médicale.

2.1.1 Les enjeux

Les données du DPI contiennent des informations sur la santé d'un patient d'une manière « générale ». Ces données servent donc des objectifs qui dépassent le simple établissement d'une prise en charge (diagnostic et traitement principalement) pour un patient donné bien qu'il s'agisse de l'objectif premier du DPI. Elles sont d'ailleurs aujourd'hui exploitées à la fois dans divers contextes (i.e. soins primaires, secondaires et tertiaires) et par des personnes de profils différents (i.e. médecins, infirmiers, radiologistes, pharmaciens, techniciens de laboratoires, personnel administratif, etc.) [38]. Le patient lui-même peut être amené à exploiter, à accéder et à compléter le DPI notamment dans le cadre du DMP [41], ou du futur Espace Numérique de Santé [42]. Le DPI offre une couverture exhaustive de l'information de santé ce qui le place au centre de nombreux enjeux pour de multiples acteurs différents :

Enjeux de Santé : L'un des enjeux le plus évident est l'amélioration de la qualité des soins dans son ensemble. L'exploitation efficace du DPI permet un meilleur échange des informations entre les différents professionnels de santé et favorise notamment une aide aux décisions cliniques et une réduction des erreurs médicales.

Enjeux sociétaux : Le DPI représente des avantages sociétaux en termes de satisfaction du personnel médical mais aussi en terme de recherche clinique [43]. Néanmoins, si l'informatisation n'est pas réalisée dans les meilleures conditions d'ergonomie et d'organisation, les rejets restent possibles. Ceci a notamment pu être constaté dans le cadre d'une étude « avant-après » réalisée au CHU de Rouen, lors de l'arrêt de la prescription informatisée [44]. La disponibilité de données électroniques agrégées offre en effet à la fois un plus large panel d'informations et une plus grande liberté d'accès aux données médicales qui facilite la recherche clinique notamment dans le cadre de la constitution de cohortes de patients.

Enjeux financiers : L'implémentation de systèmes appropriés d'accès au DPI représente un potentiel d'économie important en terme d'organisation, de temps et de réduction de perte financière [45].

2.1.2 Les défis

Bien que les établissements de santé aient fait d'importants progrès en ce qui concerne la collecte des données de leurs patients au sein de DPIs, l'exploitation de ces données reste néanmoins limitée. L'implémentation de SRIs ou plus généralement de SIHs destinés à l'exploitation de ces dernières a été en effet beaucoup plus lente, notamment en terme de réelle interopérabilité [46]. Un ensemble de standards à différents niveaux reste encore à définir et/ou à adopter pour pleinement tirer parti du DPI [47]. De nombreux défis cliniques, éthiques et techniques dépendent de l'établissement et de l'utilisation de ces standards pour pleinement satisfaire l'ambition donnée internationalement au DPI :

L'interopérabilité des SIs : Pour assurer une prise en charge efficace et sûre de leurs patients, les professionnels de santé ont besoin d'accéder à des informations à la fois exhaustives et détaillées concernant la santé de leurs patients. Le partage des informations entre les différents acteurs de santé, et donc l'interopérabilité des systèmes, est alors primordiale pour une prise en charge de qualité [48, 49]. L'adoption de standards destinés à l'uniformisation du contenu et de la structure des données patient reste néanmoins encore aujourd'hui lente (cf. sous-section 2.1.3 p. 47). Différents facteurs peuvent expliquer cette situation. Comme évoqué précédemment, les données que contiennent les divers DPIs actuels montrent une grande diversité en terme de structure comme en terme de nature. Cela implique une hétérogénéité des modèles de données. La communication de données patient est structurellement complexe dans le sens où l'information clinique n'est pas issue uniquement de l'interprétation des données isolées propres au patient en question mais aussi de l'association et de la composition de ces données entre elles. La définition de standards de formalisation, d'accès et d'échanges de ces données requiert par conséquent une abstraction permettant de représenter, de manière exhaustive, la diversité de l'information clinique. Cette formalisation du DPI n'est pas uniquement un frein au développement purement informatique de SI générique d'accès au DPI. Elle constitue également, un frein au développement d'algorithmes génériques dans le domaine de l'aide à la décision médicale par exemple.

Sécurité et confidentialité : La nature « sensible » des données nominatives des patients rend l'accès aux données juridiquement problématique. Peu de corpus anonymisés de données patient ou encore de comptes-rendus annotés et librement exploitables sont actuellement disponibles notamment en France et en Europe. Á titre d'exemple, notre équipe a tenté, sans succès, de créer un corpus anonymisé à partir du projet **L**ecture **R**apide en **U**rgence du **D**ossier **I**nformatique du patient (LERUDI) financé par l'ASIP Santé. En revanche, certains corpus existent en provenance des État-Unis permettant la mise en place de compétitions en indexation automatique voire en recherche d'information. À noter que les

règles de déidentification sont encore plus sévères aux État-Unis par rapport à la France, ce qui est paradoxal.

Plus généralement, l'accès à des données nominatives y compris par les professionnels de santé issus de l'établissement dont proviennent les données patient est très encadré juridiquement. Ces verrous juridiques constituent un frein majeur à la recherche dans le champ de la RI sur des données patient. L'établissement de standards d'authentification, d'intégrité et de cryptage des données sont par conséquent nécessaires.

Dans le cadre de ce travail de thèse, j'ai notamment integré un module ad hoc basique de déidentification des textes cliniques permettant de retirer les éléments identifiant de ces derniers (e.g. noms, prenoms, dates de naissances, etc.). Ce dernier a été utilisé dans l'étude décrite dans le chapitre 8 p. 199.

Interopérabilité sémantique : L'interopérabilité sémantique permet à des données échangées entre des systèmes interconnectés de préserver leurs sens. Elle a pour objectif de permettre l'interprétation des données en terme d'information et/ou de connaissance de manière cohérente, c'est à dire, en accord avec l'information exprimée originellement. Pour assurer cette interopérabilité sémantique, un modèle conceptuel d'information doit être partagé entre la source de l'information et la cible de cette dernière. Dans le cadre de la médecine, elle est particulièrement importante et complexe. Elle implique la prise en compte de la variabilité des pratiques cliniques et des domaines de la médecine ainsi que leurs évolutions en fonction des connaissances médicales et du temps. Une approche systématique de nommage et de représentations rigoureuses et hiérarchiques de l'information sous-jacente aux données du DPI est donc nécessaire pour rendre son exploitation cohérente. Cette interopérabilité sémantique passe notamment par la standardisation des terminologies actuellement utilisées pour coder l'information clinique. En France, la **C**lassification **C**ommune des **A**ctes **M**édicaux (CCAM) [50] est utilisée pour coder les actes médicaux tandis que la **C**lassification **I**nternationale des **M**aladies (**10ème** révision) (CIM–10) [51] sert au codage des motifs d'entrée des séjours (cf. sous-sous-section 2.1.4.2 p. 53). Ces deux classifications ont, dans le contexte du PMSI, un objectif médico-économique, alors qu'historiquement elles ont été inventées pour des raisons épidémiologiques. Ainsi, en raison de sa couverture terminologique insuffisante, la CIM–10 ne peut pas être considérée comme une terminologie de référence pour le codage de tous les diagnostics. En somme, il n'existe aujourd'hui pas réellement de terminologies et de « langages » de référence propres à la gestion des données patient qui fassent totalement consensus. Celle qui s'en rapproche le plus est sans doute la **S**ystematized **N**omenclature **O**f **MED**icine–**C**linical **T**erms (SNOMED–CT®) [52], qui n'est que partiellement traduite en français. La France n'a toujours pas décidé, en 2019, d'être membre du consortium SNOMED International, contrairement à la Belgique, le Canada ou encore la Suisse (trois pays partiellement francophones).

Évaluation des SIs : Les entrepôts de données patient possèdent une volumétrie de données très importante souvent proche des problématiques de traitement de données massives (Big Data). Dans un contexte de santé, des temps d'exécution acceptables par les professionnels de santé doivent être définis afin de rendre les SRIs d'accès au DPI utile. Contrairement à la RI documentaire où de nombreuses métriques existent pour mesurer la performance et la qualité de ces SIs ou plus précisément des moteurs de recherche, l'évaluation des SRIs dédiés au DPI reste, a contrario, moins standardisée et plus subjective du fait de la complexité du DPI.

2.1.3 Les standards de représentation existants

Jusqu'à présent, les principaux efforts d'interopérabilité réalisés sur les SIs d'accès au DPI ont consisté en l'envoi de messages électroniques prédéfinis transmis à l'aide de normes d'échange telles que **H**ealth **L**evel **Seven** (HL7) ou encore **E**lectronic **D**ata **I**nterchange **F**or **A**dministration, **C**ommerce and **T**ransport (EDIFACT). Cependant, ces efforts ont majoritairement été faits à des fins commerciales, administratives ou organisationnelles (communications acheteurs/vendeurs, facturations, gestion des agendas, etc.) et peu d'efforts d'interopérabilité axés sur le partage de

données cliniques ont été réalisés [46]. Néanmoins, plusieurs projets destinés à la définition de standards pour le DPI ont été initiés et plusieurs normes en sont issues.

2.1.3.1 La norme EHRcom (CEN/ISO 13606)

La norme EN 13606 est une norme initialement produite par le **C**omité **E**uropéen de **N**ormalisation (CEN) au sein du groupe de travail **E**lectronic **H**ealth **R**ecord **COM**munication (EHRcom). C'est aujourd'hui un standard ISO composé de cinq parties. Cette norme a été utilisée pour construire le DPI du CHU de Rouen. On trouve dans la préface de la partie 1 de ce standard une description générale de l'objectif de la norme [53] :

> **Définition 3 (ISO 13606) :**
> *L'objet général de l'ISO 13606 est de définir une architecture d'information rigoureuse et durable destinée à communiquer tout ou partie du Dossier Informatisé de Santé (DIS) d'un seul sujet de soins (patient) afin de permettre l'interopérabilité des systèmes et composants nécessitant de transmettre (accès, transfert, ajout ou modification) les entrées du dossier DIS via des messages électroniques ou des objets répartis :*
>
> — *en préservant le sens médical original que l'auteur a voulu donner ;*
>
> — *en reflétant la confidentialité des données voulues par l'auteur et le patient.*

L'approche adoptée par l'ISO 13606 est basée sur un modèle double :

Un modèle de référence : Il permet de répondre aux problématiques générales d'interopérabilité et confidentialité des données. Il définit en somme la structure globale du DPI. Ce modèle est décrit de la manière suivante dans [53] : ce modèle « *représente les caractéristiques globales des entrées des dossiers de santé, leur mode d'agrégation, et les informations contextuelles nécessaires pour satisfaire aux exigences relatives à l'éthique, au droit et à la provenance. Ce modèle définit un ensemble de classes formant les briques de base génériques du DIS.* ». La figure 2.1 précise les principales classes de ce modèle et les relations structurant le DPI qu'elles entretiennent. L'ISO 13606 structure l'information d'un DPI (ou d'un extrait de ce dernier) en la répartissant au sein de la hiérarchie de classe suivante : FOLDER → COMPOSITION → ENTRY → ELEMENT. Cette hiérarchie permet de rendre compte à la fois du « contexte clinique » de l'information et des données relatives à l'information elle même. Le degré d'information contextuelle étant décroissant à mesure que l'on descend au sein de cette hiérarchie globale. Les classes FOLDER, SECTION et CLUSTER sont purement destinées à l'organisation et au regroupement des informations afin de les rendre cliniquement cohérentes. Chaque classe est détaillée de manière précise au sein de l'ISO 13606 :

FOLDER et COMPOSITION : Le DPI est organisé en COMPOSITIONS, éventuellement regroupées selon une hiérarchie FOLDER. Un FOLDER étant relatif aux soins dispensés pour une seule affection par une équipe ou un établissement de santé, ou au cours d'une période de temps définie (e.g. Traitement du diabète, Pédiatrie, Hôpital Saint Michel, Épisodes 2000-2001, Italie, etc.). Les COMPOSITIONs correspondent à l'ensemble des informations consignées dans un DPI par un agent, suite à une seule rencontre médicale ou séance de documentation de dossier (e.g. Note d'évolution, formulaire de résultat d'essai de laboratoire, compte-rendu radiologique, consultation, lettre d'adressage, bilan de diabète etc.).

COMPOSITION et ENTRY : Les COMPOSITIONS sont composées d'ENTRIES. Une ENTRY contient les informations enregistrées suite à un acte médical, une observation, une interprétation médicale, ou une intention. Une ENTRY correspond aux « données de consultation » (e.g. un symptôme, un résultat d'essai, un médicament prescrit, un diagnostic différentiel, formule leucocytaire, mesure de la pression artérielle etc.). Les ENTRIES peuvent être regroupées au sein d'une hiérarchie de SECTIONS. Une SECTION correspond alors à une en-tête médicale et reflétant généralement le flux d'informations collectées au cours d'une rencontre médicale, ou

structurées pour en faciliter la lecture ultérieure par l'humain (e.g. Motif de la rencontre, Antécédents, symptômes subjectifs, Analyse, Plan, Régime, Examen abdominal etc.)

ENTRY et ELEMENTS : Un ELEMENT constitue la ramification distale de la hiérarchie de DPI, contenant une seule valeur de données (e.g. pouls, nom du médicament, symptôme, poids corporel. Un CLUSTER permet d'organiser des structures de données multiples emboîtées telle une série chronologique, et de représenter les colonnes d'un tableau (e.g. Résultats d'audiogramme, interprétations d'électroencéphalogramme, diagnostics différentiels comparatifs)

Un archétype : Il assure une interopérabilité sémantique. Il est décrit de la manière suivante dans [53] : des « *métadonnées servant à représenter les caractéristiques spécifiques de différentes catégories de données cliniques, susceptibles de nécessiter une représentation pour répondre aux exigences de chaque profession, spécialité ou service particulier.* ». De manière plus formelle, c'est un modèle qui « *représente la définition formelle de combinaisons préétablies de classes considérées comme briques de base définies dans le Modèle de référence pour des domaines ou des organisations médicaux particuliers. Un archétype représente l'expression formelle d'un concept de domaine distinct, exprimé sous forme de contraintes imposées aux données dont les instances se conforment au modèle de référence.* ».

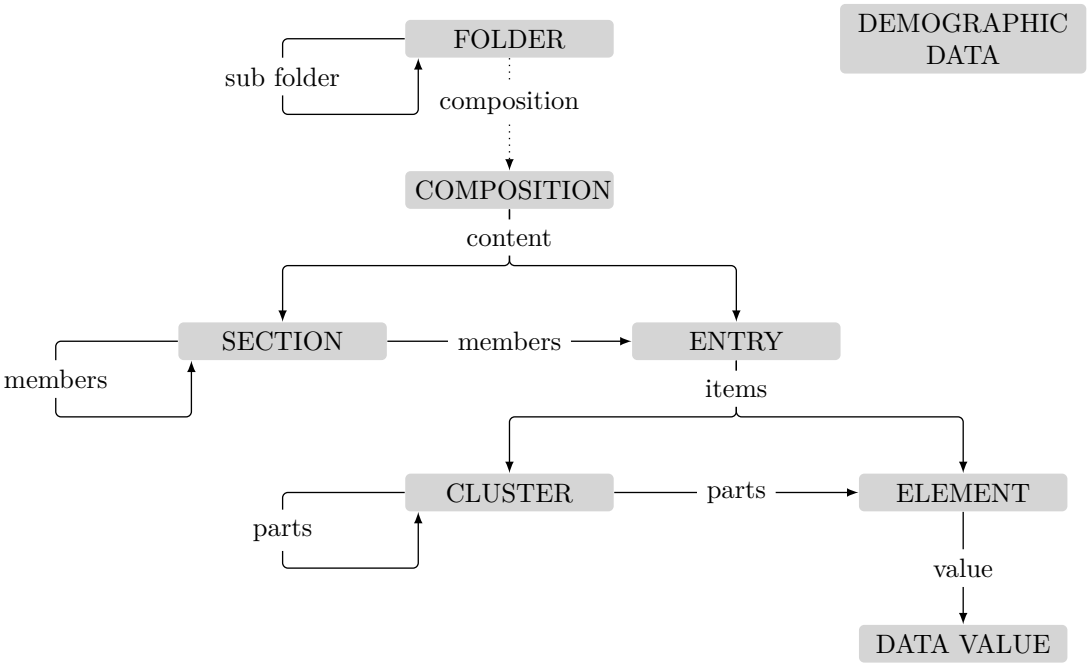


FIGURE 2.1 – Principales classes définies par le modèle de référence du standard ISO 13606.
Source : *http://veratechnas1.synology.me:6969/iso13606/information.html*

2.1.3.2 Les standards Health Level Seven

HL7 est une organisation fondée en 1987 dont le but est le développement d'une famille de normes et de spécification pour l'échange, l'intégration et le partage informatisé de données cliniques, financières et administratives entre SIHs. Les standards HL7 prennent la forme de normes de messages destinés à l'échange d'informations cliniques entre applications. De nombreuses normes HL7 sont aujourd'hui reconnues officiellement à la fois comme des standards formels américains par l'**A**merican **N**ational **S**tandards **I**nstitute (ANSI) mais également comme des standards internationaux par l'ISO. La publication initiale de la version 3 des standards HL7 (HL7 version 3 (HL7 v3)) a eu lieu en 2005. Cette version est basée sur une nouvelle méthodologie formelle (**HL7 D**evelopment **F**ramework (HDF) (ISO/HL7 27931)) et sur une philosophie orientée objet.

HL7 Reference Information Model (HL7–RIM) (ISO/HL7 21731**) :** La pierre angulaire de la famille de standards définis par la version 3 de HL7 est le standard conceptuel HL7–RIM. C'est en réalité la racine des tous les modèles d'information développés dans le cadre de cette version et elle constitue une partie importante de la méthodologie employée pour cette dernière (HDF). HL7–RIM fournit un modèle d'information formel définissant les classes et attributs majeurs sur lesquels reposent les différents messages HL7 et exprime donc par extension les besoins en terme d'information dans les différents contextes cliniques ou administratifs.

HL7 Clinical Document Architecture (HL7–CDA ®) (ISO/HL7 27932**) :** HL7 fournit le standard HL7–CDA ® qui définit une structure générique de messages pour l'échange de documents cliniques. C'est un standard basé sur le métalangage XML qui spécifie l'encodage et la structure des « documents cliniques » (e.g. résumés de sortie, images, rapports de pathologie etc.) dans un but d'échanges de ces derniers entre les professionnels de santé et les patients. Il spécifie également le but de ces échanges. La version 2 de HL7–CDA ® se conforme à la version 3 de HL7. Elle est basée sur HL7–RIM et exploite les types de données de HL7 v3. Cette norme permet notamment de standardiser l'échange des divers comptes-rendus et permet de modéliser les documents cliniques en prenant en compte les six caractéristiques majeures suivantes :

1. La persistance : la disponibilité du document pour une longue période dans le temps ;
2. La gestion : la gestion du document par des établissements de confiance (e.g. hôpitaux utilisant HL7–CDA ®) ;
3. Le potentiel d'authentification : la garantie de la qualité des données ;
4. Le contexte : la définition d'un contexte par défaut (e.g. identité du patient, auteur du document) ;
5. La complétude : la globalité du document (avec une éventuelle authentification) et non pas seulement une partie de ce dernier ;
6. L'accessibilité : le document peut être visualisé via différents outils (e.g. navigateurs web, smartphones, tablettes).

HL7–CDA ® offre la possibilité de gérer également les données textuelles non structurées ainsi que les liens vers des documents textuels externes (e.g. pdf, docx, rtf) ou des images (e.g. jpg, png). La gestion des données structurées repose sur l'exploitation de systèmes de codage comme SNOMED–CT® ou **L**ogical **O**bservation **I**dentifiers **N**ames and **C**odes (LOINC).

HL7 Fast Healthcare Interoperability Resources (HL7–FHIR ®) : HL7–FHIR ® est un standard d'échange électronique d'informations de santé. Ce standard définit un ensemble de formats de données et d'éléments appelés « Resources » représentant des concepts cliniques de manière granulaire. Contrairement à la norme HL7–CDA ®, les ressources peuvent être manipulées et transmises au choix de manière isolée ou de manière agrégée pour former un document complexe. Des informations de santé « atomiques » (e.g. patients, diagnostic, prescriptions etc.) peuvent donc être échangées séparément par l'intermédiaire de leurs propres **U**niform **R**esource **L**ocators (URLs). La norme HL7–FHIR ® permet de modéliser des données en dehors du scope clinique (e.g. informations financière) alors que que la norme HL7–CDA ® est théoriquement limitée au cas d'utilisation clinique. Bien que HL7–FHIR ® repose sur les formats de données standards HL7, sa conception à été réalisée avec un souci de simplicité de son implémentation sans sacrifice de l'intégrité des données. Ainsi, les ressources sont basées sur de simples structures de données XML ou **J**ava**S**cript **O**bject **N**otation (JSON ) et leurs échanges reposent sur un protocole **H**ypertext **T**ransfer **P**rotocol (HTTP) suivant le principe d'architecture REST (viz.RESTful).

2.1.4 La structure des données du DPI

2.1.4.1 Une structure complexe

En France, l'article R1112–2 du code de Santé Publique précise un certain nombre d'éléments que contient le dossier médical et qui sont synthétisés dans la définition 4 :

Définition 4 (Contenu du dossier médical : article R1112–2 du code de Santé Publique) :

Documents établis durant le séjour :

- *lettre d'admission ;*
- *l'évaluation clinique initiale (motif d'hospitalisation, anamnèse, examen clinique, conclusion et prescriptions initiales) ;*
- *les informations relatives à la prise en charge : évaluation clinique, prescriptions, transfusions, soins reçus (quels que soient les professionnels les dispensant), examens complémentaires ;*
- *le ou les comptes-rendus opératoires ou d'accouchement ;*
- *le dossier d'anesthésie ;*
- *le dossier de soins infirmiers ;*
- *les différents consentements et les directives anticipées ;*
- *les correspondances échangées entre professionnels de santé.*

Documents établis à la fin du séjour :

- *le compte-rendu d'hospitalisation ;*
- *les prescriptions établies à la sortie du patient ;*
- *la fiche de liaison infirmière.*

source : *http://www.chu-rouen.fr/cismef/wp/wp-content/uploads/2019/01/Le-Dossier-M%C3%A9dical.pdf*

Dans cette définition, le dossier médical est vu comme un dossier regroupant l'ensemble des documents utiles aux professionnels de santé dans le cadre de la prodigation des soins aux patients. Dans un contexte de RI, une analyse plus rigoureuse de la nature de ces données est cependant nécessaire. La caractéristique première de cet ensemble de données est leur hétérogénéité. Celle-ci apparaît à deux niveaux :

Hétérogénéité des types d'information : Les données du DPI sont ainsi également produites lors du processus de prise en charge des patients et donc, par des professionnels de santé issus de spécialités médicales différentes. On retrouve au sein du DPI des informations remplissant des objectifs variés comme par exemple des informations relatives aux examens biologiques, de l'imagerie, des courriers destinés à la communication entre professionnels de santé, des observations médicales, des prescriptions, etc. Ces informations sont disséminées à travers différents documents issus de multiples services et hôpitaux et sont établis à des moments différents de la prise en charge des patients. Comme évoqué précédemment, l'organisation de ces documents et informations est difficile à standardiser de manière cohérente malgré les efforts effectués en ce sens. Si la multiplicité des sources dont est issue l'information du DPI complexifie la colligation technique et logique de ces dernières, leur hétérogénéité rend également leur modélisation conceptuelle ardue. Ces informations font ainsi intervenir de multiples éléments d'information entretenant entre eux des relations lourdes de sens d'un point de vue médical.

Hétérogénéité des nature de données : Les données cliniques sont fournies en grande partie à travers des blocs de textes tels que les comptes-rendus. En France, le consensus

d'experts estime que 80% des données du DPI sont non structurées. La médecine repose traditionnellement sur une transmission des informations à travers des documents textuels (cf. chapitre 3 p. 69). Cependant, les données cliniques incluent également des données structurées qui, bien que plus minoritaires, ont un rôle informatif important. On peut trouver notamment parmi celles-ci :

- les données relatives aux patients (e.g. age, sexe, etc.) ;
- les données numériques des examens biologiques ;
- les données organisationnelles et administratives des séjours, hospitalisations et prises en charge incluant leurs dates, leurs types, les unités fonctionnelles dans lesquelles ils ont été effectués etc. ;
- les informations du PMSI ;

Les informations du PMSI servent en premier lieu des objectifs de facturation pour les établissements de santé. Elles permettent néanmoins, dans un usage secondaire, de définir des informations cliniques majeures de manière structurée telles que les motifs d'hospitalisations et/ou les diagnostics des patients ainsi que les actes médicaux et chirurgicaux qu'ils ont subi. Cette définition passe par l'exploitation de classifications (qui sont décrites plus précisément dans la section suivante).

Le DPI inclut également des données complexes telles que des données d'imagerie (e.g. radiologies, échographies, tomodensitométries, Imageries par Résonance Magnétique (IRM), anatomopathologies, dermatologies, etc.) ainsi que des données sous forme de signaux (électrocardiographies (ECG), électroencéphalographies (EEG), électrorétinogrammes (ERG), etc.). Ces données ne seront cependant pas considérées dans le cadre de cette thèse ni même dans le contexte plus général de l'EDSS du CHU de Rouen.

En somme, les données du DPI peuvent être réparties en quatre catégories cohérentes dans le cadre d'une vision clinique (Figure 2.2). Dans le cadre de cette thèse, le niveau établissement n'a pas été considéré compte tenu que toutes les données exploitées proviennent du CHU de Rouen.

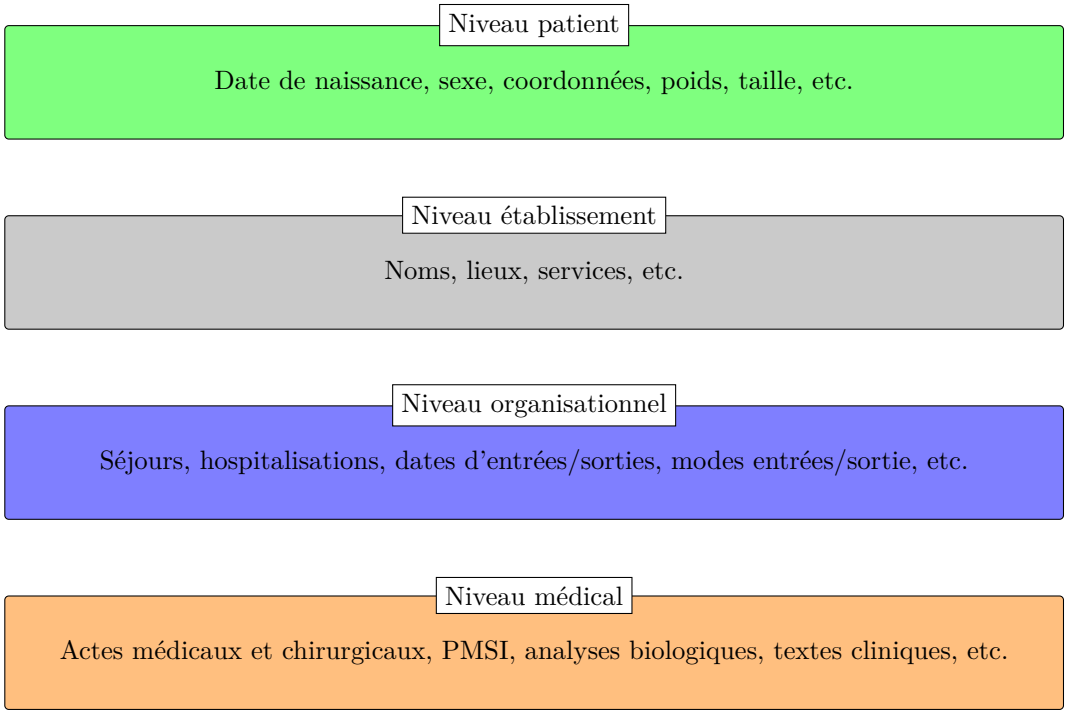


FIGURE 2.2 – Répartition des données du DPI en quatre catégories.

2.1.4.2 Les terminologies utilisées pour le DPI

Depuis 1991, les établissements de Santé français ont l'obligation de procéder à l'évaluation et à l'analyse de leur activité dans le cadre du **P**rogramme de **M**édicalisation des **S**ystèmes d'**I**nformation (PMSI). Ce dernier constitue l'un des dispositifs de la réforme du système de santé français initié dans les débuts des années 1980. Depuis 2005, le PMSI sert en outre de base au calcul de la rémunération des hôpitaux dans le cadre de la **T**arification à l'**A**ctivité (T2A). Ceci fait du PMSI un point stratégique pour les divers établissements de santé.

La **C**lassification **I**nternationale des **M**aladies (**10ème** révision) (CIM–10) et la **C**lassification **C**ommune des **A**ctes **M**édicaux (CCAM) jouent toutes deux un rôle prépondérant dans le cadre de la réalisation du PMSI. L'**A**gence **T**echnique de l'**I**nformation sur l'**H**ospitalisation (ATIH), dont l'une des fonctions premières est la collecte des données du PMSI, impose en effet l'utilisation de ces dernières comme terminologies de codage des diagnostics et des actes médicaux.

En France, la CIM–10 et la CCAM sont ainsi deux terminologies dont l'usage est systématisé dans le processus de soin. Elles jouent par conséquent un rôle informatif fiable. Cependant, comme partiellement évoqué précédemment, elles ne permettent pas à elles seules de répondre à toutes les exigences de représentation sémantique de l'informatisation de santé.

Dans cette section, ces deux terminologies sont brièvement présentées.

2.1.4.2.1 La CIM–10

La standardisation terminologique en médecine a débuté dès les années 1880. Pendant longtemps, la CIM–10 est restée la seule ressource terminologique disponible dans cette discipline. Elle est éditée par l'**O**rganisation **M**ondiale de la **S**anté (OMS) [1] qui en donne la définition suivante :

> **Définition 5 (la CIM–10 selon l'OMS [51]) :**
> *« The purpose of the ICD is to permit the systematic recording analysis, interpretation and comparison of mortality and morbidity data collected in different countries or areas and at different times. The ICD is used to translate diagnoses of diseases and other health problems from words into an alphanumeric code, which permits easy storage, retrieval and analysis of the data. In practice, the ICD has become the international standard diagnostic classification for all general epidemiological and many health management purpose »*
> *L'objectif de la CIM[a] est de permettre l'analyse systématique, l'interprétation et la comparaison des données de mortalité et de morbidité recueillies dans différents pays ou régions et à des époques différentes. La CIM est utilisée pour traduire en codes alphanumériques les expressions en langage naturel désignant des diagnostics de maladies et d'autres problèmes. Ce codage permet un stockage, une extraction et une analyse facile des données. Dans la pratique, la CIM est devenue le standard international de classification des diagnostics dans tous les contextes épidémiologiques généraux et dans de nombreux contextes de gestion de la Santé.*
>
> *a.* **C**lassification Statistique **I**nternationale des **M**aladies et des Problèmes de Santé Connexes

La CIM–10 possède un caractère très « généraliste » dans le sens où elle permet non seulement de classer des maladies mais aussi une très vaste variété de signes, de symptômes, de lésions traumatiques, d'empoisonnements, de circonstances sociales et de causes externes de blessures ou de maladies. Plus généralement, elle est considérée comme englobant toutes les affections reconnues par la profession médicale [54].

1. url : `https://www.who.int/fr`

La CIM–10 fournit plus de 155 000 concepts terminologiques, ou codes, en 43 langues différentes incluant le Français. Structurellement, la CIM–10 établit un regroupement de ces codes en blocs eux-mêmes regroupés en 22 chapitres. Les blocs et les chapitres correspondent ainsi davantage à des intervalles de codes qu'à des codes à proprement parler même si un libellé leur est attribué. Cependant, une hiérarchisation en catégories, sous-catégories puis sous-division de ces concepts est possible. Pour illustrer ce propos, un extrait de cette classification est donné dans l'exemple 1.

Exemple 1 :
Extrait de l'arborescence CIM–10 relatif au code « fracture de la voûte du crâne, fracture ouverte » [*S*0201 (*CIM–10*)] *:*

« lésions traumatiques, empoisonnements [...] » [*S*00 – *T*98 (*CIM–10*)] ***(Chapitre)***
  « lésions traumatiques de la tête » [*S*00 – *S*09 (*CIM–10*)] ***(Bloc)***
    « fracture du crâne et des os de la face » [*S*02 (*CIM–10*)] ***(Catégorie)***
      « fracture [...] du crâne » [*S*020 (*CIM–10*)] ***(Sous-catégorie)***
        « fracture [...] du crâne, fracture ouverte » [*S*0201 (*CIM–10*)] ***(Sous-division)***

Le PMSI, à travers la CIM–10, constitue une source d'information clinique structurée fiable. En plus d'être structurée, cette information présente l'avantage d'être standardisée, sûre et systématiquement renseignée à l'échelle de la France puisque les hôpitaux y sont contraints par la loi et que leur santé financière en dépend. Cependant, la CIM–10 a, d'une part, initialement été conçue à des fins épidémiologiques, et est, d'autre part, aujourd'hui utilisée à des fins médico-économiques. Il convient donc de relativiser l'exhaustivité de cette terminologie du point de vue d'un EDS car elle ne permet pas une expression pleine et entière de la sémantique de l'information clinique, y compris en ce qui concerne les diagnostics.

2.1.4.2.2 La CCAM

La CCAM est l'outil de référence pour la description et le codage des actes techniques médicaux effectués dans les cabinets médicaux libéraux et réalisés par les médecins dans les établissements de santé publics et privés français. Tout comme la CIM–10, elle est utilisée dans le cadre du PMSI pour mesurer l'activité médicale et sert de base au calcul de leur rémunération. La CCAM a été mise en application au milieu des années 2000 en remplacement d'autres systèmes de codage tels que le **C**atalogue **d**es **A**ctes **M**édicaux (CdAM) ou encore la **N**omenclature **G**énérale des **A**ctes **P**rofessionnels (NGAP).

La CCAM est une nomenclature française. Les codes de cette dernière permettent de représenter aussi bien des gestes techniques que des actes intellectuels cliniques et sont composés de quatre lettres suivis d'un identifiant de trois chiffres. Un code permet ainsi de décrire un acte en fonction de :

— l'appareil anatomique considéré (1ère lettre) ;
— l'organe ou la fonction considéré (2ème lettre) ;
— l'action réalisée (3ème lettre) ;
— la voie ou la technique utilisée (4ème lettre).

Les codes CCAM sont munis de libellés et sont organisés/hiérarchisés au sein d'une arborescence générale comportant à la racine 19 chapitres. Parallèlement, la CCAM fournit également des arborescences propres et/ou des libellés pour :

— les appareils anatomiques et les organes ou fonctions considérés ;
— les actions réalisées ;
— les voies ou techniques employées.

Un exemple de cette structure est donné dans l'exemple 2.

Exemple 2 :
Le positionnement du code

« Macroélectromyographie, par électrode aiguille » [*AHQB*006 (*CCAM*)]

dans l'arborescence générale de la CCAM est le suivant :

« SYSTÈME NERVEUX CENTRAL, PÉRIPHÉRIQUE [...] » [01 (*CCAM*)] ***(Chapitre)***
« ACTES DIAGNOSTIQUES SUR LE SYSTÈME NERVEUX » [01.01 (*CCAM*)] ***(Menu)***
« Explorations électrophysiologiques du système nerveux » [01.01.01 (*CCAM*)] ***(Menu)***
« Électromyographie [EMG] » [01.01.01.01 (*CCAM*)] ***(Menu)***
« Macroélectromyographie, par électrode aiguille » [*AHQB*006 (*CCAM*)] ***(Menu)***

L'appareil anatomique et l'organe visé par cet acte sont définis par le sous-code « AH » qui s'intègre au sein d'une arborescence propre :

« SYSTEME NERVEUX » [*A* (*CCAM*)]
« Nerfs spinaux (y compris la partie intrarachidienne) » [*AH* (*CCAM*)]

De même que l'action définie par le sous-code « Q » :

« ACTIONS D'OBSERVATION » [04 (*CCAM*)]
« GUIDER, ENREGISTRER, EXAMINER, MESURER » [*Q* (*CCAM*)]

Enfin la voie empruntée par cet acte est définie par le sous code « B » dont le libellé est :

« ACCÈS TRANSPARIÉTAL » [*B* (*CCAM*)]

À la différence de la CIM–10, la CCAM est très spécifique. Dans un contexte de RI purement textuelle, les libellés qu'elle contient présentent une utilité très limitée. Ces derniers sont en effet notoirement parfois très long et sont par conséquent peu susceptibles d'être saisis par un utilisateur ou d'être présents au sein de textes cliniques (e.g. « *Électromyographie de 7 muscles striés ou plus au repos et à l'effort par électrode aiguille, avec mesure des vitesses de conduction motrice et de l'amplitude des réponses musculaires de 5 nerfs ou plus avec étude de la conduction proximale par électrode de surface, et mesure des vitesses de la conduction sensitive et de l'amplitude du potentiel sensitif de 5 nerfs ou plus* » [*AHQB*033 (CCAM)]). Cependant, le codage des actes étant réalisé de manière systémique, ce dernier présente un intérêt dans le cadre d'une RI structurée visant à rechercher des informations en lien avec des actes médicaux connus à l'avance.

Le DPI est un outil informatique principalement destiné à l'amélioration de la prise en charge des patients. Il permet d'**agréger** les différents documents relatifs aux patients. Des normes principalement destinées au partage et à l'organisation de ce dossier existent mais restent souvent complexes à mettre en place et ne permettent pas nécessairement une pleine structuration de l'information clinique qu'il contient. Les EDSs, en revanche, visent à **colliger** de manière cohérente ces informations. Dans la section suivante, je définis ces derniers et dresse un état de l'art de leur utilité ainsi que des outils existants leur fournissant un accès.

2.2 Les Entrepôts de Données de Santé (EDS)

Un **E**ntrepôt de **D**onnées de **S**anté (EDS) ou **E**ntrepôt de **D**onnées Cliniques (EDC) centralise au sein d'un système de gestion de données unique, les informations cliniques et démographiques produites autour d'une large population de patients. Les données d'un EDS sont généralement collectées à partir de sources variées puis agrégées et structurées au sein d'un modèle de données uniforme et d'une architecture matérielle unique assurant ainsi une cohérence globale et la fiabilité de l'information. Un EDS inclut notamment les données du DPI mais également les données issues des différents systèmes de production tels que les systèmes d'information de laboratoires d'analyses biologiques ou d'imagerie médicale, les données des systèmes de prescription informatisés ou encore celles du dossier infirmier. Les EDCs sont définis ainsi :

Définition 6 (Définition d'un EDC selon l'Organisation Internationale de Normalisation [55]) :
« Grouping of data accessible by a single data management system, possibly of diverse sources, pertaining to a health system or sub-system and enabling secondary data analysis for questions relevant to understanding the functioning of that health system, and hence supporting proper maintenance and improvement of that health system. A Clinicla Data Warehouse (CDW) tends not to be used in real time. However, depending on the rapidity of transfer of data to the data warehouse, and data integrity, near real-time applications are not excluded. »
Regroupement de données accessibles via un unique système de gestion de données, provenant potentiellement de diverses sources et relatives aux systèmes ou sous-systèmes de santé. Ces données permettent ainsi une analyse de données secondaires pertinentes dans le cadre des questions de compréhension du fonctionnement des système de santé et en favorisent donc une maintenance adéquate. Un EDC n'est à priori pas destiné à être utilisé en temps réel. Cependant, selon la rapidité de transfert des données vers l'entrepôt de données et l'intégrité de ces données, son exploitation dans des applications proches du temps réel n'est pas exclue.

2.2.1 La problématique des données massives (Big Data)

Le volume de données produites par les diverses activités de santé est extrêmement important. À titre d'exemple, en 2011, l'ensemble des données produites par le système de santé des États-Unis représentait 150 exaoctets soit 150×10^{18} octets [56, 57]. En comparaison, une étude publiée en février 2013 estime à 850 exaoctets la quantité totale de données produites aux États-Unis en 2012 pour une projection à 6,6 zettaoctets (i.e. 6.6×10^{21} octets) d'ici 2020 [58]. Les données de santé représentent donc une part substantielle du challenge « Big Data » dans sa globalité.

Plus localement, on peut également citer une récente étude menée au CHU de Rouen [59]. Cette étude visait à récupérer les différents documents de santé (viz. comptes-rendus hospitaliers, comptes-rendus d'acte, courriers, ordonnances etc.) produits entre le mois de janvier 2000 et le mois de juillet 2017 afin de les indexer automatiquement à l'aide de plus de 40 terminologies de santé. Durant cette étude 11 928 168 documents de santé ont été extraits et plus de 5 milliards d'annotations automatiques ont été générées.

Dans un contexte de santé, et plus particulièrement dans le cadre des problématiques de RI, le simple stockage de tels volumes d'information n'est pas le seul défi à relever :

La rapidité avec laquelle il est possible d'accéder à ces données constitue également un paramètre essentiel [60]. Le succès d'un SRI basé sur un EDS requiert des temps de recherche de l'information « raisonnable » au regard du rythme de travail des divers professionnels de santé susceptibles d'exploiter un tel outil.

De même, l'hétérogénéité de la nature des données impose des modèles de données plus complexes que ceux que l'on peut rencontrer dans le cadre de la RI documentaire classique. Ces

modèles doivent, en effet, prendre en compte des données de natures différentes (e.g. données numériques, textuelles, chronologiques, imageries) et provenant de source différentes mais logiquement reliées entre elles.

Les EDS s'inscrivent pleinement dans une problématique de « Données Massives » ou « Big Data » dont les enjeux majeurs sont souvent synthétisés à l'aide de la règle des 5V :

Volume **V**élocité **V**ariété **V**éracité **V**aleur.

2.2.2 Les objectifs de l'EDS

Les données d'un EDS fournissent une vue de différents paramètres de santé d'une large population de patients qui peuvent avoir des profils similaires ou distincts. Elles constituent donc par nature un vivier de données propices à l'extraction d'informations biomédicales pertinentes. On distingue en santé les notions de recherche interventionnelle et de recherche non-interventionnelle :

La recherche interventionnelle se définit comme un type de recherche dans laquelle une « intervention inhabituelle sur le patient » est réalisée. Dans le cadre de la recherche biomédicale il s'agit d'une intervention sur la personne non justifiée par sa prise en charge habituelle. Dans le cadre des soins courants, la recherche interventionnelle ne porte pas sur les médicaments et tous les actes et produits sont respectivement réalisés et administrés mais selon des modalités particulières de surveillance. Dans tous les cas, ce type de recherche est réalisé suivant un protocole précis et selon le principe de la médecine fondée sur les preuves (Evidence Based Medicine) incluant notamment un essai randomisé contrôlé.

La recherche non-interventionnelle est quant à elle davantage « observationnelle ». Aucun protocole spécifique de prise en charge ou de stratégie médicale n'est fixé à l'avance. Elle consiste en une observation d'une collection d'échantillons biologiques ou de données particulières d'une cohorte de patients sélectionnés sur critères spécifiques.

Les EDSs constituent une source d'information adaptée à ces deux types de recherche. Dans le cadre de la recherche interventionnelle, l'utilisation des EDS peut être envisagée pour la réalisation des études de faisabilité des essais cliniques. De même, l'exploitation et l'extraction des nombreuses données épidémiologiques que contiennent les EDSs représentent un enjeu de la recherche non-interventionnelle notamment dans le cadre de la constitution de registres et de cohortes de patients.

Les EDSs permettent également de renforcer la prise en compte des indicateurs médico-économiques dans la recherche clinique ainsi qu'une optimisation du PMSI et plus généralement des recettes de T2A (e.g.[61]) mais aussi la création de nouveaux indicateurs utiles au pilotage des établissements de santé [62].

Les EDSs offrent enfin des perspectives importantes de recherche davantage fondamentales. Diverses techniques d'intelligence artificielle ont déjà été utilisées ces dernières années principalement sur les données du DPI. Dans [10], des travaux visant à identifier des cohortes de patients à l'aide de techniques de TALN, de statistiques, de fouille de données et d'apprentissage automatique sont relatés. Dans [63], une méthode d'apprentissage profond (Deep Learning), basée sur des réseaux de neurones, est exploitée à des fins de prédiction de maladies. Les volumes et l'exhaustivité supérieure des données disponibles au sein des EDSs en font par conséquent des outils propices à l'exploration :

— de méthodes d'intelligence artificielle ;

— de méthodes de traitement de données massives visant à extraire de nouvelles connaissances à partir de données brutes.

L'exploitation conjointe de ces méthodes représente un potentiel important en termes de progression de la recherche biomédicale notamment dans le cadre du développement d'outils d'aide à la décision.

2.2.3 Panorama des EDSs existants

La construction d'entrepôts de données cliniques est un domaine de recherche largement traité dans la littérature. Il existe, cependant, assez peu d'outils open-source et/ou génériques et/ou distribués compte tenu de la spécificité et du caractère sensible et privé des données cliniques. Cependant, certaines initiatives de standardisation ont, néanmoins, été entreprises :

Le **C**ommon **D**ata **M**odel (CDM) du programme de recherche **O**bservational **M**edical **O**utcomes **P**artnership (OMOP) financé par l'institut américain de la Santé a notamment contribué à la standardisation des EDS d'une manière générale [64]. Ce dernier a été utilisé dans plusieurs travaux de recherche (e.g. [24]). Il repose sur un modèle de type **E**ntity–**A**ttribute–**V**alue (EAV) [2] permettant l'agrégation de données de santé provenant de sources disparates. Il permet d'en constituer une représentation terminologique commune et d'effectuer des analyses systématiques sur ces données.

Le datamart **I**nformatics for **I**ntegrating **B**iology and the **B**edside (i2b2) est sans aucun doute la solution à la fois la plus connue et la plus adoptée dans le monde dans le cadre de la construction d'EDSs.

En France, il existe également plusieurs EDSs qui ont été développés. On distingue notamment **Con**tinuum **So**ins-**Re**cherche (ConSoRe) [65], **E**ntrepôt de données biomédicales de l'**HOP**ital (eHOP) [66] ou encore Dr. Warehouse (Dr Warehouse) [67].

Bien que peu génériques, et non distribuées, certaines solutions constituent des références du domaine, comme par exemple **S**tanford **T**ranslational **R**esearch **I**ntegrated **D**atabase **E**nvironment (STRIDE) [1] (États-Unis). La majorité de ces dernières correspondent néanmoins à des solutions propres aux établissements de santé au sein desquels elles ont été développées et utilisées localement dans leurs SIHs et possèdent des caractéristiques diverses et variées.

Dans cette section, un panel d'EDSs ou de systèmes dédiés à l'accès aux données cliniques sont présentés. Ces derniers sont présents de manière régulière dans la littérature scientifique et constituent des outils référents dans le domaine. Bien que non exhaustif, cet échantillon permet de mettre en évidence les principales caractéristiques de ces systèmes et d'identifier leurs limites et leurs perspectives d'amélioration. Une analyse de ces derniers sera effectuée dans la sous-sous-section 2.2.3.10 p. 66.

2.2.3.1 Informatics for Integrating Biology and the Bedside

Informatics for **I**ntegrating **B**iology and the **B**edside (i2b2) [68, 69] est un framework libre permettant d'intégrer des données cliniques et génomiques existantes afin d'en permettre la recherche et d'assurer la mise au point de thérapies ciblées et personnalisées. i2b2 a été initialement développé à Boston au Massachusetts General Hospital au sein du système de santé Partner's HealthCare System dans lequel il servait d'architecture pour le Research Patient Data Registry. Suite à une demande de financement auprès de l'Institut Américain de la Santé (National Institutes of Health) de la part de chercheurs de la Harvard Medical School et du Massachusetts General Hospital, le code d'i2b2 est rendu public en 2007.

i2b2 bénéficie aujourd'hui d'une large utilisation à travers le monde et d'une communauté riche. Son utilisation est très largement répandue aux États-Unis notamment dans le cadre de la recherche clinique et de la constitution de cohorte de patients [70]. Il est aussi utilisé au sein de divers EDS français et notamment celui de l'**A**ssistance **P**ublique – **H**ôpitaux de **P**aris (AP–HP) [71], du **C**entre **H**ospitalier **U**niversitaire (CHU) de Rennes ou encore celui du **C**entre **H**ospitalier **R**égional **U**niversitaire (CHRU) de Nancy [72]. Une instance d'i2b2 à été testée au CHU de Rouen pendant la thèse de Chloé CABOT, mais n'est pas utilisée jusqu'à ce jour en pratique courante. Les performances en terme de temps de réponse ne sont pas adaptées à la volumétrie de notre EDS (cf. Table 1.2 p. 40).

Le projet i2b2 est décrit comme une « ruche » de modules (viz. « i2b2 Hive ») indépen-

2. : « **E**ntité–**A**ttribut–**V**aleur »

dants les uns des autres et inter-opérants au sein d'une **A**rchitecture **O**rientée **S**ervice (AOS). La Figure 2.3 p. 59 montre l'ensemble des modules composant le projet **i2b2**.

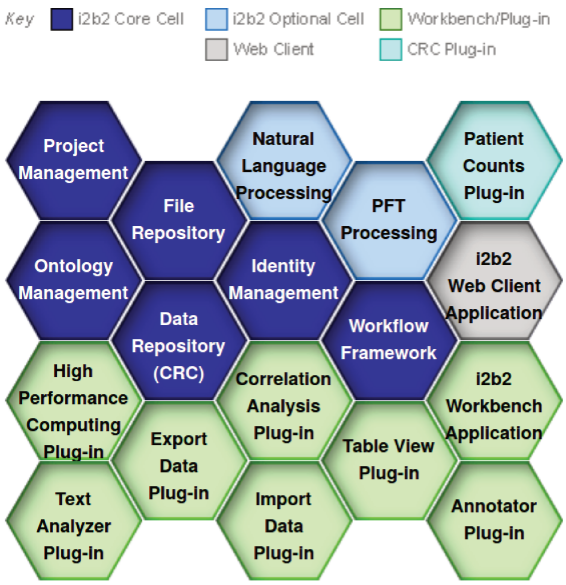


FIGURE 2.3 – Collection de modules constituant le projet **i2b2**
Source : *https://www.i2b2.org/software/index.html*

Le module « Clinical Research Chart » (cf. cellule « Data Repository modules (CRC) » de la Figure 2.3 p. 59) et le module « **i2b2** Workbench » (cf. cellule « **i2b2** Workbench Application » de la Figure 2.3 p. 59) jouent tous deux un rôle essentiel dans le **i2b2**.

Le module « Clinical Research Chart » est relatif au système de stockage des données employé par **i2b2** qui est en l'occurrence un SGBDR. Trois types de bases de données sont supportées : SQL Server, ORACLE 10g et PostgreSQL (Postgre SQL). Le modèle de données utilisé par **i2b2** est quant à lui connu sous le nom de « modèle en étoile » (viz. « star schema » en anglais) en raison de l'aspect de son diagramme de classe (cf. Figure 2.4 p. 60).

Le modèle en étoile d'**i2b2** est un modèle de type EAV initialement proposé par Ralph KIMBALL [74]. Ce type de modèle a été largement employé comme système de représentation de l'information ces dernières années et plus particulièrement dans les cadres des bases de données biomédicales hétérogènes et complexes [75]. Cet « aspect EAV » confère au modèle en étoile d'**i2b2** une généricité dans le sens où il permet d'intégrer sans altération structurelle du modèle de données lui même (e.g. ajout, suppression, modification de tables ou de colonnes etc.) un grand nombre de données cliniques indépendamment à la fois de leurs représentations dans leurs systèmes de stockage d'origine, de leurs logiques de représentation et de leurs variabilités au cours du temps. Il est à noter cependant que certaines structurations d'informations cliniques ne peuvent être recréées au sein du modèle d'**i2b2** [70].

Le modèle **i2b2** est pensé autour de la notion de « fait » matérialisé par une table centrale (viz. la table « `OBSERVATION_FACT` ») reliée à plusieurs tables de dimensions (e.g. `PATIENT_DIMENSION`, `VISIT_DIMENSION`, `CONCEPT_DIMENSION`, `PROVIDER_DIMENSION`).

Un fait correspond dans ce modèle à une observation sur un patient faite à un moment précis, par un acteur spécifique et au cours d'un événement précis. Les faits contiennent des données quantitatives et factuelles qui correspondent aux données requêtables par un utilisateur. Un fait est décrit à l'aide d'attributs basiques tels que :

— le numéro du patient concerné par l'observation ;
— le code du concept décrivant le type de l'observation ;
— les dates de début et de fin de cette dernière ;

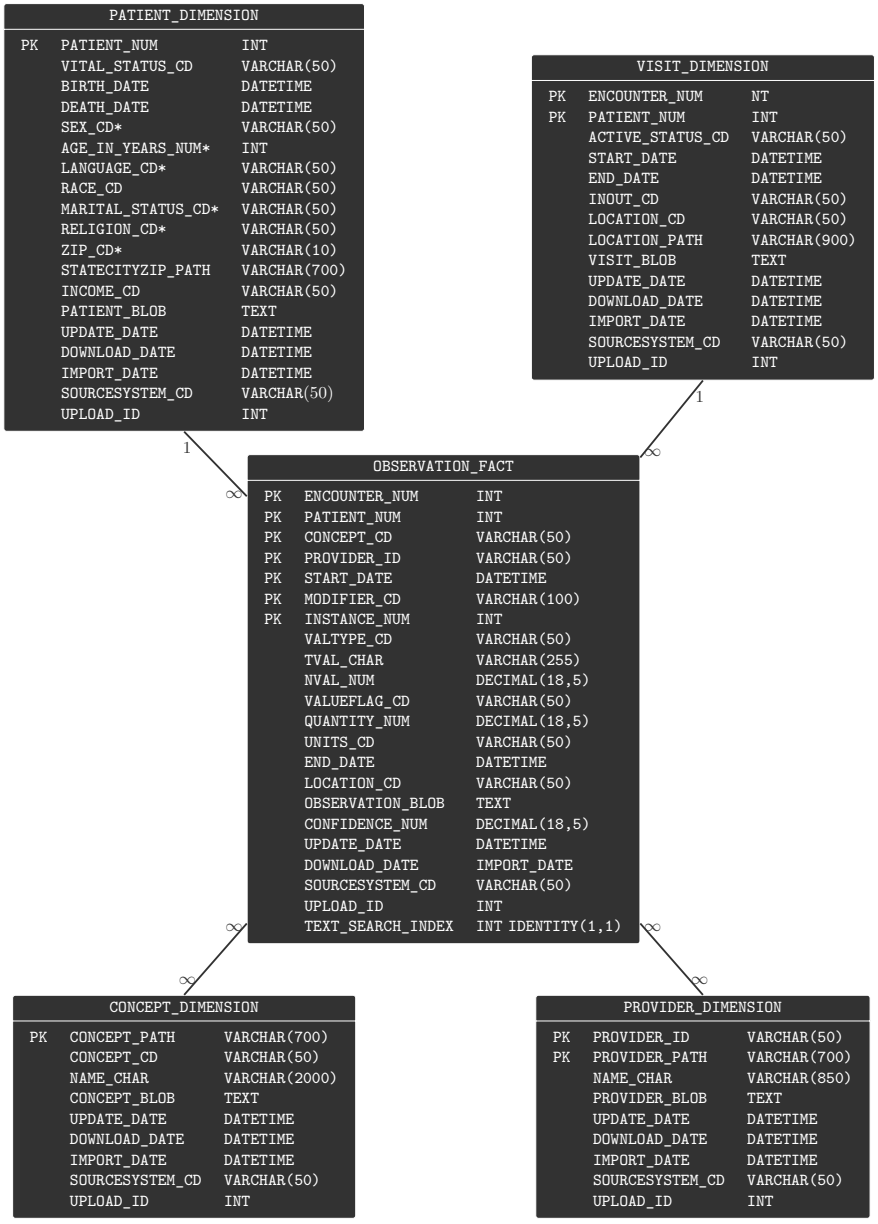


FIGURE 2.4 – Modèle en étoile d'i2b2
Source : *adapté de for Integrating Biology and the Bedside [73]*

— etc.

Les tables de dimension contiennent, quant à elles, des codes qui peuvent être intégrés au sein d'une hiérarchie précise. Ces codes sont rattachés aux faits et ont pour rôle de décrire et de caractériser pleinement ces derniers.

En pratique, les cinq tables du modèle en étoile i2b2 ne sont pas suffisantes pour intégrer l'ensemble des données biomédicales d'un EDS. De nombreuses tables supplémentaires sont nécessaires pour définir les codes décrivant et structurant les faits. L'un des atouts majeurs d'i2b2 est la modélisation des méta-données [76] (cf. cellule « Ontology Management » de la ruche Figure 2.3 p. 59). Les méta-données d'i2b2 permettent de définir le vocabulaire qui permet de décrire finement les faits. Ce vocabulaire permet évidemment de définir les SOCs classiquement utilisés dans le cadre des données cliniques (e.g. CIM-9,[3] SNOMED–CT®,[4] LOINC,[5] etc.) mais aussi des hiérarchies de codes locaux à l'EDS. i2b2 donne la possibilité d'organiser ces concepts

3. url : `https://www.cdc.gov/nchs/icd/icd9.htm`
4. url : `https://www.snomed.org/snomed-ct/`
5. url : `https://loinc.org/`

au sein d'une hiérarchie de dossier ce qui permet une description compréhensible et intuitive des faits du point de vue de l'humain.

i2b2 fournit une interface d'accès aux données à travers l'outil **i2b2** ***Workbench Query Tool***. Cette interface permet de composer des requêtes afin d'extraire de la base de données de l'EDS un ensemble de patients répondant à des critères cliniques et démographiques (cf. Figure 2.5 p. 61). Les requêtes de cet outil sont composées de manière graphique par simple glisser–déposer de contraintes prédéfinies et organisées au sein d'une structure arborescente. Cette arborescence est en réalité définie par l'organisation des concepts en hiérarchie de dossier évoquée au paragraphe précédent. Ainsi la définition des méta-données d'**i2b2** permet non seulement une description des faits de manière fine et intuitive au niveau conceptuel mais également de définir la manière dont l'utilisateur pourra interagir avec la base de données en influant sur la présentation des contraintes cliniques et démographiques au sein de l'outil d'accès.

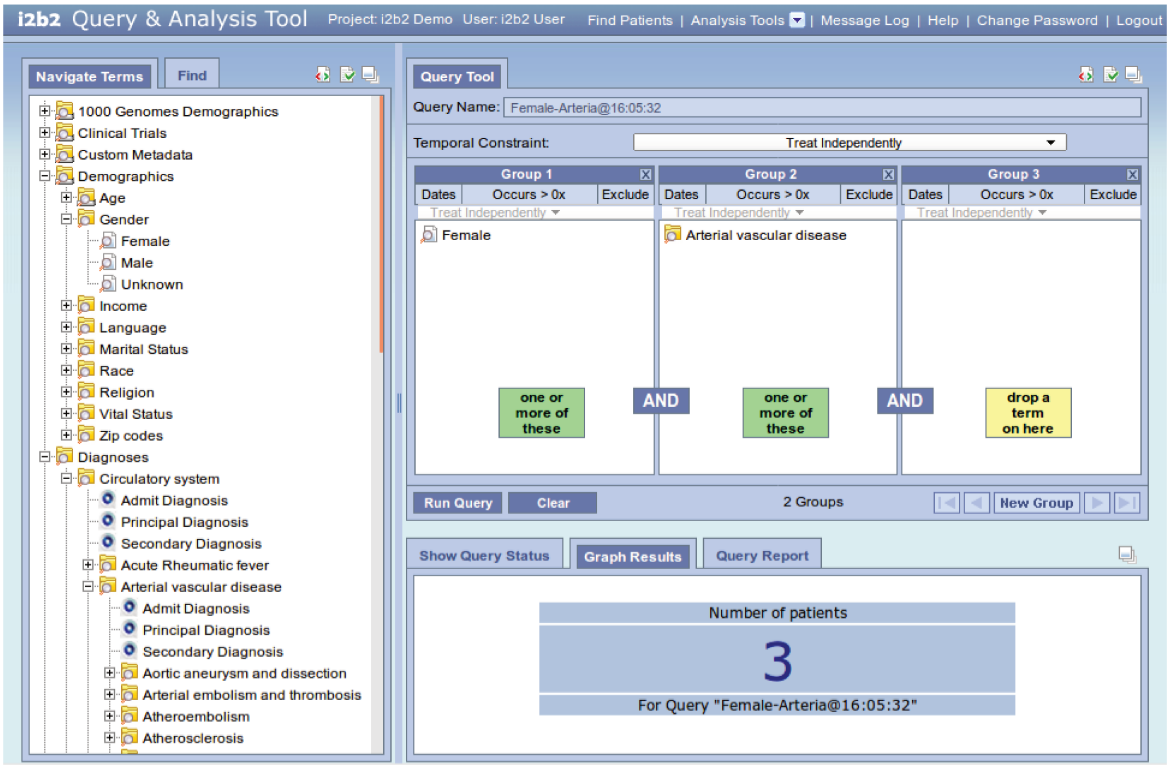


FIGURE 2.5 – Outil de requêtage de démonstration d'**i2b2** (**i2b2** Workbench Query tool). Les patients ici recherchés sont les femmes ayant fait l'objet d'un diagnostic quelconque de la catégorie des maladies vasculaires artérielles. Les contraintes `Female` et `Arterial vascular disease` ont été glisser–déposer depuis le champ `Navigate Terms` vers les blocs correspondants du champ `Query Tool`.

2.2.3.2 Stanford Translational Research Integrated Database Environment

Stanford **T**ranslational **R**esearch **I**ntegrated **D**atabase **E**nvironment (STRIDE) [1] est un projet **R**echerche et **D**éveloppement (R&D) initié en 2003 par la division **I**nformation **R**esources and **T**echnology (IRT) de l'école de médecine de Stanford aux États-Unis. En 2003, lorsque ce projet a été lancé, les fonctionnalités offertes par les systèmes existants, tels que **ca**ncer **B**iomedical **I**nformatics **G**rid (caBIG) et **i2b2**, étaient jugées non optimales, en particulier par rapport au besoin d'un entrepôt clinique entièrement identifié. Utilisé quotidiennement au Stanford University Medical Center, STRIDE constitue la plateforme d'accès et de gestion de l'EDC de ce dernier. Il vise à fournir des outils pour la recherche clinique et la recherche translationnelle. En 2009, l'EDC contenait les informations cliniques de plus de 1,3 millions de patients pris en charge au centre Médical de l'Université de Stanford depuis 1995. Des volumétries plus détaillées sont données dans la Table 2.1.

Types de données	volumes de données	antériorités des données
Consultations cliniques	10,5 Millions	depuis 1994
Diagnostics (CIM-9)	15 Millions	depuis 1994
Actes (CPT, CIM-9)	10 Millions	depuis 1994
Comptes-rendus de Radiologie	1,8 Millions	depuis 2005
Comptes-rendus de Pathologie	1 Million	depuis 1995
Comptes-rendus	4,8 Million	depuis 2005
Résultats d'examens biologiques	93 Millions	depuis 2000
Ordonnances	4,3 Millions	depuis 2006

TABLE 2.1 – Principales catégories de données stockées dans l'EDS de STRIDE. [1]

Techniquement, STRIDE repose sur un SGBDR Oracle 11g muni d'un modèle EAV basé sur la norme HL7–RIM. La représentation des données est réalisée avec une vision orientée Objet dérivée du modèle HL7–RIM (viz. représentation des données en « Entity », « Role », « Acts », etc.). Tout comme **i2b2**, STRIDE utilise au sein de son modèle une couche sémantique des standards terminologiques pour représenter les concepts médicaux importants et leurs relations entre eux (e.g. SNOMED 3.5, RxNorm [77] (Nomenclature standardisée éditée par NLM qui contient l'ensemble des médicaments disponibles aux États-Unis), **C**lassification **I**nternationale des **M**aladies (CIM), **C**urrent **P**rocedural **T**erminology (CPT), etc.).

La base de données physique de STRIDE est partitionnée en trois « sous-bases » :

— une base de données cliniques (l'EDC) ;

— une base de données pour les données de recherches ;

— une base de données pour les données relatives aux échantillons biologiques.

L'accès et la gestion des données de ces trois composants reposent sur une architecture commune de type AOS. Elle est exploitée par différentes applications à des fins cliniques ou de recherches. Bien que les données soient partitionnées, cette architecture permet de combiner les données de ces trois sources.

Le requêtage des informations au sein de la base de données de STRIDE est réalisé infine par génération d'une requête **S**tructured **Q**uery **L**anguage (SQL). Une couche de sécurité permet néanmoins de maîtriser l'accès aux informations au sein de ce processus de génération de la requête SQL. Toutes les informations non-structurées (viz. comptes-rendus et autres documents médicaux) peuvent être requêtées en plein texte grâce à l'exploitation d'index Oracle Text tandis que l'accès aux données structurées est basé sur l'utilisation de TOs et de leurs hiérarchies.

L'application ***Anonymous Patient Cohort Discovery Tool*** permet la recherche de cohortes de patients. De manière similaire à **i2b2**, la constitution de la requête est réalisée à partir de « glisser–déposer » de contraintes visualisables au sein d'une structure arborescente (cf. Figure 2.6 p. 63). L'identité des patients n'est pas révélée dans cet outil, des méthodes de binning [6] ayant été utilisées pour anonymiser les cohortes. D'autres applications permettent l'analyse de cohortes par l'intermédiaire de graphiques ou de diagrammes, l'extraction de données cliniques, la gestion de données de recherches ou relatives à des échantillons biologiques.

6. : « mise en récipient »

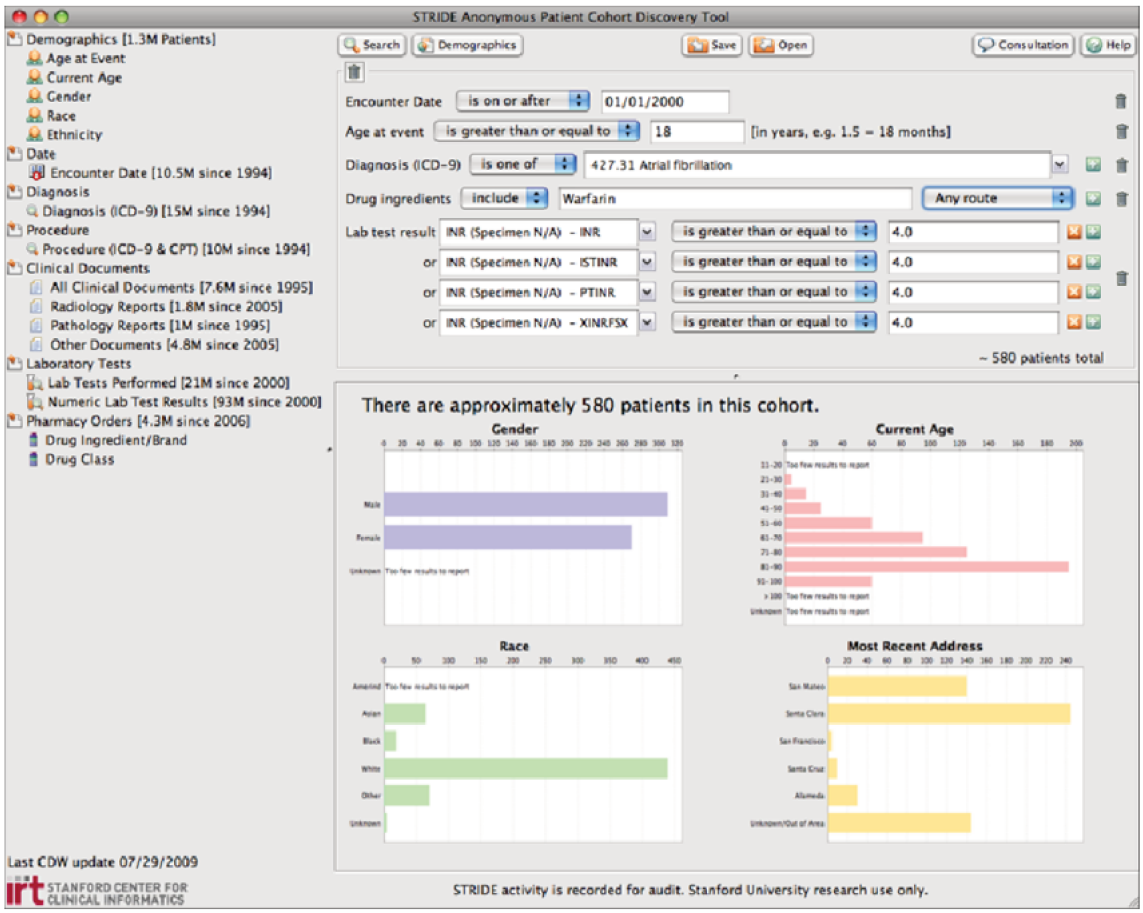


FIGURE 2.6 – Copie d'écran de l'outil de création de cohortes de patients basé sur STRIDE [1]

2.2.3.3 Enterprise Data Trust

L'**E**nterprise **D**ata **T**rust (EDT) [78] est un EDS industriel initié en 2005 à la clinique Mayo à Rochester dans le Minnesota aux État-Unis. Il recueille des données sur les soins prodigués aux patients, l'éducation, la recherche et l'administration afin d'appuyer la RI, la veille économique et la prise de décisions de haut niveau. L'EDT s'appuie intensivement sur des technologies industrielles (e.g. IBM InfoSphere Information Server, Teleran iSight & iGuard, SAP BusinessObjects, Sybase PowerDesigner, etc.) et permet l'intégration et l'exploitation de volumes extrêmement importants de données (e.g. plus de 7 millions de patients uniques, 64 millions de diagnostics, 268 millions de tests etc.). Il est à noter que la clinique Mayo est une structure de grande envergure comptant plus de 50 000 employés. L'architecture et les fonctionnalités de l'EDT reposent sur des composants techniques historiques et des travaux de gouvernance de longue date sur : la gestion des données et des métadonnées, la modélisation des données et les vocabulaires normalisés. Ces initiatives fournissent à cet EDS une organisation fiable de l'information patient, de la génomique et des données de recherche ainsi que des capacités d'interrogation pour la sélection des cohortes et la génération de données agrégées.

2.2.3.4 Data Warehouse for Translational Research

Aux États-Unis le Windber Research Institute a développé le **D**ata **W**arehouse **f**or **T**ranslational **R**esearch (DW4TR) [79]. Ce dernier a été utilisé dans le cadre de multiples projets de recherches translationnelles. L'un des aspects fondamentaux de cet outil est sa représentation hautement structurée des informations médicales. Il adopte, en effet, une représentation tridimensionnelle de ces informations dont les trois composantes permettent de décrire : les données cliniques, les données moléculaires et les informations temporelles. Les données du DW4TR sont, dans un premier temps, collectées dans un SGBDR ORACLE muni d'un modèle de

données de type EAV et sont ensuite hébergées dans un modèle de données extensible qui permet des les organiser en une structure de modules hiérarchiques héritée d'ontologies spécialement développées à cet effet. Deux interfaces graphiques permettent de fournir un accès à des données agrégées destinées à l'analyse de ces dernières telles que des moyennes, des écarts-types, des volumétries, des données catégorielles ou encore des vues chronologiques.

2.2.3.5 SMart EYE DATabase

SMart **EYE DAT**abase (SMEYEDAT) [80] est un EDS hautement spécialisé en ophtalmologie. Il est développé au centre universitaire ophtalmologique de Munich en Allemagne. SMEYEDAT repose sur une base de données Microsoft SQL mise à jour quotidiennement à partir du SIH et utilise un modèle de données en étoile centré sur le patient pour la représentation des données. Qlikview [7] (Qliktech, Radnor, Pennsylvanie, État-Unis) a été implémenté comme outil analytique pour visualiser et explorer les données patient. Cette interface permet la sélection des patients selon des critères et des vues spécifiques au domaine opthalmologique.

2.2.3.6 Entrepôt de données biomédicales de l'HÔPital

L'**E**ntrepôt de données biomédicales de l'**HOP**ital (ehop) [66] est un moteur de requêtes permettant d'interroger l'EDS du CHU de Rennes. Initié avec le projet Roogle [81], cet outil a été conjointement développé par l'équipe ***Données Massives en Santé*** du **L**aboratoire **T**raitement du **S**ignal et de l'**I**mage (LTSI) (unité Inserm) et le CHU de Rennes. ehop est aujourd'hui en cours de déploiement dans différents hôpitaux du « grand Ouest » en France (e.g. CHU d'Angers et de Brest). Ces déploiements s'inscrivent plus largement dans le cadre du projet R–CDC, co-coordoné par les CHU de Rennes et de Brest, visant à mettre en place un réseau de Centre de Données Cliniques qui serait alors le premier de ce type en France, avec l'expérience de ConSoRe et certains **C**entre de **L**utte **C**ontre le **C**ancer (CLCC).

ehop a pour mission globale de traiter et exploiter les gisements de données patient afin de faciliter la recherche médicale. Le moteur de requêtes ehop permet d'interroger des données patient pour une utilisation secondaire (viz. étude de faisabilité, pré-screening) en permettant l'analyse et la fouille de données sur l'EDS du CHU de Rennes. Ces fonctionnalités sont assurées par un module *R*. ehop est décrit comme un outil pouvant être exploité dans des contextes très variés (e.g. recherche clinique, étude épidémiologique, évaluations thérapeutiques, pharmacovigilance, détection d'infection nosocomiale, analyse médico-économique, etc.).

La collecte des données au sein des SIs des différents établissements (e.g. données du PMSI, comptes-rendus de consultation, d'hospitalisation et de passage aux urgences, prescriptions, données générées par les actes de soins, etc.) est assurée par la plateforme d'interopérabilité de la société Enovacom. La Table 2.2 donne un aperçu sommaire des volumes de données gérés par ehop au sein du CHU de Rennes.

Types de données	volumes de données
Patients	1,2 Millions
Documents	27 Millions
Éléments de données	130 Millions

TABLE 2.2 – Volumes de données gérés par ehop au CHU de Rennes. [4]

L'échange sécurisé des données est assuré par différents standards d'interopérabilité tels que HL7 mais aussi le standard Phast PN13 pour la prise en charge médicamenteuse du patient ainsi que la norme française **H**armoniser et **PR**omouvoir l'**I**nformatique **M**édicale (HPRIM) pour le transport des examens de biologie.

7. url : `https://www.qlik.com/us/products/qlikview`

ehop est basé sur le SGBDR Oracle et la technologie NoSQL MongoDB. Le système permet une interrogation des données structurées mais également une recherche plein texte sur les documents non structurés. Un enrichissement sémantique (i.e. indexation des documents médicaux) est réalisé à l'aide de plusieurs terminologies médicales (viz. CIM–10, thesaurus de l'**A**ssociation pour le **D**éveloppement de l'**I**nformatique en **C**ytologie et en **A**natomie **P**athologique (ADICAP), SNOMED 3.5, LOINC).

Afin de respecter les règles éthiques, juridiques et déontologiques, l'accès à l'outil ehop se fait par l'intermédiaire d'un professionnel habilité du **D**épartement d'**I**nformation **M**édicale (DIM) de Rennes. Un entretien entre le clinicien formulant une demande d'information et ce professionnel permet de définir les critères d'interrogation de l'entrepôt. De plus, une dé-identification et une stratégie de traçabilité des accès aux documents fournis in-fine au clinicien est mise en place.

2.2.3.7 CONtinuum SOins–REcherche

Continuum **So**ins-**Re**cherche (ConSoRe) [65] est un projet initié en 2013 par UNICANCER, la fédération nationale des CLCCs. Le développement de cet outil a été réalisé par la société SWORD [8] et sa première version a vu le jour en 2015. ConSoRe est un outil très spécialisé dans le domaine de l'oncologie permettant de retrouver des informations cancérologiques disséminées dans les textes cliniques des centaines de milliers de dossiers des patients des CLCCs français. On dénombre, au total, une vingtaine de ces centres [9]. Huit d'entre eux bénéficient déja de ConSoRe [10] et l'installation est en cours dans trois autres centres [11]. Il permet à ces CLCCs d'effectuer des recherches de patients et de constituer des cohortes à la fois dans un périmètre restreint au CLCC en question ou à l'échelle nationale. Seules des volumétries sont cependant fournies en sortie de l'outil dans le cadre d'un requêtage national. C'est, de plus, un outil autonome nécessitant une phase d'**E**xtraction **T**ransformation **E**nrichissement **P**ublication (ETEP) des diverses données qu'il exploite (e.g. comptes-rendus médicaux, échantillons biologiques, diagnostics, données de chimiothérapie, etc.) comprenant notamment une phase d'annotation à l'aide de divers standards internationaux (e.g. CIM–10, **C**lassification **I**nternationale des **M**aladies **O**ncologiques (**3ème** révision) (CIM–O–3) [82], ADICAP, Vidal, etc.). Ces TOs sont également utilisées dans la processus de requêtage pour exprimer des requêtes structurées au sein d'un formulaire.

2.2.3.8 Dr. Warehouse

Dr. Warehouse (DrWarehouse) [67] est un entrepôt de données open-source principalement orienté sur la RI au sein des textes cliniques et conçu pour une utilisation quotidienne dans le cadre de trois cas d'usages spécifiques : (1) la recherche clinique, (2) la détection d'hypothèses ou la fouille de données et (3) la recherche translationnelle. Ce projet a été initié, en 2015, au sein de l'institut de recherche **Imagine** spécialisé dans les maladies génétiques et faisant lui-même partie de l'**Hôpital Necker-Enfants malades** [12] à Paris (France). DrWarehouse adopte un modèle de données centré sur le document (i.e. texte clinique) et non le patient. Ce modèle permet l'intégration de données textuelles en premier lieu, mais également de données structurées (e.g. séjours, données sur le patient, etc.) et notamment des données codées (e.g. analyses biologiques, diagnostics, etc.). DrWarehouse repose sur le SGBDR ORACLE et plus particulièrement sur le module Oracle Text pour fournir des fonctionnalités évoluées de RI au sein des textes cliniques telles que la détection des négations et des antécédents ou des

8. url : `https://www.sword-group.com/en/expertise/`
9. url : `http://www.unicancer.fr/le-reseau-des-centres-de-lutte-contre-le-cancer`
10. Il s'agit des centres suivants : François Baclesse (Caen), Jean- Perrin (Clermont-Ferrand), François-Leclerc (Dijon), Oscar Lambret (Lille), Léon-Bérard (Lyon), l'Institut Paoli-Calmettes (Marseille), l'Institut Curie (Paris) et l'Institut du Cancer de Montpellier.
11. Il s'agit des centres suivants : l'Institut Bergonié (Bordeaux), l'institut de cancérologie de Lorraine (Nancy) et le Centre Paul Strauss (Strasbourg). Dans le cadre d'une collaboration, cet outil devrait, en outre, être installé au sein des hôpitaux universitaires de Strasbourg
12. url : `http://hopital-necker.aphp.fr/`

contextes familiaux par exemple [83].

L'interrogation de l'outil s'effectue à l'aide d'une requête en langage naturel générant une recherche au sein des différents textes cliniques contenant les mots clés de cette dernière. La liste des patients et des extraits de texte au sein desquels ces mots ont pu être identifiés est fournie à l'utilisateur. Des filtres portant sur les données structurées peuvent être appliqués.

2.2.3.9 L'EDS de l'Hôpital Européen George–Pompidou

En 2008, l'**H**ôpital **E**uropéen **G**eorges–**P**ompidou (HEGP) (Paris, France) de l'AP–HP a lancé la création d'un EDS [71] basé sur le datamart i2b2. Ce dernier est fortement intégré dans le SI clinique de l'hôpital qui s'appuie lui même sur plusieurs solutions industrielles (e.g. ONECALL de McKesson, Act management (CPOE) de MEDASYS, plate-forme d'intégration de THALES). L'infrastructure centrale de l'EDS repose également sur le SGBDR ORACLE pour le stockage (1,2 million de patients, 1 million de séjours) et le framework i2b2 pour la représentation des données. Plusieurs applications clientes sont connectées au système pour fournir un accès technique aux données mais, en ce qui concerne les chercheurs, le client i2b2 est celui qui est principalement utilisé.

2.2.3.10 Discussion

Dans le cadre général de la gestion et de la manipulation de données cliniques, les bases de données relationnelles ont déjà montré par le passé certaines limites en termes de scalabilité [84]. Une grande majorité des systèmes d'accès à des informations cliniques repose toujours sur un SGBDR. Parmi les neufs systèmes présentés précédemment, six exploitent pleinement ce type de SGBD comme méthode de stockage et de modélisation des données : i2b2, STRIDE, DW4TR, SMEYEDAT, DrWarehouse, l'EDS de l'HEGP. L'EDT et ConSoRe, quant à eux, font intensivement usage de solutions industrielles pour faire face à l'extrême volumétrie des données. Ces deux outils offrent donc des perspectives de recherches moindres compte-tenu du manque de reproductibilité de ces travaux.

L'orientation persistante des EDSs existants vers des SGBDRs s'explique premièrement par la maturité de ces systèmes de stockage. Le modèle relationnel est, en effet, la méthode la plus courante et la plus éprouvée pour stocker et interroger des données sous diverses formes. Comme il le sera plus amplement mis en exergue dans le chapitre 4, de nouveaux types de SGBDs ont vu le jour ces dernières années. Ces derniers semblent offrir de meilleurs performances dans le cadre de données cliniques [85]. Ils reposent sur un paradigme de stockage des données radicalement différent qui n'offre pas la même rigueur de modélisation, d'accès et de sécurisation des données et des transactions. Ainsi, les SGBDRs, et notamment le SQL qui les accompagne, sont encore aujourd'hui un gage de sécurité quant à la mise en place d'une RI efficace au sein d'un EDS. Ceci reste d'autant plus vrai compte tenu de la nature dynamique, sporadique et hétérogène des données cliniques [86].

Dans la pratique, cette complexité de l'information clinique implique diverses spécialisations de nombreux EDSs. SMEYEDAT et ConSoRe sont, par exemple, des outils donnant un accès à des données se rapportant respectivement aux domaines de l'ophtalmologie et de l'oncologie.

D'autres outils tirent leur spécificité des fonctionnalités qu'ils proposent. **E**lectronic **ME**dical **R**ecord **S**earch **E**ngine (EMERSE), DrWarehouse ou encore ehop axent ainsi davantage leurs efforts sur la recherche au sein des textes cliniques. Il est à noter que DrWarehouse et ehop permettent l'ajout de filtres portant sur les méta-données et autres données structurées alors que EMERSE est un outil purement destiné à la recherche au sein de données non structurées [13].

Les solutions plus génériques telles que i2b2, STRIDE mais aussi DW4TR adoptent un modèle de type **E**ntity–**A**ttribute–**V**alue. Ce type de modèle apporte une plus grande flexibilité quant à la modélisation de l'information clinique au sein d'un modèle de données relationnel habituellement « rigide ». Le CDM de l'OMOP peut également être classé dans cette catégorie

13. EMERSE n'est pas, à proprement parler, un EDS mais constitue un outil de référence de la littérature

de modèle.

Cette adoption généralisée de modèles EAV dénote une difficulté particulière à établir une représentation suffisamment générique de l'information clinique à même de satisfaire l'étendue des cas d'usages de cette dernière. Cela se traduit également par une multiplicité des applications clientes (i.e. interfaces) permettant d'interroger ces EDSs en fonction du contexte d'utilisation. À titre d'exemple, STRIDE s'accompagne d'interfaces différentes pour la sélection de cohortes de patients, la visualisation de ces cohortes à l'aide de diagrammes, l'extraction d'informations cliniques, la gestion de données de recherches ou encore la gestion de données biologiques. De même, DW4TR fournit deux interfaces d'interrogation. Ces outils fournissent généralement des données agrégées (e.g. moyennes, écarts-types, diagrammes, volumétries, etc.).

À l'exception de DrWarehouse dont le modèle de données est volontairement centré sur le concept de document clinique, la plupart des EDSs adoptent une vision « mono-centrée » sur le concept de patient. Cette vision est perceptible à la fois au niveau du modèle de données et des fonctionnalités de recherches fournies. Les applications clientes fournissent généralement en sortie des listes de patients ou des données agrégées s'y rapportant. En outre, leur sélection s'effectue majoritairement par l'intermédiaire de formulaires pré-établis ou dynamiques dans lesquels les concepts médicaux annexes tels que celui de séjour, d'analyse biologique, de diagnostic etc. sont vus comme des méta-données potentielles du concept de patient et non comme un concept autonome pouvant faire l'objet d'un intérêt propre. Ces derniers dérivent pourtant du processus de prise en charge des patients et constituent des informations importantes et utiles à diverses questions cliniques que peuvent être amenés à se poser les professionnels de santé. La prise en compte de ces informations de manière individuelle au sein d'un processus de RI plus générique pourrait donc ouvrir la porte à de nouvelles perspectives. Il convient donc d'étudier dans quelles mesures les méthodes classiques de la RI peuvent s'avérer utiles à la réalisation de cet objectif et s'accorder avec cette structuration en entités multiples de l'information clinique.

Dans le chapitre suivant, je présente dans un premier temps les méthodes classiques de la RI. Dans un second temps, je m'attache à mettre en évidence la singularité des besoins de RI dans le domaine de la Santé.

Chapitre 3

La recherche d'information en Santé

Sommaire

3.1 Les fondements de la recherche d'information 70
3.1.1 Le contexte 70
3.1.2 Le rôle 71
3.1.3 Le principe 71
3.2 Les modèles de recherche d'information classiques 74
3.2.1 Le modèle Booléen 74
3.2.2 Le modèle Vectoriel 76
3.2.3 Le modèle Probabiliste 79
3.3 Le contexte de la santé 82
3.3.1 La recherche d'information textuelle en Santé 83
3.3.2 La recherche d'information au sein de données cliniques 86

L'omniprésence de l'Internet et du Web dans notre société actuelle fait de la **R**echerche d'**I**nformation (RI) une notion très largement répandue. D'un point de vue scientifique, celle-ci ne se limite cependant pas à la recherche de ressources Web et sa mise en application s'intègre dans une cadre plus large que celui du développement de moteurs de recherche sur le Web.

Dans une vision purement historique et épurée de considération technique et/ou scientifique, la RI tire ses racines des premières tentatives d'organisation d'informations destinées à faciliter leur récupération. C'est donc avec la création des premières bibliothèques que les prémices de la RI ont vu le jour. L'utilisation d'index permettant de catégoriser le contenu de ces bibliothèques est alors devenue la norme. Avec l'arrivée des ordinateurs et la création automatisée de ces derniers la RI a connu un essor important.

La notion de RI en tant que domaine de recherche telle que nous la connaissons aujourd'hui est apparue dans les années 1950 notamment sous l'impulsion de pionnier tels que Allen KENT qui publie en 1955 les métriques phares de précision et de rappel [87] ou encore Joseph BECKER et Robert HAYES qui publient le premier livre sur la RI [88].

Aujourd'hui, la notion de RI évolue dans un univers de recherche très vaste. Elle a pris diverses formes et a donné naissance à de nouvelles problématiques de recherche. Il convient donc de fixer les limites de ce domaine de recherche et de définir les concepts sur lesquels elle repose. Dans cette section, la notion d'**information** sera brièvement contextualisée. Une définition de la RI sera ensuite donnée avant d'en définir le principe. Les différents modèles de RI classique sont présentés et je mettrai en évidence les problématiques spécifiques du domaine de la santé et sa mise en œuvre pour une RI efficace.

3.1 Les fondements de la recherche d'information

La notion de RI fait appel à celle d'**Information**. Bien que celle-ci soit une notion à priori relativement « commune », elle reste difficile à définir de manière universelle. Afin de clarifier le sens de cette dernière dans le contexte spécifique de la RI, la section suivante s'attache à la définir de manière relative.

3.1.1 Le contexte

On distingue dans le domaine des sciences de l'information quatre concepts clés : le concept de « Donnée », le concept d'« Information », le concept de « Connaissance » et celui de « Sagesse ». Ces concepts dépassent cependant le cadre de l'informatique. Il existe dans la littérature scientifique un nombre important de définitions à la fois incohérentes et incompatibles de ces derniers, ne faisant, de surcroît, l'objet d'aucun consensus [89]. Ces notions sont néanmoins systématiquement définies de manière relative. La pyramide **D**ata, **I**nformation, **K**nowledge and **W**isdom (DIKW) [1] permet de les hiérarchiser et de les définir les uns par rapport aux autres (cf. Figure 3.1).

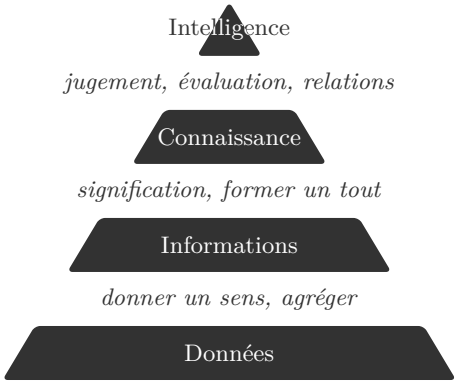


FIGURE 3.1 – Pyramide DIKW (Data, Information, Knowledge, Wisdom) [2]

Dans l'absence de consensus on se contentera de la « vision » suivante :

Données : Les données sont des ensembles de signes et/ou de symboles « bruts ». Elles constituent une mesure, une représentation ou une perception empirique des propriétés d'un objet, d'un fait ou d'un événement ayant une existence dans l'environnement. Les données sont par nature des observations objectives non interprétées. Dans la pratique, elles se présentent sous la forme de mots, de textes, de nombres, de diagrammes, d'images, etc..

Informations : Les informations sont obtenues par l'analyse, l'organisation ou encore l'agrégation structurée des données. Elles constituent un ensemble de données auquel un sens a été donné par l'établissement de connexions relationnelles entre elles. Contrairement aux données, les informations constituent une interprétation et permettent de fournir une description contextuelle et utile d'une situation spécifique.

Connaissances : Elle constitue une accumulation d'information et une synthèse de ces dernières et permet la compréhension d'un sujet ou d'un domaine spécifique. La connaissance s'acquiert par l'expérience. Contrairement à l'information, la connaissance est souvent décrite comme non transférable. En informatique, et plus spécifiquement dans le cadre du Web Sémantique, la connaissance est souvent vue comme de l'information apprise des données et des informations qui en sont issues.

Sagesse : La sagesse, parfois aussi remplacée par « Intelligence », constitue une fonction mentale. Elle est issue d'une acquisition pleine et entière des connaissances procurant la capacité de s'en servir pour prendre de « bonnes décisions ». Elle correspond à la capacité

1. : « **D**onnée, **I**nformation, Connaissance et Intelligence »

d'accroître l'efficacité, de porter un jugement de valeur, de prendre des décisions judicieuses et d'utiliser la connaissance pour le bien commun.

Dans cette section, j'ai défini le concept d'information de manière relative à ceux de donnée, de connaissance et de sagesse. La section suivante s'attache ainsi à donner une définition de la RI et du rôle de ce domaine de l'informatique.

3.1.2 Le rôle

La RI est avant tout un **domaine de l'informatique** dont le but premier est de fournir des **outils et méthodes** permettant à des utilisateurs d'accéder simplement à des informations pertinentes relatives à leurs besoins. Ricardo BAEZA-YATES et Berthier RIBEIRO-NETO donnent la définition suivante de la RI [90] :

Définition 7 (Recherches d'Information [90]) :
« Information retrieval deals with the representation, storage, organization of, and access to information items such as documents, Web pages, online catalogs, structured and semi-structured records, multimedia objects. The representation and organization of the information items should be such as to provide the users with easy access to information of their interest. »
*La recherche d'information traite de la représentation, du stockage, de l'organisation et de l'accès à des « **éléments d'information** » tels que des documents, des pages Web, des catalogues en ligne, des enregistrements structurés et semi-structurés, des objets multimédia. La représentation et l'organisation de ces éléments d'information doivent être telles qu'elles permettent aux utilisateurs d'accéder facilement aux informations qui les intéressent.*

Aujourd'hui, le domaine de recherche de la RI ne se limite plus à l'indexation de textes et à la recherche de méthodes permettant de les sélectionner. Elle regroupe, en effet, aujourd'hui de nombreuses sous-disciplines ayant pour certaines donné lieu à de véritables branches de la recherche en informatique telles que :

- les méthodes de modélisation de l'information ;
- le Web Sémantique ;
- la conception d'interfaces utilisateurs et de visualisation de données ;
- la classification de textes et plus généralement les méthodes d'apprentissage ;
- les langages d'une manière générale incluant aussi bien le champs de recherche du TALN que les langages de requête.

D'un point de vue plus général, la RI est motivée de manière sous-jacente par deux grands types de problématiques à la fois complémentaires et s'impactant mutuellement :

- une problématique purement informatique visant à améliorer les performances et l'efficacité [2] des algorithmes et des outils permettant de rechercher de l'information (e.g. moteurs de recherche, algorithmes de tri, filtrage, index, etc.) ;
- une problématique centrée autour de l'utilisateur davantage « cognitive » et visant à analyser et prendre en compte le comportement et les besoins des utilisateurs.

3.1.3 Le principe

L'ensemble des fonctions nécessaires à la RI sont assurées par des **S**ystèmes de **R**echerche d'**I**nformation (SRIs). Ces derniers reposent sur un entrepôt d'informations généralement dénommé **entrepôt central**. Ces entrepôts sont simplement les bases de données qui maintiennent

2. Dans ce contexte, « les performances » font référence au temps de traitement des algorithmes et outils tandis que « l'efficacité » fait référence à leurs capacités à être à la fois précis et exhaustifs.

les fameux **éléments d'information** de la définition 7. Ils contiennent donc, par exemple, des pages Web dans le cas classique de la RI sur l'Internet, ou encore, des « documents » dans le cadre plus spécifique de la RI bibliographique et documentaire.

Le point d'entrée d'un SRI est, quant à lui, une requête définie par l'utilisateur. Celle-ci se présente généralement sous forme d'une chaîne de caractères et permet à l'utilisateur d'exprimer son besoin d'information. Dans une vision plus large il peut cependant s'agir de formulaires.

L'objectif de tout SRI est alors de **retrouver tous les éléments d'information pertinent vis à vis des besoins exprimés par l'utilisateur à travers sa requête**. La Figure 3.2 illustre le processus de RI à l'aide d'un tel système.

Notation 1 :
Dans la suite de la section 3.1 on notera :

- $E \in \mathbb{N}^*$ *le nombre d'éléments d'information ;*
- $\mathcal{E}$ *l'ensemble de tous ces éléments d'information ;*
- e_i *où* $i \in [\![1\,;\ E]\!]$ *tout élément d'information de* $\mathcal{E}$*.*

de telle sorte que :

$$\mathcal{E} = \{e_1,\ e_2, \cdots,\ e_E\}$$

L'ensemble $\mathcal{E}$ *est donc l'ensemble des éléments d'information du SRI et sera appelé le* ***corpus global****.*

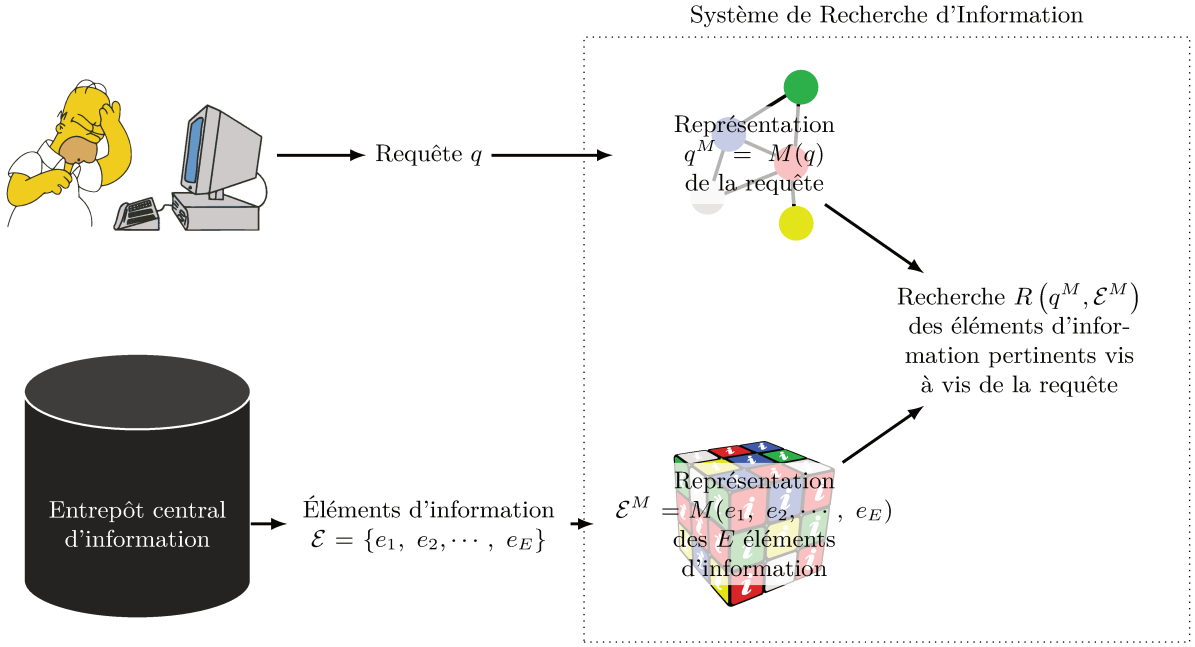


FIGURE 3.2 – Processus de recherche d'information classique.

La définition d'un SRI passe par l'élaboration de deux composants algorithmiques :

Une stratégie de modélisation M : qui permet d'effectuer une **double modélisation** :

- une modèlisation q^M de la requête utilisateur q ;
- une modèlisation $\mathcal{E}^M$ de l'ensemble $\mathcal{E}$ des éléments d'information.

Celle-ci a pour but d'extraire et de donner des représentations informatiquement traitables de l'information qu'ils contiennent. La plupart des SRIs donnent, en effet, accès à des informations qui sont nativement fournies (au moins en partie) sous forme de données non structurées (i.e. essentiellement des blocs de texte). Le rôle de la stratégie de modélisation est alors de **constituer des « objets » synthétiques et représentatifs de l'information contenue dans ces données d'une part, et de l'« intention » exprimée**

à travers les requêtes utilisateurs d'autre part. En somme, cette dernière joue donc un rôle d'**interprétation**.

Une stratégie d'appariement R : qui a pour objectif de mettre en correspondance la représentation informatique de la requête avec celle des éléments d'information (i.e. appariement). Elle vise à **sélectionner les éléments d'information de l'entrepôt central qui sont pertinents vis à vis de la requête utilisateur**. La stratégie d'appariement inclut parfois une étape finale de tri de ces éléments effectuée à l'aide d'une **fonction de classement** [3]. Elle permet d'attribuer un score de pertinence à chaque ressource pertinente identifiée, et ainsi, de présenter en premier lieu à l'utilisateur les éléments d'information les plus pertinents.

Les rôles de ces deux composants algorithmiques sont respectivement de « mettre à disposition de l'information » et de « fournir des méthodes pour la rechercher ».

Cette décomposition des SRIs en deux composants algorithmiques est cependant abstraite. Même si, dans la pratique, ces derniers peuvent être physiquement identifiables, ils n'en demeurent pas moins dépendants l'un de l'autre. L'implémentation d'une stratégie d'appariement se fait ainsi relativement à la modélisation des données choisies et inversement.

Ensemble, la stratégie de modélisation et la stratégie d'appariement définissent le modèle de RI du SRI. Il existe une multitude de types de modèles de RI. Trois grandes classes de ces derniers sont abordées dans la section suivante.

3. : « Ranking function »

3.2 Les modèles de recherche d'information classiques

Ricardo BAEZA-YATES et Berthier RIBEIRO-NETO proposent une taxonomie des différents modèles de RI qui est donnée en Figure 3.3 :

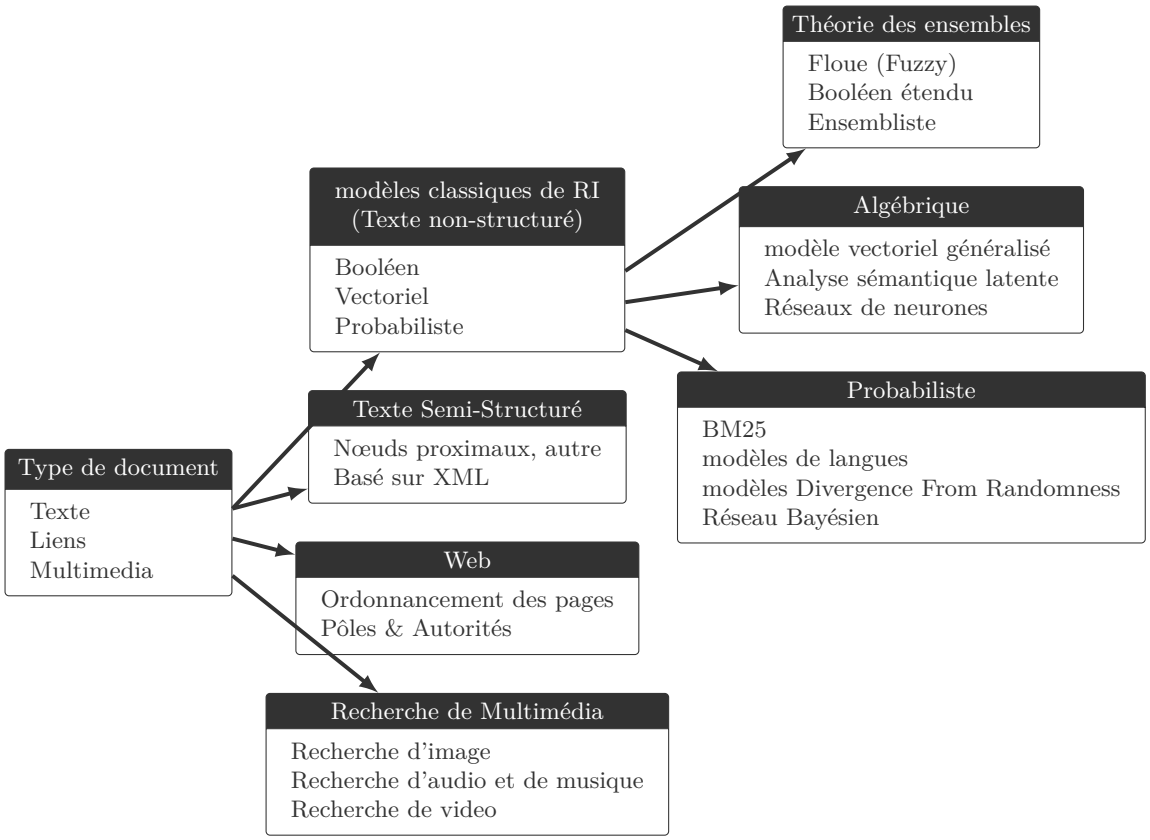


FIGURE 3.3 – Taxonomie des modèles de RI.
Source : *Baeza-Yates and Ribeiro-Neto [90, p. 60]*

Dans le cadre de cette thèse, la RI s'effectue à la fois au sein de données structurées (e.g. données relatives aux patients, données relatives aux analyses biologiques, etc.) et non structurées en ce qui concerne les différents textes cliniques.
On retrouve dans cette taxonomie les trois grandes approches classiques de la RI sur les textes non structurés :

— l'approche Booléenne ;
— l'approche Vectorielle ;
— l'approche Probabiliste.

Ces trois approches seront brièvement abordées dans la suite de ce mémoire. Une attention plus importante sera néanmoins accordée au modèle Booléen compte tenu de son intérêt particulier dans le cadre de la RI au sein d'un EDS.

3.2.1 Le modèle Booléen

Le modèle Booléen est un modèle de RI basé sur la **théorie des ensembles** et l'**algèbre de Boole**. Dans ce modèle, chaque élément d'information $e_i \in \mathcal{E}$ est représenté par une simple conjonction des termes qui apparaissent dans ce dernier. Cet « ensemble de termes » est classiquement nommé **Sac de mots** et constitue l'unique structure sur laquelle repose la stratégie de modélisation. La mise en place de cette stratégie revient à construire un index indiquant la **présence** ou l'**absence** de chaque terme du corpus global (i.e. de l'ensemble $\mathcal{E}$) au sein des différents éléments d'information qui le composent.

Notation 2 :
Dans la suite de la section 3.1 on notera :

- $T \in \mathbb{N}^*$ *le nombre de termes distincts apparaissant dans l'ensemble de tous les éléments d'information ;*
- $\mathcal{T}$ *l'ensemble de tous ces termes ;*
- t_i *où* $i \in [\![1\,;\,T]\!]$ *tout terme de* $\mathcal{T}$.

de telle sorte que :

$$\mathcal{T} = (t_1,\ t_2, \ldots,\ t_T)$$

Cet index associe à chaque terme $t_i \in \mathcal{T}$ et à chaque élément d'information $e_j \in \mathcal{E}$ la valeur binaire 0 ou 1 selon que le terme t_i y est absent ou présent. Cela revient à constituer une matrice de E lignes et T colonnes à valeurs binaires de la forme suivante :

$$\begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ \vdots \\ e_E \end{array} \begin{array}{c} \begin{array}{cccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & \cdots & t_T \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & \cdots & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Lorsque cet index indique la présence d'un terme t_i dans un élément d'information e_j, ce terme constitue alors un **terme indexant** de l'élément d'information e_j et que l'élément d'information e_j est **indexé** avec le terme t_i.

Cette vision binaire est également exploitée pour la représentation logique des requêtes utilisateurs. Ces dernières prennent par conséquent la forme d'**expressions Booléennes**. Les opérateurs Booléens classiques `ET`, `OU` et `NON` permettent de **lier logiquement des termes indexant entre eux**. Chaque terme indexant présent dans une requête désigne ainsi, l'ensemble des éléments d'information indexés avec ce terme. L'opérateur `ET` correspond à l'opération ensembliste d'intersection, l'opérateur `OU` à l'union et l'opérateur `NON` au complémentaire de l'ensemble. Une requête Booléenne est modélisée par une expression ensembliste et l'appariement s'effectue en calculant ces expressions (voir exemple 3).

Exemple 3 :
Soit un entrepôt central composé de 5 *éléments d'information* e_1, e_2, e_3, e_4 *et* e_5 *et* 7 *termes notés* a, b, c, d, e, f *et* g *pouvant potentiellement indexer ces éléments d'information. Le détail des indexations choisies est donné ci-dessous :*

e_1	a	b	c	d	e	f	g
e_2			c	d	e	f	g
e_3	a	b			e		g
e_4	a		c			f	
e_5		b	c				

$$\Leftrightarrow \quad \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Soit la requête Booléenne $q =$ `"a" ET ("b" OU NON "e")`. *Les termes* `"a"`, `"b"` *et* `"e"` *de* q *correspondent alors aux ensembles :*

$$\texttt{"a"} = \{e_1,\ e_3,\ e_4\}$$

$$\texttt{"b"} = \{e_1,\ e_3,\ e_5\}$$

$$\texttt{"e"} = \{e_1,\ e_2,\ e_3\}$$

L'interprétation de la requête Booléenne q en terme d'opérations ensemblistes est alors la suivante :

$$\begin{aligned} q^M &= \{e_1,\ e_3,\ e_4\} \cap (\{e_1,\ e_3,\ e_5\} \cup \complement\{e_1,\ e_2,\ e_3\}) \\ &= \{e_1,\ e_3,\ e_4\} \cap (\{e_1,\ e_3,\ e_5\} \cup \{e_4,\ e_5\}) \\ &= \{e_1,\ e_3,\ e_4\} \cap \{e_1,\ e_3,\ e_4,\ e_5\} \\ &= \{e_1,\ e_3,\ e_4\} \end{aligned}$$

Le SRI renverra donc les éléments d'information e_1, e_3 et e_4 pour la requête q.

Le modèle de RI Booléen est le modèle historique de la RI. Il a été largement exploité notamment dans le cadre des moteurs de recherche bibliographiques. Il constitue un modèle aisé à implémenter et efficace [91]. Son formalisme rigoureux, son exactitude et son aspect « mécanique » et/ou « systématique » constituent à la fois sa principale force et sa principale faiblesse. Il permet, en effet, un requêtage sûr et fin des informations à l'aide de requêtes logiques, précises, facilement composables et « transparentes » dans le sens où la méthode d'exécution employée par le SRI pour l'exécuter est aisée à comprendre pour l'utilisateur. Il n'offre, en revanche, aucune flexibilité compte tenu que les résultats ne répondant qu'approximativement aux requêtes utilisateurs ne sont pas renvoyés. De plus, les termes indexant n'étant pas pondérés, ce type de modèle ne permet pas l'implémentation de fonction de Ranking rendant possible un tri par pertinence des résultats.

3.2.2 Le modèle Vectoriel

Le modèle Vectoriel [92–94] est apparu afin de pallier au manque de flexibilité du modèle Booléen. Ce modèle tire une partie substantielle de ses origines de la conception du SRI **S**ystem for the **M**echanical **A**nalysis and **R**etrieval of **T**ext (SMART) [95] développé dans les années 1960 au sein de l'université privée New-Yorkaise Cornell.

L'idée directrice de ce type de modèle consiste à modéliser les éléments d'information et les requêtes utilisateurs sous la forme de vecteurs. L'idée principale consiste à pondérer les termes de chaque document en fonction de leur importance plutôt que de leur attribuer une valeur binaire relative à leur présence/abscence. La stratégie de modélisation d'un modèle vectoriel repose sur l'attribution d'un poids $p_{i,j}$ à chaque couple $(e_i,\ t_i)$ d'élément d'information et de terme. Ces poids peuvent alors être utilisés comme coordonnée d'un vecteur $\overrightarrow{e_i}$ qui constitue alors la représentation e_i^M de n'importe quel élément d'information $e_i \in \mathcal{E}$. De manière analogue, une représentation vectorielle $q^M = \overrightarrow{q}$ des requêtes utilisateur q peut être obtenue.

La structure de vecteur présente l'avantage de permettre l'exploitation des opérations algébriques classiques des espaces vectoriels et de définir des mesures de pertinence d'un élément d'information vis à vis d'une requête moins drastique que le modèle Booléen. Le principe de base de cette modélisation reste le même et revient à constituer une matrice de E lignes et T colonnes dont les valeurs $p_{i,j}$ indiquent le poids du terme t_j dans l'élément d'information e_i.

$$\begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ \vdots \\ e_E \end{array} \begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & t_3 & t_4 & \cdots & t_T \end{array} \\ \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} & \cdots & p_{1,T} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} & \cdots & p_{2,T} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} & \cdots & p_{3,T} \\ p_{4,1} & p_{4,2} & p_{4,3} & p_{4,4} & \cdots & p_{4,T} \\ p_{5,1} & p_{5,2} & p_{5,3} & p_{5,4} & \cdots & p_{5,T} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{E,1} & p_{E,2} & p_{E,3} & p_{E,4} & \cdots & p_{E,T} \end{pmatrix} \end{array} \begin{array}{l} \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \vdots \\ \rightarrow \end{array} \begin{array}{l} \\ \overrightarrow{e_1} = (p_{1,1},\ p_{1,2}, \cdots,\ p_{1,T}) = e_1^M \\ \overrightarrow{e_2} = (p_{2,1},\ p_{2,2}, \cdots,\ p_{2,T}) = e_2^M \\ \overrightarrow{e_3} = (p_{3,1},\ p_{3,2}, \cdots,\ p_{3,T}) = e_3^M \\ \overrightarrow{e_4} = (p_{4,1},\ p_{4,2}, \cdots,\ p_{4,T}) = e_4^M \\ \overrightarrow{e_5} = (p_{5,1},\ p_{5,2}, \cdots,\ p_{5,T}) = e_5^M \\ \vdots \\ \overrightarrow{e_E} = (p_{E,1},\ p_{E,2}, \cdots,\ p_{E,T}) = e_E^M \end{array}$$

La stratégie d'appariement, quant à elle, repose sur un calcul de similarité entre les différents vecteurs $\overrightarrow{e_i} = (p_{i,1},\ p_{i,2}, \cdots,\ p_{i,T})$ des éléments d'information et celui de la requête $\overrightarrow{q} = (p_{q,1},\ p_{q,2}, \cdots,\ p_{q,T})$. La Figure 3.4 illustre ce principe en trois dimensions :

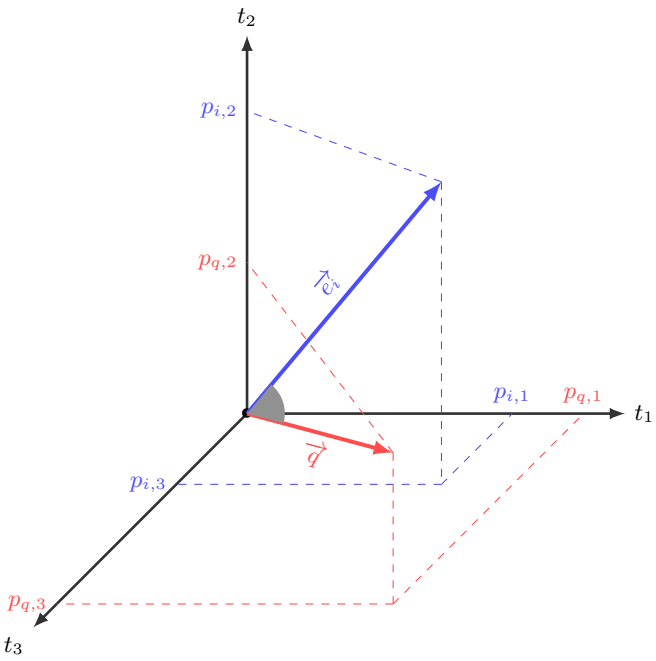


FIGURE 3.4 – Représentation vectorielle d'un document e_i et d'une requête q dans un espace à trois dimensions (i.e. trois termes indexants) t_1, t_2 et t_3

La Figure 3.4 donne, en effet, la représentation vectorielle d'un élément d'information e_i quelconque et d'une requête q dans un espace à trois dimensions (i.e. trois termes potentiels au total dans le corpus global). Les coordonnées du vecteur $\overrightarrow{e_i}$ (resp. $\overrightarrow{q}$) quantifient alors le poids de chacun des trois termes t_1, t_2 et t_3 au sein de l'élément d'information e_i (resp. la requête q).

La similarité entre deux vecteurs de même dimension peut être quantifiée à l'aide de l'angle séparant ces deux vecteurs. Un calcul classique de similarité entre ces vecteurs consiste à évaluer le cosinus de cet angle à l'aide d'un **produit scalaire** :

$$sim(e_i, q) = \frac{\overrightarrow{e_i} \cdot \overrightarrow{q}}{||\overrightarrow{e_i}|| \times ||\overrightarrow{q}||} = \frac{\sum_{k=1}^{T} p_{i,k} \times p_{q,k}}{\sqrt{\sum_{k=1}^{T} p_{i,k}^2} \times \sqrt{\sum_{k=1}^{T} p_{q,k}^2}} \tag{3.1}$$

Il existe cependant d'autres mesures de similarité utilisées dans la littérature telles que :

Le coefficient de Dice :

$$sim(e_i, q) = \frac{2 \times \sum_{k=1}^{T} p_{i,k} \times p_{q,k}}{\sum_{k=1}^{T} p_{i,k}^2 \times \sum_{k=1}^{T} p_{q,k}^2} \tag{3.2}$$

La mesure de Jaccard :

$$sim(e_i, q) = \frac{\sum_{k=1}^{T} p_{i,k} \times p_{q,k}}{\sum_{k=1}^{T} p_{i,k}^2 \times \sum_{k=1}^{T} p_{q,k}^2 - \sum_{k=1}^{T} p_{i,k} \times p_{q,k}} \tag{3.3}$$

La mesure de recouvrement :

$$sim(e_i, q) = \frac{\sum_{k=1}^{T} p_{i,k} \times p_{q,k}}{\min\left(\sum_{k=1}^{T} p_{i,k}, \sum_{k=1}^{T} p_{q,k}\right)} \tag{3.4}$$

Ce calcul de similarité permet non seulement de ne pas rejeter les éléments d'information qui ne correspondent que partiellement à la requête q, mais confère également au modèle vectoriel la possibilité de trier les résultats contrairement au modèle Booléen qui ne fournit aucune fonction de classement.

L'une des problématiques essentielles relatives au modèle de RI vectoriel est la définition des poids $p_{i,j}$ afin qu'ils reflètent l'importance qu'ont les termes au sein des documents. Il a été proposé de considérer la fréquence de ces termes au sein des éléments d'information [96]. Les recherches et expérimentations empiriques menées sur les méthodes de pondération par Sparck Jones [92] et Salton et Yang [93] ont mené à l'élaboration d'une méthode efficace de pondération des termes appelée **T**erm **F**requency–**I**nverse **D**ocument **F**requency (TF–IDF) [4]. La plupart des SRIs basés sur un modèle de RI vectoriel exploitent encore aujourd'hui cette méthode ou l'une de ses variantes. Cette méthode se base sur deux constats :

— l'importance d'un terme $t_j \in \mathcal{T}$ au sein d'un élément d'information $e_i \in \mathcal{E}$ est d'autant plus grande que la fréquence brute [5] $f_{i,j}$ de t_j dans e_i est forte ;

— l'importance d'un terme $t_i \in \mathcal{T}$ d'une manière générale [6], est d'autant plus faible que la fréquence brute [7] n_i de documents dans lesquels t_i apparaît est forte.

Ainsi la méthode TF–IDF définit deux mesures $TF_{i,j}$ et IDF_i qui sont définies dans la définition suivante :

Définition 8 (Term Frequency–Inverse Document Frequency) :
Soit $\mathcal{E} = (e_1,\ e_2, \cdots,\ e_E)$ l'ensemble des éléments d'information et $\mathcal{T} = (t_1,\ t_2, \cdots,\ t_T)$ l'ensemble des termes des éléments d'information de $\mathcal{E}$.

Étant donnés $e_i \in \mathcal{E}$ un élément d'information, $t_j \in \mathcal{T}$ un terme et $f_{i,j}$ le nombre d'occurrences de t_j dans e_i, la ***Fréquence du terme*** *t_j* ***relativement à*** *e_i est définie par :*

$$TF_{i,j} = 1 + \log\left(f_{i,j}\right)$$

Étant donnés $t_i \in \mathcal{T}$ un terme et n_i le nombre d'éléments d'information dans lesquels t_i apparaît, la ***Fréquence inverse de Document du terme*** *t_i est définie par :*

$$IDF_i = \log\left(\frac{E}{n_i}\right)$$

Compte tenu de cette définition, le poids $p_{i,j}$ d'un terme t_j pour un élément d'information e_i est obtenu par :

$$p_{i,j} = \begin{cases} TF_{i,j} \times IDF_i & \text{si } f_{i,j} > 0 \\ 0 & \text{sinon} \end{cases} \tag{3.5}$$

Des variantes des mesures $TF_{i,j}$ et IDF_i ont cependant été proposées dans la littérature [97, 98] :

4. : « **F**réquence du **T**erme–**F**réquence **I**nverse du **D**ocument »
5. L'emploi du mot « fréquence » est abusif. Bien qu'il soit le terme privilégié dans la littérature il s'agit bien d'une fréquence **brute** et donc d'un nombre d'occurrence.
6. i.e. quelque soit l'élément d'information considéré
7. Il s'agit également du nombre d'occurrence n_i de document dans lesquels t_i apparaît et non de la fréquence (en Hertz) à proprement parler).

Variantes $TF_{i,j}$	
Binaire	$\{0,\ 1\}$
Fréquence brute	$f_{i,j}$
Double normalisation $\frac{1}{2}$	$\frac{1}{2} + \frac{1}{2} \times \frac{f_{i,j}}{\max_i f_{i,j}}$
Double normalisation K	$K + (1 - K) \times \frac{f_{i,j}}{\max_i f_{i,j}}$

Variantes IDF_i	
Unitaire	1
Fréquence inverse lissée	$\log\left(1 + \frac{E}{n_i}\right)$
Fréquence inverse max	$\log\left(1 + \frac{\max_i n_i}{n_i}\right)$
Fréquence inverse probabiliste	$\log\left(\frac{E - n_i}{n_i}\right)$

TABLE 3.1 – Variantes des mesures $TF_{i,j}$ et IDF_i

3.2.3 Le modèle Probabiliste

Le modèle de RI probabiliste a été proposé pour la première fois en 1976 par Robertson et Sparck Jones [99]. Le principe directeur de ce type de modèle est d'effectuer un classement des éléments d'information basé sur des probabilités.

Étant donnés une requête q et un élément d'information $e_i \in \mathcal{E}$, il existe un sous-ensemble $\mathcal{R} \subset \mathcal{E}$ d'éléments d'information pertinents vis à vis de q et par conséquent un sous-ensemble $\overline{\mathcal{R}} = \mathcal{E} \setminus \mathcal{R}$ d'éléments d'information non pertinents vis à vis de q. Un modèle probabiliste estime la probabilité que e_i soit pertinent vis à vis de la requête q. L'objectif est ensuite de trier les éléments d'information en fonction d'un ratio de la probabilité.

La stratégie de modélisation exploite une représentation vectorielle binaire des éléments d'information de telle sorte que la représentation e_i^M d'un élément d'information $e_i \in \mathcal{E}$ est un vecteur de dimension T :

$$e_i^M = \overrightarrow{e_i} = (p_{i,1},\ p_{i,2}, \cdots,\ p_{i,T}) \text{ où } \forall j \in [\![1\,;\ T]\!], p_{i,j} = \begin{cases} 1 & \text{si } t_j \text{ apparaît dans } e_i \\ 0 & \text{sinon} \end{cases} \tag{3.6}$$

Les requêtes sont quant à elles représentées comme des sous-ensembles de terme.

La stratégie d'appariement consiste alors à calculer pour toute requête q et tout élément d'information $e_i \in \mathcal{E}$ le score de similarité suivant :

$$sim\,(e_i, q) = \frac{P\,(\mathcal{R} \mid \overrightarrow{e_i})}{P\left(\overline{\mathcal{R}} \mid \overrightarrow{e_i}\right)} \tag{3.7}$$

où :

— $P\,(\mathcal{R} \mid \overrightarrow{e_i})$ est la probabilité que l'élément d'information e_i de représentation $e_i^M = \overrightarrow{e_i}$ soit pertinent vis à vis de q.

— $P\left(\overline{\mathcal{R}} \mid \overrightarrow{e_i}\right)$ est la probabilité que l'élément d'information e_i de représentation $e_i^M = \overrightarrow{e_i}$ soit non pertinent vis à vis de q.

Ce score de similarité peut alors être reformulé à l'aide de la règle de Bayes :

$$sim\,(e_i, q) = \frac{P\,(\overrightarrow{e_i} \mid \mathcal{R}) \times P\,(\mathcal{R}) \times \cancel{P\,(e_i)}}{P\left(\overrightarrow{e_i} \mid \overline{\mathcal{R}}\right) \times P\left(\overline{\mathcal{R}}\right) \times \cancel{P\,(e_i)}} \tag{3.8}$$

Pour une requête q donnée, le ratio $\frac{P(\mathcal{R})}{P(\overline{\mathcal{R}})}$ reste constant quelque soit l'élément d'information e_i considéré. Ce dernier modifie la valeur des différentes mesures $sim\,(e_i, q)$ mais pas l'ordre de

ces derniers. Ainsi, ce ratio peut être ignoré :

$$sim\,(e_i, q) \sim \frac{P\,(\overrightarrow{e_i} \mid \mathcal{R})}{P\left(\overrightarrow{e_i} \mid \overline{\mathcal{R}}\right)} \tag{3.9}$$

où :

— $P\,(\overrightarrow{e_i} \mid \mathcal{R})$ est la probabilité qu'un élément d'information pertinent vis à vis de q possède la représentation $\overrightarrow{e_i}$.

— $P\left(\overrightarrow{e_i} \mid \overline{\mathcal{R}}\right)$ est la probabilité qu'un élément d'information non pertinent vis à vis de q possède la représentation $\overrightarrow{e_i}$.

✲ Notation 3 :
Pour alléger l'écriture de $sim\,(e_i, q)$ on notera par la suite :

— *$x_j = P\,(t_j \mid \mathcal{R})$ la probabilité que le terme t_j apparaisse dans un élément d'information pertinent vis à vis de q.*

— *$y_j = P\left(t_j \mid \overline{\mathcal{R}}\right)$ est la probabilité que le terme t_j apparaisse dans un élément d'information non pertinent vis à vis de q.*

Une mesure de similarité équivalente (i.e. préservant le tri des éléments d'information selon la valeur de cette mesure) et s'exprimant à l'aide des probabilité x_j et y_j peut alors être calculée :

$$sim\,(e_i, q) \sim \sum_{p_{i,j}=1} \log\left(\frac{x_j \times (1 - y_j)}{y_j \times (1 - x_j)}\right) \tag{3.10}$$

Si l'on note :

— n_j le nombre d'éléments d'information qui contiennent le terme t_j ;

— R le nombre d'éléments d'information pertinents vis à vis de q ;

— r_j le nombre d'éléments d'information pertinent qui contiennent le terme t_j.

alors :

$$x_j = P\,(t_j \mid \mathcal{R}) = \frac{r_j}{R} \tag{3.11}$$

$$y_j = P\left(t_j \mid \overline{\mathcal{R}}\right) = \frac{n_j - r_j}{E - R} \tag{3.12}$$

et :

$$sim\,(e_i, q) \sim \sum_{p_{i,j}=1} \log\left(\frac{r_j \times (E - n_j - R + r_j)}{(R - r_j) \times (n_j - r_j)}\right) \tag{3.13}$$

Afin d'éviter les problèmes pour les valeurs extrêmes, le lissage suivant a cependant été proposé :

$$sim\,(e_i, q) \sim \sum_{p_{i,j}=1} \log\left(\frac{(r_j + 0.5) \times (E - n_j - R + r_j + 0.5)}{(R - r_j + 0.5) \times (n_j - r_j + 0.5)}\right) \tag{3.14}$$

Le calcul de cette similarité requiert néanmoins la connaissance et/ou l'estimation des paramètres r_j et R. Le modèle probabiliste a initialement été conçu pour définir ces paramètres manuellement dans le cadre d'un processus récursif. L'idée étant de fixer une valeur initiale de ces paramètres (i.e. pour chaque requête) et de les affiner à l'aide d'itérations successives en évaluant la pertinence des premiers éléments de ressource renvoyés.

Compte tenu du volume d'information mis à disposition des SRIs modernes, cette option est relativement difficile à mettre en application. Une méthode permettant d'automatiser ce processus récursif en l'absence d'informations de pertinence et sans intervention humaine a néanmoins été proposée par Croft et Harper [100]. Cette méthode consiste à initialiser les paramètres x_j et y_j pour chaque terme $t_j \in \mathcal{T}$ comme suit.

$$x_j = 0.5 \text{ et } y_j = \frac{n_j}{E} \tag{3.15}$$

Ainsi la mesure de similarité de base donnée par l'Équation 3.10 devient :

$$sim\,(e_i, q) \sim \sum_{p_{i,j}=1} \log\left(\frac{E - n_j}{n_j}\right) \tag{3.16}$$

Cette dernière peut être calculée pour chaque requête q et chaque élément d'information e_i compte tenu qu'elle ne nécessite aucune autre donnée que le nombre total d'éléments d'information E et le nombre de ces éléments qui contiennent le terme t_j. Un rang peut donc être attribué à chaque élément $e_i \in \mathcal{E}$ relativement à la requête q.

L'« affinage » des paramètres x_j et y_j pour chaque terme t_j peut alors se faire en identifiant parmi les k premiers éléments d'information le nombre r_j^* d'éléments d'information qui contiennent le terme t_j. Les paramètres x_j et y_j peuvent alors être redéfinis en :

$$x_j = \frac{r_j^*}{k} \text{ et } y_j = \frac{n_j - r_j^*}{E - k} \tag{3.17}$$

D'autre redéfinitions de ces paramètres sont également possibles telles que :

$$x_j = \frac{r_j^* + 0.5}{k + 1} \text{ et } y_j = \frac{n_j - r_j^* + 0.5}{E - k + 1} \tag{3.18}$$

ou encore :

$$x_j = \frac{r_j^* + \frac{n_j}{E}}{k + 1} \text{ et } y_j = \frac{n_j - r_j^* + \frac{n_j}{E}}{E - k + 1} \tag{3.19}$$

Le calcul de la similarité $sim\,(e_i, q)$ peut alors de nouveau être réalisé automatiquement pour chaque requête et chaque élément d'information.

Dans cette section, les trois modèles de RI classiques ont été présentés de manière générale. Les données de santé présentent néanmoins des spécificités et requiert des besoins en terme de RI variés. Les caractéristiques de l'information de santé doivent par conséquent être mises en évidence afin de comprendre dans quelle mesure les modèles de RI présentés précédemment peuvent s'y appliquer.

Dans la section suivante, je m'attache à décrire la singularité du domaine médical et des données qui en découlent afin d'expliquer pourquoi la RI au sein de données cliniques requiert des approches plus complexes.

3.3 Le contexte de la santé

L'informatisation des systèmes et le Web ont profondément modifié les pratiques des différentes disciplines quel que soit le domaine concerné. D'un point de vue technologique, ces évolutions ont eu pour principales conséquences de favoriser les échanges et la disponibilité d'informations entraînant, de facto, une « explosion » du volume de ces dernières.

La santé est par nature un domaine dans lequel l'information joue un rôle prépondérant. Ce constat est d'autant plus flagrant en ce qui concerne la médecine. En dépit de l'apparition d'outils de RI et de l'informatisation des pratiques médicales, les professionnels de santé consacrent aujourd'hui toujours autant de leur temps de travail à la gestion d'information [14] que dans les années 1970 [101, 102].

La médecine est une/un « science/art de l'observation » pour laquelle l'information ne joue pas qu'un rôle d'outil mais constitue un élément fondateur et indissociable de celle ci. Cette dépendance de la médecine vis à vis de l'information est notamment concrètement perceptible au sein même de la définition du principe de Médecine Fondée sur les Preuves qui définit aujourd'hui encore la ligne directrice de la pratique médicale moderne [8] :

Définition 9 (Médecine Fondée sur les Preuves) :
D'après Sackett et al. [103] : « Evidence based medicine is the conscientious, explicit, and judicious use of current best evidence in making decisions about the care of individual patients. The practice of evidence based medicine means integrating individual clinical expertise with the best available external clinical evidence from systematic research. By individual clinical expertise we mean the proficiency and judgment that individual clinicians acquire through clinical experience and clinical practice. »
La médecine fondée sur les preuves est l'utilisation consciencieuse, explicite et judicieuse des données les plus probantes dans les prises de décisions relatives aux soins apportés à chaque patient. La pratique de la médecine fondée sur les preuves implique l'intégration de chaque expertise clinique aux meilleures données cliniques externes disponibles issues de recherches systématiques. Par expertise clinique individuelle, nous entendons la compétence et le jugement/discernement que chaque clinicien acquiert par son expérience clinique et sa pratique clinique.

Plus récemment, elle est décrite par Straus et al. [104] comme suit : « Evidence-based medicine (EBM) requires the integration of the best research with our clinical expertise and our patient's unique values and circumstances. »
La médecine fondée sur les preuves requiert l'intégration des recherches les plus probantes à nos (en tant que professionnel de Santé) expertises cliniques et aux valeurs et circonstances uniques de nos (idem) patients.

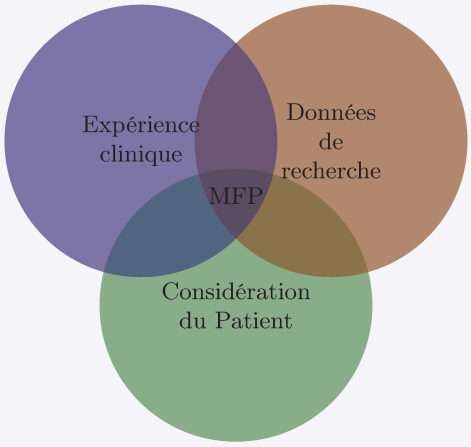


FIGURE 3.5 – Triade de la Médecine Fondée sur le Preuves (MFP)

8. à minima en ce qui concerne la médecine pratiquée majoritairement dans la société occidentale moderne

Ainsi, la RI peut intervenir au moins à deux niveaux dans le cadre de l'informatique médicale :

Au niveau des données de recherche : ces données proviennent généralement de la recherche médicale fondamentale, des recherches cliniques ou encore de tests diagnostiques. Elles se matérialisent concrètement en connaissances issues en partie de littérature scientifique, des guides de bonne pratique ou tout autres documents de références. Dans ce contexte, les méthodes de RI classique, présentées dans la section précédente (cf. section 3.2), et que l'on pourrait qualifier de « RI documentaire », s'avèrent pertinentes.

Au niveau de l'expérience clinique : l'expérience clinique se fonde en partie sur l'observation des expériences passées et de l'analyse des bénéfices, des causes et conséquences, des décisions et des actions médicales engagées par le passé. Dans le cadre de la pratique de la médecine, l'accès à l'ensemble des signes cliniques des patients et la comparaison de ces derniers avec des cas comparables permet au clinicien de comprendre les mécanismes des maladies et de le guider dans l'attitude thérapeutique à adopter [105]. À ce titre, une RI plus proche de la donnée, fournissant un accès à l'historique médical d'une population de patients présente un fort intérêt à la pratique médicale et s'intègre, précisément, dans la problématique de RI au sein des DPIs et des EDSs.

La mise en pratique de ces RIs se heurte néanmoins à des spécificités et des obstacles propres au milieu de la santé. Certaines solutions à ces problèmes ont été apportées, notamment dans le cadre de la recherche documentaire. En revanche, en ce qui concerne la RI au sein des données spécifiques relatives aux patients, les progrès ont été plus lents et de nombreux challenges restent à relever.

3.3.1 La recherche d'information textuelle en Santé

La santé est un domaine qui se subdivise en une multitude de spécialités diverses et variées. Comme d'autres, ce domaine est constitué de divers spécialistes, dont la façon de travailler diffère et ayant adopté un langage propre à leurs disciplines. Un des objectifs communs à tous ces professionnels reste, néanmoins, le bien être du patient. À ce titre, une discipline de transmission et de communication des informations s'impose à ces derniers et de larges volumes d'informations hétérogènes ont toujours été traditionnellement générés dans le domaine de la santé.

Cette « tradition » de l'information dans le domaine de la santé présente la particularité de s'accompagner d'un langage naturel particulièrement vaste, riche, sophistiqué et pour lequel les outils de RI classique peuvent s'avérer inefficaces. Ce langage se compose de termes et d'expressions de concepts médicaux complexes entretenant des liens sémantiques forts entre eux et ayant une signification précise. Afin d'optimiser la communication, les professionnels de santé ont, de plus, développé des mécanismes de langage tels que l'usage régulier d'acronymes, de phrases courtes et formatées, de notations ou encore une organisation concise des différentes informations favorisant leur acquisition systémique.

Afin de palier à la trop grande diversité du « langage médical », le domaine de la santé fait l'usage de **terminologies** (et d'**ontologies**) de santé ou plus généralement de **SOCs**. Ces structures informatiques fournissent des **concepts terminologiques et ontologiques** qui **servent à représenter et à identifier de manière rigoureuse et organisée** : les concepts, les idées, les notions, ou tout simplement les objets du domaine médical [54]. Elles établissent ainsi une **hiérarchie** entre les concepts terminologiques ainsi que des **relations dites intra-terminologiques** qui confèrent à ces derniers une sémantique.

L'emploi des terminologies permet non seulement une **description sémantique de l'information de santé** mais joue, également, un rôle primordial dans la **standardisation** de cette description. Il est, en effet, le garant de l'interopérabilité des données et des systèmes.

Les terminologies de santé sont notamment largement exploitées pour la RI bibliographique. Une approche classique consiste à munir les SRIs bibliographiques d'une stratégie de modélisation basée sur ces concepts terminologiques. Cette approche requiert évidemment une étape

d'**indexation des éléments d'information** (i.e. indexation contrôllée [9]), mais offre des possibilités de RI étendues comparées à la représentation habituelle des éléments d'information par l'ensemble des termes qu'ils contiennent réellement (i.e. indexation libre). Une indexation peut s'effectuer manuellement ou par l'intermédiaire d'une indexation automatique supervisée par un humain. Dans ce dernier cas des annotateurs sémantiques sont utilisés pour extraire et identifier les concepts terminologiques présents au sein des éléments d'information. Une fois extraits, ces concepts constituent alors la représentation des éléments d'informations qui sera exploitée dans la stratégie d'appariement. Dans la suite de cette section, la terminologie MeSH, dont une description plus détaillée est donnée dans l'Annexe A, est essentiellement utilisée pour illustrer les propos. Cependant, le domaine de la santé fait état de nombreux SOCs (e.g. 75 TOs pour HeTOP, $\geqslant$ 130 TOs en anglais pour le méta-thésaurus de l'UMLS). Ces derniers ont été conçus dans des objectifs différents et sont plus ou moins spécifiques à un domaine particulier. Par exemple, la terminologie **Orphanet** Rare Disease Ontology (orphanet) est spécialisée dans les maladies rares et l'ontologie **F**oundational **M**odel of **A**natomy (FMA) est spécialisée dans l'anatomie. La terminologie SNOMED–CT®, qui s'apparente à une ontologie, est quant à elle beaucoup plus générique et permet une expressivité beaucoup plus importante. Certaines terminologies sont de plus davantage destinées à assurer une interopérabilité des systèmes qu'à être utilisées dans le cadre d'outils de RI. C'est par exemple le cas de la CCAM qui permet le codage des actes médicaux au sein des établissements de santé français [10].

De nombreux travaux ont été effectués autour des SOCs dans le domaine de la santé. L'UMLS est notamment composé d'un lexique, d'un réseau sémantique et d'un méta-thésaurus largement utilisé dans le domaine de la santé. Il permet d'agréger une multitude de TOs de manière cohérente en établissant des alignements entre les différents concepts de ces dernières. Il incorpore notamment la CPT, la CIM–10, le MeSH, la SNOMED–CT®, LOINC, la **W**orld **H**ealth **O**rganization **A**dverse Drug **R**eaction **T**erminology (WHO–ART), **NORM**alized names for Clinical Drugs (RxNorm), la Gene Ontology (GENEONTOLOGY) [11], OMIM. De nombreux annotateurs sémantiques ont de plus été développés afin d'automatiser la représentation des ressources bibliographiques.

L'une des plus importantes mises en application de ce principe dans le domaine de la santé est la base de données bibliographiques **MED**ical **LI**terature analysis and Retrieval System On**LINE** (MEDLINE®) [12]. MEDLINE® est gérée et maintenue par la NLM. Elle est consultable à distance depuis 1972. Cette dernière répertorie des citations (i.e. des titres et des résumés) d'articles issus de la la littérature scientifique relatifs au domaine biomédical. Depuis janvier 1996, le moteur de recherche PubMed (PubMed) [13] fournit un ensemble de fonctionnalités de RI permettant d'y accéder. Ce dernier donne un accès à plus de 29 millions de citations provenant de MEDLINE®, de livres en ligne ou de revues spécialisées dans les sciences de la vie. Il est aujourd'hui devenu une référence en terme de recherche bibliographique dans le domaine Biomédical d'une manière générale. Dans PubMed, et comme dans la plupart des bases de données de la NLM [106], les ressources bibliographiques sont indexées à l'aide du thésaurus MeSH. Le moteur de recherche DocCiSMeF [21], développé au sein de l'équipe, utilise également cette terminologie de santé comme terminologie pivot d'indexation des ressources (bien qu'il exploite également d'autres TOs du portail HeTOP).

Le MeSH, comme la plupart des terminologies de santé, fournit une arborescence de concepts terminologiques structurés. L'exploitation de telles structures permet d'implémenter des fonctionnalités de RI beaucoup plus riches. Une des fonctionnalités la plus connue est l'expansion hiérarchique. Elle permet d'étendre les recherches de l'utilisateur en recherchant les ressources indexées avec les termes hiérarchiquement inférieurs. Par exemple, si un utilisateur saisit le libellé

9. C'est donc une indexation reposant sur l'utilisaion d'un vocabulaire **contrôllé**.
10. Les libellés de cette terminologie sont en effet parfois extrêmement longs ce qui les rend peu susceptibles d'être saisis par un utilisateur (e.g. « *Appendicectomie avec toilette péritonéale pour péritonite aigüe généralisée, par coelioscopie ou par laparotomie avec préparation par coelioscopie* » [$HHFA025$ (CCAM)])
11. url : `http://geneontology.org/`
12. : « Système d'analyse et de recherche de documentation médicale en ligne »
13. url : `https://www.ncbi.nlm.nih.gov/pubmed/`

du concept terminologique « *cardiopathies* » [*D*006331 (MeSH)] alors le SRI pourra également retourner les ressources indexées avec les concepts terminologiques « *anévrysme cardiaque* » [*D*006322 (MeSH)], « *bas débit cardiaque* » [*D*002303 (MeSH)], etc.. De même les libellés alternatifs (i.e. synonymes) fournis par les terminologies de santé pour les différents concepts terminologiques permettent une recherche plus flexible dans le sens où, par exemple, les ressources indexées avec « *cardiopathies* » [*D*006331 (MeSH)] peuvent également être recherchées avec les synonymes « maladies cardiaques » ou « maladies du cœur ». Enfin, dans le cadre du thésaurus MeSH, les qualificatifs permettent également une description plus fine des ressources et donc une RI plus précise (cf. Annexe A). Il est alors possible d'indexer une ressource à l'aide d'un couple (Concept terminologique, qualificatif) et de rechercher les ressources indexées avec une telle association.

D'une manière générale, l'exploitation de concepts Terminologiques et/ou Ontologiques comme support de description de ressources bibliographiques peut être vu comme une volonté d'apporter une sémantique à la modélisation des ressources bibliographiques. Afin que cette sémantique puisse être pleinement exploitée, les SRIs bibliographiques et documentaires adoptent le plus souvent un modèle de RI Booléen ou vectoriel muni d'un langage de requête Booléen spécifique dont l'expressivité va au delà de l'expressivité Booléenne habituelle. Ils conservent, en effet, l'utilisation des opérateurs Booléens mais fournissent généralement des éléments de syntaxe visant à exploiter une plus grande granularité de représentation des ressources. Il permettent généralement de requêter indépendamment les différents champs descriptifs (i.e. méta-données) des ressources tels que le titre, le résumé, l'auteur ou encore la date et bien sûr de maximiser l'exploitation de l'expressivité sémantique apportée par l'indexation terminologique des ressources. On retrouve cette tendance dans de nombreux SRIs bibliographiques (e.g. PubMed, LiSSa, LILACS [107], **B**anque de **D**onnées en **S**anté **P**ublique (BDSP),[14] Cochrane Library,[15] EMBASE ,[16] etc.).

Exemple 4 (requête Booléenne augmentée) :
Dans le cadre du moteur de recherche PubMed, il est par exemple possible de constituer des requêtes du type :

```
"asthma in children"[TIAB] AND "asthma/drug therapy"[MH]
          AND "Review"[PT] AND "Chipps BE"[AU]
```

où :

- `"asthma in children"[Title/Abstract]` *permet de rechercher les ressources contenant* `"asthma in children"` *dans le titre ou le resumé.*
- `"asthma/drug therapy"[MeSH Terms]` *permet de rechercher les ressources traitant de l'asthme dans le cadre plus spécifique du traitement médicamenteux.*
- `"Review"[Publication Type]` *permet de rechercher les ressources qui correspondent à des revues de la littérature.*
- `"Chipps BE"[Author]` *permet de rechercher les ressources dont l'auteur est* `"Chipps BE"`

La complexité du domaine biomédical et des données qui en proviennent a amené à modéliser l'information de manière plus granulaire et organisée. Afin de tirer parti de cette structuration plus précise, des modes de requêtage plus avancés sont proposés par les outils de RI bibliographiques. La RI en santé présente également la particularité d'utiliser de nombreuses TOs qui offrent une description et une recherche sémantique des contenus textuels en santé. Bien qu'indispensables, ces méthodes ne permettent pas de répondre à la totalité des cas d'usages de la RI au sein des données de santé.

14. url : `http://www.bdsp.ehesp.fr/Base/`
15. url : `https://www.cochranelibrary.com/`
16. url : `https://www.elsevier.com/solutions/embase-biomedical-research`

3.3.2 La recherche d'information au sein de données cliniques

L'**information clinique** est définie comme l'information nécessaire à l'amélioration de la pratique clinique et de son efficacité [108]. Elle se matérialise essentiellement par des données relatives aux patients ou par des connaissances médicales **organisées** et **utiles pour la prise de décision médicale** [109, 110]. L'information clinique se distingue de la notion de connaissance médicale, dont la transmission est assurée par le biais de littérature scientifique. Elle peut néanmoins en résulter. L'établissement d'un diagnostic constitue, par exemple, une information clinique à part entière bien que ce dernier résulte de l'application de connaissances médicales.

L'un des aspects fondamentaux de l'information clinique réside dans son extrême hétérogénéité. Comme le montre intrinsèquement la section 2.2, l'information clinique dans son ensemble est transmise à travers de nombreuses données qui sont de natures différentes, produites dans des contextes et des objectifs de santé variés et qui concernent différents aspects de santé des patients. Cette multiplicité des données tend à s'amplifier avec l'apparition des objets connectés et des applications qui leurs sont dédiées depuis le début des années 2010 (e.g. Google Fit, Apple Health etc.).

L'un des défis majeurs de la RI dans les données cliniques réside dans l'aptitude des SRIs à mettre en relation ces dernières. Les questions que les professionnels de santé peuvent être amenés à se poser sont, en effet, souvent complexes et nécessitent la contextualisation et la mise en relation de diverses données et informations cliniques relatives aux patients [48]. La précision du domaine médical a toujours fait de l'expression des besoins d'information une problématique à part entière de la RI en santé [111–114].

La structuration des données et la modélisation des informations qui en découlent permet de favoriser la précision de la RI mais également sa spécificité. Il semble, en effet, cohérent que le choix d'une modélisation entraîne l'adoption d'une vision particulière de l'information et conditionne ainsi l'étendue et les caractéristiques des fonctionnalités de RI qui pourront être fournies. Dans le cadre de l'information clinique, cette modélisation reste néanmoins sujette à plusieurs limitations. La diversité des cas d'usages de l'information clinique requiert une modélisation générique de cette dernière. De plus, les professionnels de santé restent parfois rétissants à produire des données structurées notamment en raison de l'investissement en temps et de la perte d'expressivité que peut représenter cette production [115]. Une grande quantité de l'information clinique, et plus généralement de l'information de santé, existe ainsi sous forme de données non structurées. La structuration a posteriori de l'information de santé reste ainsi la norme. Elle peut être partiellement effectuée via l'exploitation de TOs comme outils de représentation sémantique de l'information de santé. C'est d'ailleurs le principe sur lequel repose la RI documentaire et bibliographique. Bien qu'essentiel, ce principe ne permet cependant pas de répondre pleinement à la problématique de mise en relation contextuelle des informations cliniques. Comparativement au cas d'usage de la RI bibliographique, la RI au sein de données cliniques s'appuie sur un ensemble d'information dont l'organisation conceptuelle est plus complexe (cf. Figure 3.6).

Un SRI documentaire ou bibliographique ne permet de manipuler qu'un seul et même type de ressource. Bien que des champs descriptifs additionnels (e.g. titre, auteur, etc.) puissent éventuellement accompagner ces ressources, ils s'apparentent conceptuellement à de simples méta-données ou attributs de la ressource principale. A contrario, les données cliniques ne se limitent pas à une collection uniforme de ressources. Elles font intervenir de multiples notions médicales disjointes (e.g. patients, comptes-rendus, analyses biologiques, etc.). D'un point de vue conceptuel, il est donc plus naturel de représenter l'information clinique dans sa globalité sous forme de multiples entités interconnectées qu'à l'aide d'un objet structuré unique global. En d'autres termes, l'information clinique peut s'apparenter à un **graphe de données**.

L'utilisation des graphes comme modèle de représentation des informations de santé est devenu de plus en plus courante ces dernières années [116] et a permis de s'adresser à des problématiques diverses, notamment dans le cadre de l'extraction d'informations [117, 118]. Ces innovations ont, par ailleurs, largement été portées par celles du Web sémantique [27] et du modèle de graphe RDF. Il faut néanmoins noter que, conceptuellement, l'utilisation de graphes

comme formalisme de représentation des connaissances a débuté dès les années 1980 avec les graphes conceptuels [119, 120].

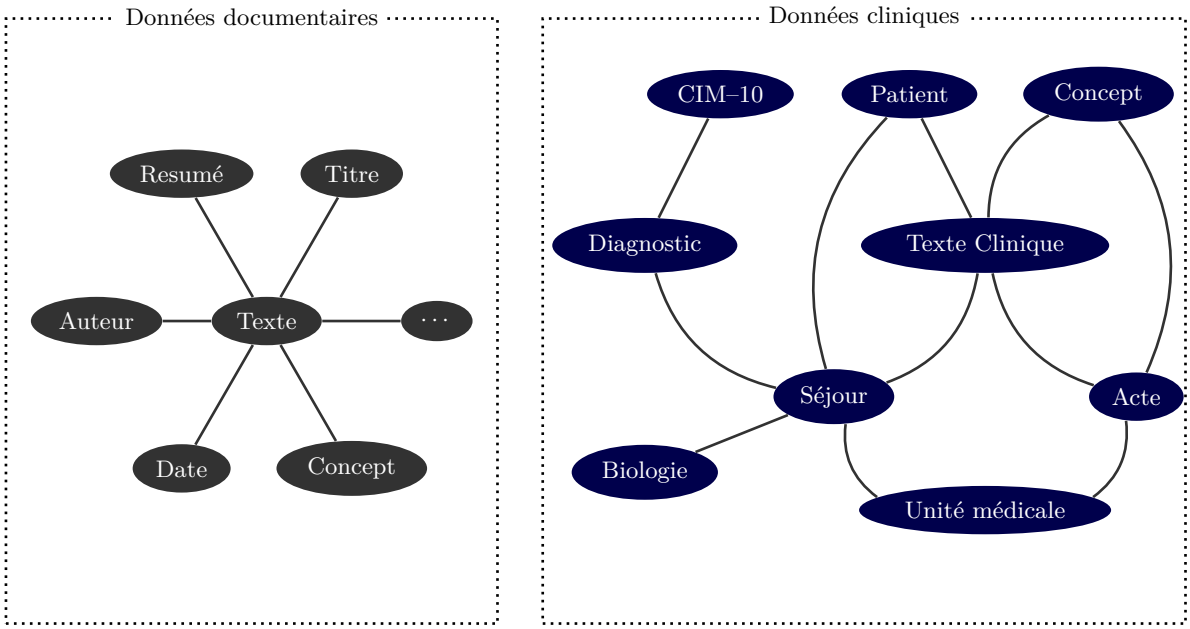


FIGURE 3.6 – Différence structurelle entre l'information documentaire et l'information clinique

La structure de graphe permet une modélisation générique de l'information. Elle engendre la migration d'un « univers mono-entité », au sein duquel l'information est vue comme un ensemble de ressources structurellement fixes et munies de méta-données, vers un univers « multi-entités » dans lequel divers types d'informations coexistent et sont en relation logique définissant une sémantique. Cette structure semble ainsi apporter suffisamment de flexibilité pour répondre à un large panel de cas d'usages relatifs à l'exploitation d'informations cliniques. Cependant, ce constat ne suffit pas à lui seul à répondre aux problématiques pratiques de RI au sein d'un EDS. La question d'une implémentation théoriquement cohérente et opérationnellement viable d'un « graphe de données cliniques » reste une « pierre angulaire » de ces dernières. La définition rigoureuse d'un modèle de graphe permettant de s'accommoder opérationnellement avec les volumétries considérables des données d'un EDS reste donc à définir. Dans le chapitre suivant, je m'attache à développer certains éléments du Web sémantique. Ce domaine apporte en effet déjà des réponses à la problématique de données interconnectées notamment par le biais de la RI sémantique. Le modèle sémantique définit, en effet, des unités sémantiques (concepts, instances de concepts, etc.) qui peuvent être reliées par des relations (relations hiérarchiques ou rôles) [121] au sein d'un graphe sémantique. Le Web sémantique généralise de plus cette vision notamment par l'intermédiaire du modèle de graphe **RDF**. Les travaux effectués autour du Web sémantique présentent donc un intérêt particulier dans le cadre de mes travaux de thèse.

Chapitre 4

Les méthodes de stockage de graphes de données

Sommaire

4.1 Le Web de données 91
4.1.1 Le Resource Description Framework 91
4.1.2 Ontologies 92
4.1.3 SPARQL Protocol And RDF Query Language 94
4.1.4 Synthèse 96
4.2 Les bases de données alternatives 97
4.2.1 Le paysage des bases de données alternatives 97
4.2.2 À chaque problématique son changement de paradigme 100
4.2.3 Synthèse 109

Le **Web Sémantique** [122] est une **extension du Web** standardisée par le **W**orld **W**ide **W**eb **C**onsortium (W3C®).[1] Comme son nom l'indique, l'ambition première du Web Sémantique est, de donner du « sens » à l'information du Web en rendant la « **connaissance** » contenue dans cette dernière à la fois exploitable et accessible informatiquement via les technologies du Web. En 2001, Tim BERNERS-LEE, James HENDLER et Ora LASSILA donnaient la définition 10 du Web semantic.

Ce standard préconise l'emploi de formats de données et de protocoles d'échanges standardisés visant avant tout à la **structuration** et au **partage** de l'information issue du Web.

Le Web Sémantique s'appuie sur des concepts théoriques et plus particulièrement sur une **représentation de la connaissance** rendant possible l' **interconnexion** des différentes informations présentes sur Internet. L'une des idées sous-jacentes au Web sémantique est en réalité de faire de cette information un véritable réseau de **données structurées**. Cet objectif s'inscrit dans une initiative du W3C® désignée par le terme de « Web des données » et vise à terme à faire du Web lui même un « Global Giant Graph [2] ».

Les données de santé contenues dans un EDS étant par nature interconnectées, les concepts clés relatifs au domaine du Web Sémantique apparaissent donc pertinents vis à vis des problématiques de RI au sein de ces dernières. Bien que le Web Sémantique s'articule spécifiquement autour des technologies du Web, les problématiques auxquelles il permet de répondre en terme de données sont en effet comparables à celles rencontrées par les systèmes autonomes et indépendants.

Dans ce contexte, le D2IM développe depuis plusieurs années un modèle de données générique inspiré des fondements théoriques du Web sémantique qui sera par ailleurs décrit dans le chapitre 5 p. 111. Ce dernier est notamment compatible avec le modèle ontologique du point

1. url : `https://www.w3.org/`
2. 🇫🇷 : « Graphe Global Géant »

de vue de la représentation des informations mais ne permet en revanche pas l'exploitation de mécanismes d'inférence. À titre d'exemple, cette compatibilité procure notamment à HeTOP la capacité d'intégrer des ontologies formelles telles que FMA ou HPO, ou quasi formelles comme SNOMED–CT® (OWL-DL).

En plus de fournir une représentation générique de l'information, le Web sémantique s'accompagne également d'outils potentiellement exploitables dans le cadre de la RI. L'un des aspects fondamentaux du Web sémantique est en effet son langage de requête : le **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage (SPARQL). Ce dernier permet de requêter les données avec une grande expressivité exploitant pleinement le formalise de représentation de l'information.

Le Web sémantique fournit un cadre théorique et opérationnel potentiel à la RI au sein d'un EDS. Dans cette section, les fondements et outils du Web sémantique seront rappelés afin d'évaluer leur utilité dans le cadre de la problématique de cette thèse.

Comme évoqué dans la chapitre 2, les SGBDRs ne permettent pas l'obtention de performances acceptables dans le cadre d'un accès aux informations d'un EDS. Dans la section 4.2, un état de l'art des SGBDs alternatifs existants à l'heure actuelle est dressé afin d'évaluer dans quelle mesure ces derniers peuvent s'avérer exploitables dans le cadre de la mise en place d'une RI au sein d'un EDS.

Définition 10 (Le Web Semantic par Berners-Lee et al. [122]) :
« The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users. [...] The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. [...] For the semantic web to function, computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning. [...] Two important technologies for developing the Semantic Web are already in place : eXtensible Markup Language (XML) and the Resource Description Framework (RDF). [...] XML allows users to add arbitrary structure to their documents but says nothing about what the structures mean. [...] The Semantic Web will enable machines to COMPREHEND semantic documents and data, not human speech and writings. [...] Meaning is expressed by RDF, which encodes it in sets of triples, each triple being rather like the subject, verb and object of an elementary sentence. »
Le Web sémantique structurera le contenu signifiant des pages Web, en créant un environnement où les agents logiciels itinérants d'une page à l'autre peuvent facilement exécuter des tâches complexes pour les utilisateurs. [...] Le Web sémantique n'est pas un Web séparé mais une extension du Web actuel, dans lequel l'information reçoit un sens bien défini, permettant aux ordinateurs et aux personnes de mieux travailler en coopération. [...] Pour que le Web sémantique fonctionne, les ordinateurs doivent avoir accès à des collections structurées d'informations et à des ensembles de règles d'inférence qu'ils peuvent utiliser pour mener un raisonnement automatisé. [...] Deux technologies importantes pour le développement du Web sémantique sont déjà en place : le langage XML (eXtensible Markup Language) et le RDF (Resource Description Framework). [...] XML permet aux utilisateurs d'ajouter une structure arbitraire à leurs documents mais ne dit rien sur la signification de ces structures. [...] Le Web Sémantique permettra aux machines de COMPRENDRE des documents et des données sémantiques, et non la parole ou les écrits humains. [...] Le sens est exprimé par RDF, qui le code en triplets, chaque triplet étant un peu comme le sujet, le verbe et l'objet d'une phrase élémentaire.

4.1 Le Web de données

4.1.1 Le Resource Description Framework

Les technologies et les fondements théoriques du Web sémantique s'inscrivent dans une évolution de la manière de concevoir l'information, de la représenter et de l'utiliser. Elles apportent une réponse à la volontée de **structuration** des données et de leur **disponibilité**. Dans le cadre du Web sémantique, il ne s'agit plus d'adopter une vision au sein de laquelle la page Web est vue comme la ressource principale du Web mais plutôt de migrer vers une vision où le Web permet de considérer des ressources correspondant à des objets, des concepts, des personnes, des animaux et d'une manière générale à tout ce qui **existe dans le réel**. Pour identifier ces ressource, le Web sémantique exploite des **U**niform **R**esource **I**dentifier (URI) et des **I**nternationalized **R**esource **I**dentifier (IRI). Le Web Sémantique s'est donc construit autour de cette volonté de faire de l'information présente sur Internet une interconnexion d'entités symboliques multiples (i.e. un graphe de données) et non plus de ressources physiques mono-typées (i.e. une toile de pages Web). Pour cela, le Web Sémantique repose sur le modèle **R**esource **D**escription **F**ramework (**RDF**).

RDF est un **modèle de graphe** développé par la W3C. En d'autres termes, il permet de « d'exprimer » de l'information sous forme d'un graphe de données. Une description **RDF**, quant à elle, constitue une retranscription de cette modélisation dans un langage spécifique.

RDF a été conçu avant tout dans l'objectif de décrire des ressources Web et leurs métadonnées afin de les partager sur Internet. Cependant, les concepts fondateurs de ce modèle ainsi que de ceux qui l'accompagne n'en restent pas moins applicables et dignes d'intérêt pour d'autres domaines d'application que le Web.

Le modèle **RDF** permet une modélisation d'information sous forme d'un **multigraphe orienté étiqueté** :

Multigraphe : Plusieurs arrêtes peuvent éventuellement coexister entre deux même nœuds du graphe.

Orienté : Les arrêtes correspondent à des arcs « orientés ». Autrement dit, les arrêtes du graphe possèdent un « sens ». L'un des deux nœuds reliés par l'arrête est clairement identifié comme le nœud source tandis que l'autre est identifié comme le nœud cible.

Étiqueté : Des libellés sont assignés aux arrêtes et aux nœuds du graphe.

D'un point de vue extérieur, un graphe possède une structure « redondante » composée simplement de nœuds, symbolisant des entités d'information, reliées par des arcs symbolisant des relations entre ces entités. Le principe de base du modèle **RDF** s'appuie sur cette simplicité en décomposant l'information globale sous forme de **triplets RDF** de la forme :

$$(sujet,\ prédicat,\ objet)$$

Un triplet **RDF** constitue l'atome de connaissance de toute description **RDF** ainsi que la « brique de base » à partir de laquelle ces descriptions sont construites. Visuellement, un tel triplet correspond simplement à un arc reliant un nœud source à un nœud cible. Ainsi, au sein d'un triplet **RDF** :

Le *sujet* représente une **ressource/entité/nœud** du graphe **RDF** que le triplet **RDF** permet de décrire.

Le *prédicat* représente un type de **propriété** de l'entité *sujet* décrite par ce triplet **RDF**. Au sein d'un graphe **RDF**, il identifie un arc dont le nœud source est *sujet*.

L'*objet* représente soit une donnée, soit une autre ressource (entité/nœud) qui constitue alors la **valeur** de la propriété *prédicat* applicable à l'entité *sujet*. Dans le graphe **RDF**, il constitue le nœud cible de l'arc *prédicat* partant du nœud source *sujet*.

La Figure 4.1 donne un fragment de graphe dans lequel sont identifiés deux triplets **RDF**.

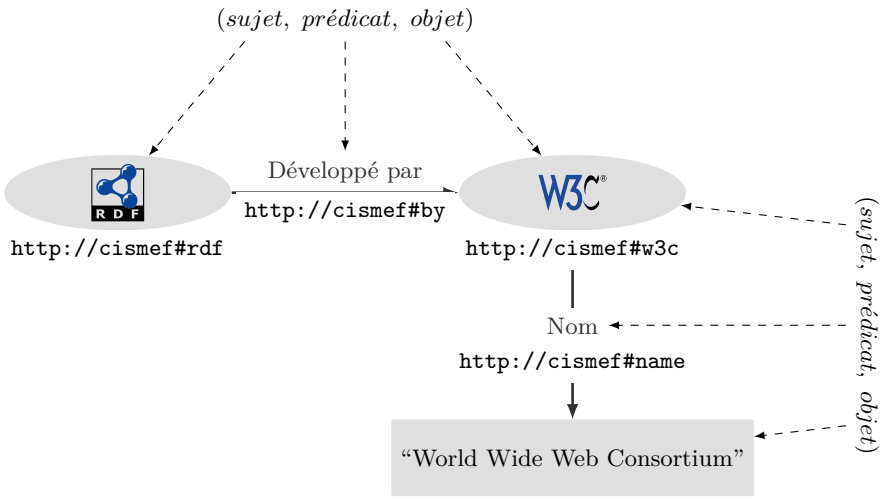


FIGURE 4.1 – Illustration de triplets **RDF**. L'information représentée par ces triplets est « *Le modèle* **RDF** *est développé par la* W3C *dont le nom est* ***W****orld* ***W****ide* ***W****eb* ***C****onsortium* ». Cette information est représentée à l'aide de deux arcs et trois nœuds. Conformément à la convention, les nœuds de ressource/entité sont représentés sous forme d'ellipses et les donnée sous forme de rectangles.

Un document structuré en **RDF** est donc constitué d'un ensemble de triplets **RDF** qui dans leur ensemble définissent intégralement un graphe **RDF** et l'information qu'il représente. Le modèle **RDF** repose pleinement sur la notion d'URI/IRI qui est utilisée dans les descriptions **RDF** comme identifiant des diverses entités, relations et propriétés référencées dans les triplets.

4.1.2 Ontologies

Définition 11 (ontologie) :
Gruber [123] donne la définition suivante d'une ontologie :
« A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. [...] An ontology is an explicit specification of a conceptualization. »
Une conceptualisation est une vue abstraite et simplifiée du monde que l'on veut représenter. [...] Une ontologie est la spécification d'une conceptualisation.

En 2001, dans le cadre du Web sémantique, Berners-Lee et al. [122] en donne à sont tour une description :
« In philosophy, an ontology is a theory about the nature of existence, of what types of things exist ; ontology as a discipline studies such theories. [...] an ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has a taxonomy and a set of inference rules. The taxonomy defines classes of objects and relations among them. [...] Classes, subclasses and relations among entities are a very powerful tool for Web use. We can express a large number of relations among entities by assigning properties to classes and allowing subclasses to inherit such properties. [...] Inference rules in ontologies supply further power. »
En philosophie, une ontologie est une théorie sur la nature de l'existence, sur les types de choses qui existent ; l'ontologie, en tant que discipline, étudie ces théories. [...] une ontologie est un document ou un fichier qui définit formellement les relations entre des termes. Typiquement, une ontologie pour le Web définit une taxonomie et un ensemble de règles d'inférence. La taxonomie définit des classes d'objets et des relations entre elles. [...] Les classes, sous-classes et relations entre entités sont un outil très puissant dans le cadre de l'utilisation du Web. Nous pouvons exprimer un grand nombre de relations entre entités en assignant des propriétés aux classes et en permettant aux sous-classes d'hériter de ces propriétés. [...] Les règles d'inférence des ontologies fournissent une puissance suplémentaire.

Les **Ontologies** sont apparues pour incorporer des **informations de sens** au sein des descriptions **RDF** de base. L'objectif principal d'une Ontologie est de rendre possible une « **inférence** » ou plus simplement d'automatiser certains **raisonnements** et **déductions** portant sur les données **RDF**.

D'un point de vue structurel, les informations supplémentaires apportées par une Ontologie ne diffèrent pas réellement des données **RDF** de base. Ces dernières sont en effet également représentées sous forme de triplets **RDF**. Dans la pratique, une Ontologie ne constitue donc pas une « augmentation » du modèle de représentation de l'information défini par le modèle de graphe **RDF** mais plutôt un moyen d'établir une distinction conceptuelle claire et rigoureuse entre les descriptions qui relèvent de l'information concrète d'une part et celles qui relèvent de l'information sémantique (plus abstraite) d'autre part. Plus spécifiquement, les Ontologies permettent d'ajouter une couche sémantique à l'information existante dont le rôle est de décrire le **vocabulaire** exploité dans ces descriptions.

Le langage **R**esource **D**escription **F**ramework **S**chema (RDFS) permet de décrire (en **RDF**) des Ontologies dites « légères ». Il permet de définir un schéma constituant un premier niveau d'information sémantique basé sur la notion de **classe**. Les différentes ressources mises en relation au sein des triplets **RDF** n'étant alors plus vues comme de simples éléments indépendants mais comme des instances de classes et donc des objets typés. De manière synthétique, le langage RDFS permet notamment de typer des ressources éventuellement de manière hiérarchique, de définir les propriétés de ces types ainsi que des prédicats qui permettent de les relier au sein des triplets **RDF** (e.g. en définissant par exemple les types « acceptables » comme ressources sources et cibles de ces prédicats).

Tout comme RDFS, le Web Ontology Language (OWL) est un langage de représentation des connaissances basé sur le modèle de graphe **RDF**. Il introduit cependant des niveaux de description sémantique plus expressifs que ce dernier. Il permet de définir des classes par l'intermédiaire de contraintes logiques empruntées des travaux sur les logiques de description. Une classe peut alors être définie comme union, disjonction, union disjointe, complément ou encore intersection de deux classes. OWL permet de définir des contraintes permettant d'assurer ou de vérifier la cohérence des données et surtout de réellement déduire de nouvelles connaissances des données. OWL permet de définir une **sémantique formelle** des données. Dans sa première version ce langage se déclinait notamment en trois sous-langages qui étaient, du moins expressif au plus expressif :

— OWL-Lite ;

— OWL-DL (DL faisant référence à **Description Logics**[3]) ;

— OWL-Full.

Les algorithmes permettant d'inférer de la connaissance à partir de propriétés logiques descriptibles à l'aide de ces trois sous-langages étant à la fois de plus en plus coûteux et de moins en moins décidables. Ainsi OWL-Lite est Décidable est permet l'obtention d'une réponse pour toute requête en temps raisonnable, OWL-DL est également décidable mais en temps exponentiel tandis que OWL-Full est indécidable et aucun logiciel de raisonnement n'est capable d'effectuer un raisonnement complet pour ce dernier.

Dans sa deuxième version, Web Ontology Language (Seconde version, 2009) (OWL2) définit trois profils : OWL2-EL, OWL2-QL et OWL2-RL. Néanmoins, toutes les ontologies OWL-Lite sont des ontologies OWL2, donc OWL-Lite peut être considéré comme un profil de OWL2. de même, OWL-DL peut également être considéré comme un profil de OWL2. Les trois profils d'OWL2 sont des sous-langages (sous-ensembles syntaxiques) d'OWL2-DL qui sont définis comme une restriction syntaxique de la spécification structurelle OWL2. En d'autres termes, ils constituent des sous-ensembles des éléments structuraux qui peuvent être utilisés dans une ontologie conforme et sont plus restrictifs qu'OWL2-DL. Ils « troquent » différents aspects de l'expressivité d'OWL en échange de différents avantages en termes de calcul et/ou de mise en

3. : « Logique de Description »

œuvre (Figure 4.2 p. 94) :

— OWL2-EL permet d'exécuter les problèmes de raisonnement en temps polynomial par rapport à la taille de l'ontologie ;

— OWL2-QL est conçu de manière à ce qu'une réponse complète soit apportée aux requêtes dans un espace de taille logarithmique [4] (plus précisément dans la classe de complexité AC^0) par rapport à la taille des données (assertions, Ensemble de axiomes Assertionnels ($\mathcal{A}Box$)). Il est conçu pour que les requêtes portant sur les données (assertions) stockées dans un SGBDR standard puissent être interrogées par le biais d'une ontologie via un simple mécanisme de réécriture de celles-ci en requêtes SQL auxquelles le système SGBDR peut ensuite répondre sans aucune modification des données.

— OWL2-RL permet la mise en œuvre d'algorithme de raisonnement en temps polynomiale par l'intermédiaire d'un système de règles opérant directement sur les triplets RDFS. La sémantique de ce profil est en partie donnée par cet ensemble de règles qui étendent l'interprétation RDFS des graphes RDF valides.

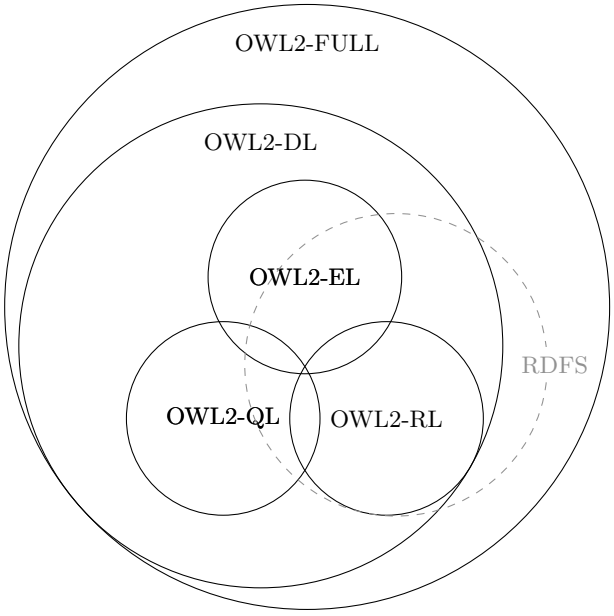


FIGURE 4.2 – Expréssivité des langages ontologiques OWL2 et RDFS [3]

4.1.3 SPARQL Protocol And RDF Query Language

Le modèle RDF ainsi que ses différentes sérialisations permettent de formaliser et d'écrire des données afin de rendre le partage et l'inter-connexion de ces dernières possible sur le Web. En revanche, l'**accès à ces données** est assuré par le SPARQL. Ce dernier constitue à la fois un **protocole** et un **langage de requête**. Sa version 1.0 est devenue une recommandation officielle du W3C en 2008.

Le protocole SPARQL repose sur l'architecture du Web et donc sur le protocole HTTP. Plus concrètement, les données du Web sémantique sont stockées au sein de bases de données, appelées « **Triplestore** [5] », spécifiquement conçues et optimisées pour le stockage et la récupération de triplets RDF.

Le langage SPARQL permet lui, une sélection et une modification de ces données RDF alors contenues dans les triplestores. Une requête SPARQL peut-être transmise à un triplestore par l'intermédiaire d'une requête HTTP pure (Figure 4.3) ou éventuellement par l'intermédiaire d'une requête SOAP. La réponse du triplestore quant à elle peut être fournie sous divers formats

4. : « LOGSPACE »
5. : « Magasin de Triplets RDF »

comme par exemple XML, RDF/XML, JSON, **C**omma-**S**eparated **V**alues (CSV), etc.

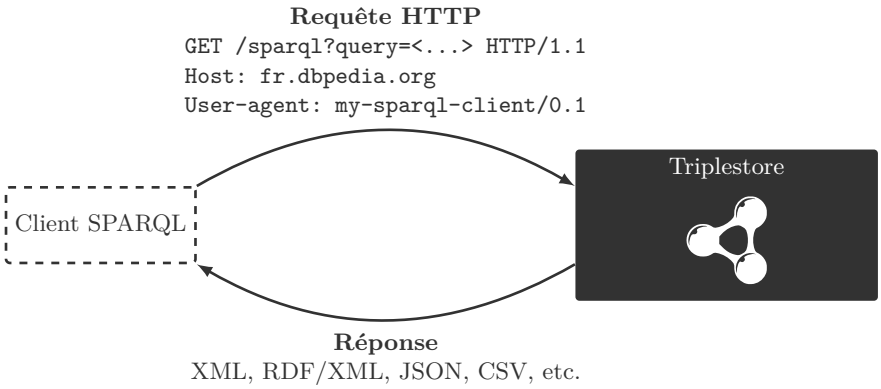


FIGURE 4.3 – Protocole SPARQL repose sur une architecture client serveur. La requête SPARQL est encapsulée dans une requête HTTP transmise au Triplestore. Ce dernier peut alors renvoyer sa réponse sous différents formats.

Le langage de requête SPARQL est au Web Sémantique ce que le SQL est aux SGBDRs. Les requêtes SPARQL possèdent en effet une forme similaire aux requêtes SQL notamment parce qu'elles se construisent à l'aide des trois clauses de base « `SELECT` », « `FROM` » et « `WHERE` ». La structure de base d'un triplestore étant le triplet **RDF**, une requête SPARQL spécifie les données à extraire sous forme de triplets **RDF** définissant ainsi des « patrons de graphe » au même titre que la syntaxe d'une requête SQL s'articule autour de la notion de table et de colonne. La Figure 4.4 fournit un exemple de requête SPARQL se rapportant à la portion de graphe **RDF** de la Figure 4.1. La syntaxe du SPARQL utilise les mêmes abréviations et raccourcis syntaxiques que de la syntaxe Turtle.

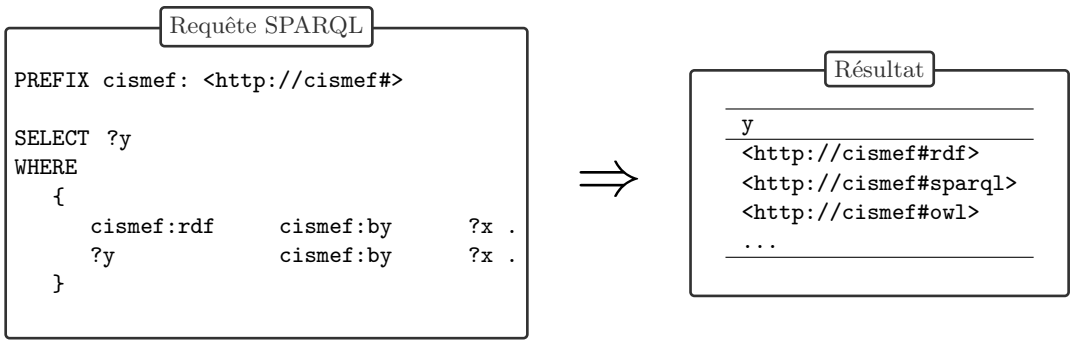


FIGURE 4.4 – Exemple simple d'une requête SPARQL basée sur le graphe **RDF** de la Figure 4.1. Cette requête permet de retourner les URIs des standards développés par le W3C.

Dans cette requête, les deux lignes contenues dans la clause `WHERE` définissent des triplets. Ce sont les URIs associées aux nœuds et arcs du graphe **RDF** qui sont utilisées pour décrire ces triplets. Le premier triplet permet de désigner une variable « `?x` » comme le nœud cible de toute relation d'URI `http://cismef#by` dont le nœud source est `http://cismef#rdf`. En se rapportant à la Figure 4.1, cela signifie que « `?x` » correspond au nœud d'URI `http://cismef#w3c` symbolisant le W3C. Le deuxième triplet permet, quant à lui, de définir la variable `?y` comme tout nœud source d'une relation de type `http://cismef#by` et de nœud cible `?x`. La requête SPARQL dans son ensemble permet de déterminer l'ensemble des standards développés par la même organisation qui assure le développement du standard **RDF**, c'est à dire le W3C.

Le langage de requête SPARQL permet également d'insérer et de supprimer des triplets

RDF d'un triplestore ou même de créer des nouveaux graphes (« SPARQL Update »). Il fournit également beaucoup de fonctionnalités permettant de construire des contraintes complexes. Parmi ces fonctionnalités on peut notamment citer : les motifs de chemins, les opérateurs de conjonction et de disjonction, la gestion du typage des données, les opérateurs algébriques, le tri des résultats, la limitation du nombre de résultat ou encore les fonctions prédéfinies (e.g. expressions régulières, Cast (i.e. conversion de type), concaténation, etc.).

4.1.4 Synthèse

D'un point de vue théorique, le Web sémantique fournit une réponse à la problématique de données cliniques inter-connectées mise en évidence dans le chapitre précédent. Il fournit un cadre théorique rigoureux et générique de représentation de l'information qui se concrétise à travers le modèle de graphe **RDF**.

Il en est, a priori, de même sur le plan opérationnel. Les technologies permettant la mise en pratique de ce modèle existent. Plus précisément, le stockage et l'accès aux informations sont rendus possibles par l'intermédiaire des triplestores et du langage de requête SPARQL. Ce dernier permet de parcourir avec une grande généricité les graphes définis à l'aide du modèle **RDF** et de définir avec finesse les portions à extraire de ce dernier. Ainsi, le Web sémantique, semble réunir tous les éléments nécessaires à la RI au sein d'un EDS.

Le Web sémantique vise cependant des objectifs qui dépassent la simple recherche de données brutes. Sa motivation principale est de fournir, à travers la définition d'Ontologies, des mécanismes d'inférence à partir de données représentées en **RDF**. Ces derniers sont mis en application par les moteurs d'exécution des requêtes SPARQL qui sont intégrés aux triplestores et qui peuvent être paramétrés afin de réaliser cette inférence avec des niveaux de profondeur différents. Bien qu'ils ouvrent des perspectives de RI intéressantes, ils ne permettent pas nécessairement une sélection des données cliniques en temps réel. On distingue historiquement deux types de triplestores :

Les triplestores basés sur un SGBDR : Ils ajoutent une couche **RDF** à un SGBDR existant. Ces triplestores souffrent ainsi des mêmes limitations que les SGBDRs quant à la montée en charge des données et sont donc peu adaptés à la problématique de données cliniques.

Les triplestores natifs : Ils sont spécifiquement construit pour le modèle de données **RDF** et qui offrent, par conséquent, de meilleurs performances d'accès aux données.

La comparaison des performances des différents triplestores est aujourd'hui un sujet d'intérêt et plusieurs jeux de données et/ou procédures d'études comparatives ont vu le jour afin de fournir un environnement solide de comparaison (benchmarks) : le Lehigh University Benchmark (LUBM) [124], le Berlin SPARQL Benchmark (BSBM) [125], SP2Bench [126] ou encore DBpedia Benchmark [127, 128]. De plus, ces dernières années, des initiatives visant à améliorer les performances des triplestores par le biais de l'utilisation de solutions NoSQL comme système de stockage ont été menées (e.g. [129–133]). Peu d'études des performances des ces systèmes ont été menées, cependant, l'utilisation de SGBDs NoSQL semble prometteuse dans le cadre de données liées. Dans [134], une comparaison de ces triplestores NoSQL avec un tripletore natif (i.e. 4store [135]) est effectué à l'aide des différents benchmarks cités précédemment. Dans cette étude montre notamment que l'utilisation de solutions NoSQL apporte un gain de performances dans le cadre de requêtes simples et qu'elle offre des perspectives d'optimisation des requêtes plus importantes que les triplestores classiques de même que de montée en charge.

Les technologies sur lesquelles repose le Web sémantique sont néanmoins destinées au stockage de l'information au sein du réseau Web. L'accès à l'information implique donc un transit de données via le protocole HTTP qui ne permet pas l'obtention de performances optimales dans le cadre d'un EDS autonome et isolé. Les technologies NoSQL présentent, toutefois, des caractéristiques adaptées des données interconnectées volumineuses qu'il apparaît donc pertinent d'objectiver. Dans la section suivante, je réalise un état de l'art des divers types de bases de données alternatives qui ont vu le jour ces dernières années. Ce dernier permettra d'évaluer la pertinence de ces systèmes dans une contexte de RI au sein de données cliniques.

4.2 Les bases de données alternatives

4.2.1 Le paysage des bases de données alternatives

Contrairement au concept de SGBDR qui se définit à partir d'une théorie scientifique rigoureuse et identifiable (i.e. l'algèbre relationnel), la notion de **base de données alternative** est quant à elle davantage historique ce qui en fait un concept complexe à définir. Il regroupe aujourd'hui différents types de SGBDs qui sont certes, pour une partie, non–relationnels mais qui reposent néanmoins sur des modèles de données, des caractéristiques techniques et des modes d'interrogation variés [136]. Bien que le terme NoSQL soit régulièrement employé pour désigner ces nouvelles solutions dans leurs ensembles, on peut distinguer aujourd'hui trois grands types de SGBD alternatif :

Les SGBD NoSQL : Ces bases de données sont avant tout conçues pour remplir les critères de scalabilité des architectures distribuées. On distingue au sein de cette famille différents types de solutions dont une description plus détaillée est donnée plus loin :

- les bases NoSQL orientées Document ;
- les bases NoSQL orientées Colonne ;
- les bases NoSQL orientées Clés – Valeur ;
- les bases NoSQL orientées Graphe.

Les SGBD NewSQL : Elles correspondent à des bases de données majoritairement relationnelles mais basées sur de nouveaux moteurs de stockage, de nouvelles technologies transparentes de fragmentation et de nouveaux matériels. Ces bases maintiennent les propriétés **A**tomicité, **C**ohérence, **I**solation et **D**urabilité (ACID) (contrairement aux SGBDs NoSQL) et le langage de requête SQL. Elles sont conçues pour répondre aux problématiques de scalabilité des architectures distribuées ou pour améliorer significativement les performances de telle sorte qu'une scalabilité horizontale n'est plus nécessaire. Ces bases n'ayant pas été abordée en pratique dans le cadre de ce travail de thèse, le sujet ne sera pas davantage abordé dans la suite de ce mémoire.

Les Caches et Data Grids mémoires : Ces bases de données stockent les données en mémoire (i.e. dans la **R**andom– **A**ccess **M**emory (RAM) [6]) afin de garantir des performances d'accès optimum. Elles couvrent un certain nombre de besoins en terme de manipulation de données (e.g. mise en cache de données préalablement persistées ou non, réplication de données au sein d'une architecture distribuées, calcul exploitant le Grid etc.). Ces solutions sont souvent assimilées aux solutions NoSQL et se rapprochent de ces dernières du fait de leurs caractéristiques majeures communes en terme de :

- distribution des données et de leurs traitements ;
- modélisation des données (i.e. les données sont stockées dans des tables de hachage de manière identique aux solutions NoSQL orientées Clé – Valeur) ;
- gestion de la cohérence des données (i.e. tous deux sont assujettis aux mêmes règles inhérentes aux architectures distribuées quant au respect des propriété ACID).

La RI au sein d'un EDS requiert, à la fois, la capacité de gérer d'importants volumes de données mais aussi, un accès rapide à ces dernières afin de garantir une utilisabilité du SRI. Certains points techniques différencient les caches & Data Grids mémoires des solutions NoSQL. Dans le cadre de mon travail de thèse, le point pertinent de divergence entre ces systèmes réside dans les objectifs pour lesquels chacun de ces systèmes ont été conçus. Les solutions NoSQLs sont conçus pour manipuler « toujours plus de données » alors que les caches et Data Grids distribués le sont pour accéder « toujours plus vite à ces données ».

Il n'existe pas de définition viable et rigoureuse globalement acceptée de ces bases de données ni même d'autorité particulière en ayant fourni une en ce qui concerne les bases de données de

6. : « Mémoire Vive »

type NoSQL [137]. Les bases de données alternatives sont nées sous l'impulsion des grandes entreprises de l'Internet à partir du début des années 2000. Ces « géants de l'Internet », contraints de manipuler non seulement des volumes de données énormes mais aussi des données toujours plus hétérogènes et interconnectées se sont alors vu confrontés aux limites des SGBDs relationnels et transactionnels qui s'étaient depuis leurs apparitions dans les années 1970 imposés comme le paradigme de SGBD dominant. En réponse à ces problématiques, ces acteurs ont commencé à développer leurs propres SGBDs. On peut notamment citer à ce titre ***Google*** avec ***BigTable*** [138], ***Facebook*** avec ***Cassandra***, ***Amazon*** avec ***Dynamo*** [139] ou encore ***SourceForge.net*** avec ***MongoDB***. L'emploi même de l'acronyme NoSQL (signifiant « **N**ot **o**nly **SQL** », c'est à dire « pas seulement SQL » en français) est lui même historique et largement critiqué pour son manque de cohérence qui génère quelques confusions (certain SGBDR tel que Postgres, Oracle ou encore SQLServer ne sont pas restreints au SQL alors qu'ils ne sont conventionnellement pas inclus dans la famille des SGBDs NoSQL). Le terme « NoSQL » a d'ailleurs été utilisé pour la première fois par Carlo STROZZI pour désigner son SGBDR qui, bien que n'exploitant pas le SQL comme langage de requête, était ironiquement tout de même basé sur le modèle relationnel [140]. Ce dernier terme a par la suite été popularisé grâce à sa reprise comme nom d'un « Meetup » organisé le 11 juin 2009 à San Fransisco par Johan Oskarson. L'objectif de ce dernier était de réunir différents acteurs s'étant appuyés sur des systèmes de stockage de données open-source, distribués et non-relationnels suite à la vague d'inspiration générée par l'exemple de BigTable (Google) et Dynamo (Amazon).

Ces SGBDs n'avaient alors pas été nécessairement pensés comme des substituts aux SGBDRs mais comme des solutions parallèles pouvant coexister avec ces derniers. Ce principe de cohabitation a alors été nommé ***persistance polyglotte*** par Scott LEBERKNIGHT dans le milieu des années 2000 en référence au principe de ***programmation polyglotte*** de Neal FORD qui visait à faire cohabiter différents langages au sein d'une même application afin de tirer partie des forces de chacun d'entre eux dans le cadre d'applications spécifiques.

La notion de SGBD NoSQL et plus généralement de base de données alternative s'est donc construite autour de ces nouveaux types de SGBDs qui répondaient à des problématiques économico-techniques et non suite à l'élaboration d'une nouvelle théorie.

Dans son rapport de 2011 [141], la société ***451 Group*** identifie 6 facteurs clés menant à l'adoption d'une base de données alternative :

La scalabilité : Nécessite de pouvoir mettre en place une architecture distribuée économiquement plus rentable et assurant une meilleurs flexibilité.

Les performances : Nécessité de meilleures performances avec la montée en charge des données (limitation des SGBDRs).

Relâchement de la cohérence : Nécessité de « relâcher » les contraintes garantissant la cohérence des données pour garantir une constante disponibilité du système.

Agilité : principe de persistance polyglotte.

Complexité : problématique de quantité et complexité des données (i.e. problématique de big data).

Nécessité : Échec des solutions déjà existantes à atteindre les objectifs de performance, de scalabilité et de flexibilité imposant l'adoption rapide d'une solution open-source.

la société ***451 Group*** établie, également, régulièrement des schémas de classification destinés à mettre en lumière le paysage des bases de données alternatives. Une reproduction non exhaustive de ces schémas est donnée Figure 4.5.

L'alternative actuelle la plus sûre pour éclaircir la notion de SGBD NoSQL reste donc de définir les problématiques économico-techniques auxquelles les SGBDs communément acceptés au sein de cette famille permettent de répondre.

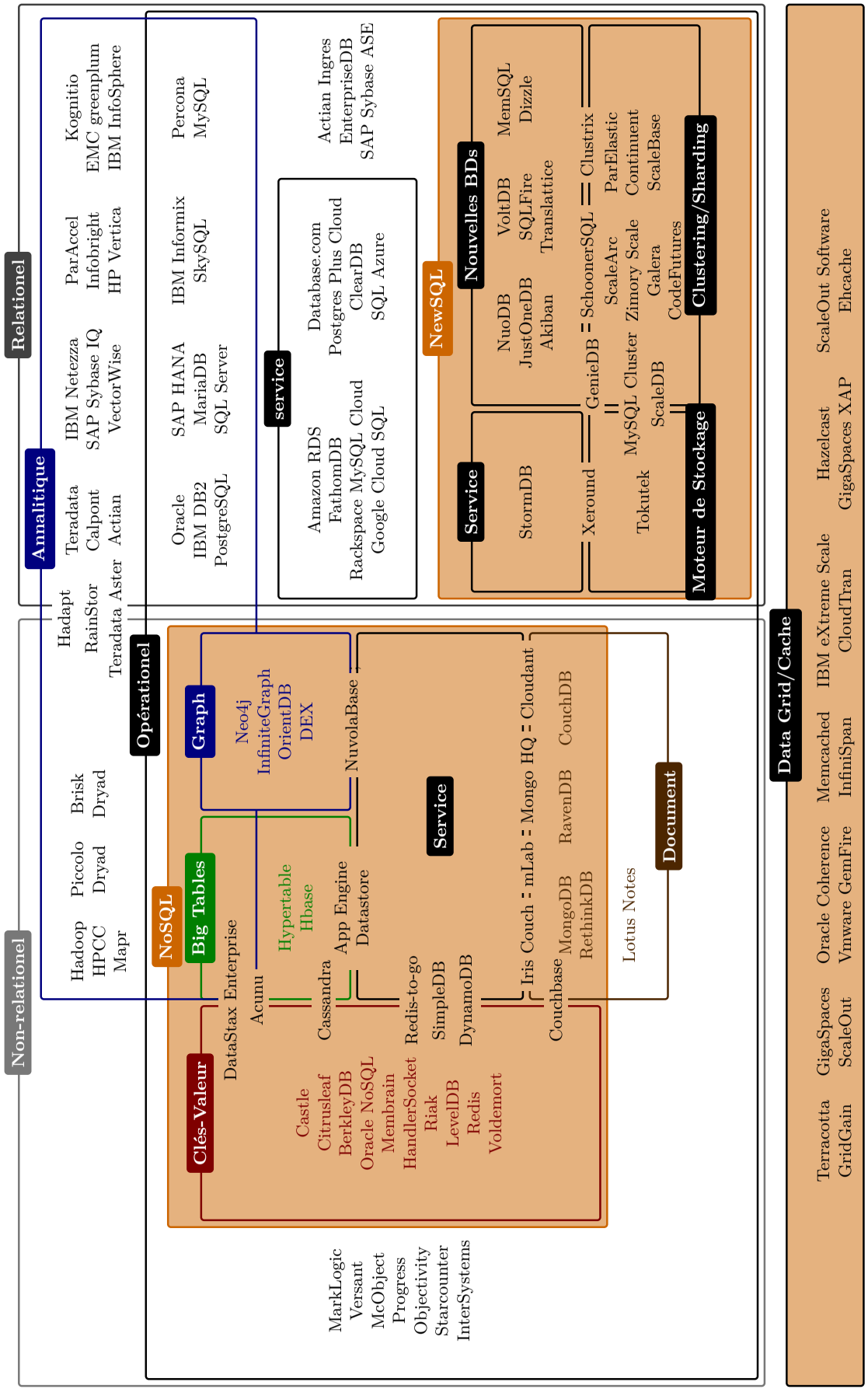


FIGURE 4.5 – Schéma illustrant le paysage des bases de données alternatives

Source : *Reproduction et adaptation de schémas édités par la société 451 Group*

4.2.2 À chaque problématique son changement de paradigme

4.2.2.1 Volumes conséquents de données vs. Données distribuées

L'une des problématiques techniques majeures et historiques ayant conduit à l'essor des SGBDs alternatifs et la nécessité d'une **architecture distribuée**.

Les SGBDRs fournissent des mécanismes de consistance des données qui en font des systèmes principalement conçus pour fonctionner sur un nœud unique imposant l'achat de coûteuses machines évolutives (i.e. possibilité d'ajout de processeurs, de RAM, de disques etc.) afin de gérer la montée en charge des données (i.e. principe de scalabilité verticale).

La croissance du volume de données à manipuler couplée à la baisse des prix des équipements informatiques a poussé les grandes plates-formes du Web à vouloir adopter une architecture en grappes de serveur (i.e. cluster) afin d'effectuer une distribution à différents niveaux :

La distribution des données elles-mêmes : Dans le cadre des problématiques de montée en charge des données, l'exploitation de grappes de serveur (i.e. cluster) et la distribution des données sur les différents nœuds de ces clusters permet à la fois la maîtrise de la charge et des performances d'accès aux données pour chaque nœud et une absorption aisée de la montée en charge des données par simple ajout de serveurs (i.e. scalabilité horizontale).

La distribution des traitement de ces données : Elle permet de décomposer les traitements en sous-tâches applicables sur les sous-jeux de données contenus par les différents nœuds du cluster et de les exécuter parallèlement sur ces machines distinctes. Cette distribution permet ainsi une réduction des temps de traitement globaux des données notamment via l'exploitation d'algorithmes dédiés (e.g. algorithme MapReduce).

Les SGBDs de type NoSQL et Data Grid répondent notamment à cette problématique de distribution bien que celle-ci ne soit pas systématique dans le cadre de l'utilisation de ces SGBDs. Ces systèmes offrent donc de meilleures performances d'accès aux données.

4.2.2.2 Données inter-connectées vs. Agrégats de données

Le méta-modèle relationnel (celui des SGBDRs) « éclate » les informations afin de les organiser en termes de relations et de tuples (viz. tables, colonnes et lignes). Avec la montée en charge de la complexité structurelle des données, ce paradigme d'organisation est rapidement devenu contraignant dans la pratique :

Manque de flexibilité : Le modèle relationnel requiert une pré-définition de la structure des données (i.e. définition des tables et des colonnes). La redéfinition a posteriori de la structure des données sur un SGBDR bien que réalisable est une tâche lourde et sensible ce qui rend les SGBDRs peu aptes à prendre en compte les éventuels changements de structure des données.

Contexte distribué : Dans un contexte distribué, l'éclatement des informations complexifie la distribution des données sur plusieurs serveurs ainsi que leur intégrité. Afin de garantir efficacement l'intégrité des données dans un contexte distribué, il est nécessaire que les données et celles qui lui sont liées par des relations d'agrégation (qui sont de surcroît indiscernable informatiquement des relations classiques) soient présentes sur un même nœud du cluster.

Afin de répondre à ces problématiques de flexibilité et surtout de distribution, une grande partie des solutions NoSQLs ont adopté des modèles de représentation variés mais ayant pour caractéristique commune la capacité de stocker des ***agrégats de données***. Plus précisément, les SGBDs NoSQLs orientés Document, Colonne et Clé-Valeur adoptent un **modèle orienté agrégat**. Ce type de modèle autorise le stockage de données complexes, agrégeant et imbriquant les informations logiquement liées au sein d'une même unité de traitement. Ces agrégats indiquent l'unité de consultation aux SGBDs et permettent la distribution de ces derniers sur

un cluster tout en conservant des mécanismes assurant l'atomicité au niveau de l'agrégat dans son ensemble. Ces systèmes n'imposent pas systématiquement la pré-définition de la structure des données.

Dans la suite les quatre types classiques de SGBDs NoSQLs sont présentés.

4.2.2.2.1 (🔑) SGBD NoSQL orienté Clé-Valeur

Ils constituent les solutions NoSQLs les plus simples en termes de modèle de données et d'utilisation. Ces systèmes stockent les données au sein de tableaux associatifs (i.e. des structures de données de type table de hachage) sous forme de couple $(clé, valeur)$. La structure de la valeur n'est pas contrainte, ne requiert pas de pré-définition (contrairement au méta-modèle relationnel) et peut ainsi correspondre à n'importe quel agrégat de données. L'accès aux valeurs et leurs insertions se fait par l'intermédiaire de la clé qui identifie de manière unique cette dernière valeur. L'avantage premier de ces systèmes réside dans les très bonnes performances d'accès aux données qu'ils offrent en raison :

— de l'accès systématique aux données par clé primaire imposé par la structure de données (i.e. table de hachage) exploitant une fonction de hachage appliquée sur la clé ;

— du stockage en cache (i.e. RAM) des données pour une part importante de ces SGBDs.

En revanche, ce modèle de représentation ne modélise aucunement l'agrégat et la structure de ce dernier n'est pas « visible » par le SGBD. Il appartient donc à l'application cliente de connaître à l'avance la structure des données stockées et d'en exploiter le contenu. Une illustration de ce modèle de représentation est donnée dans Figure 4.6.

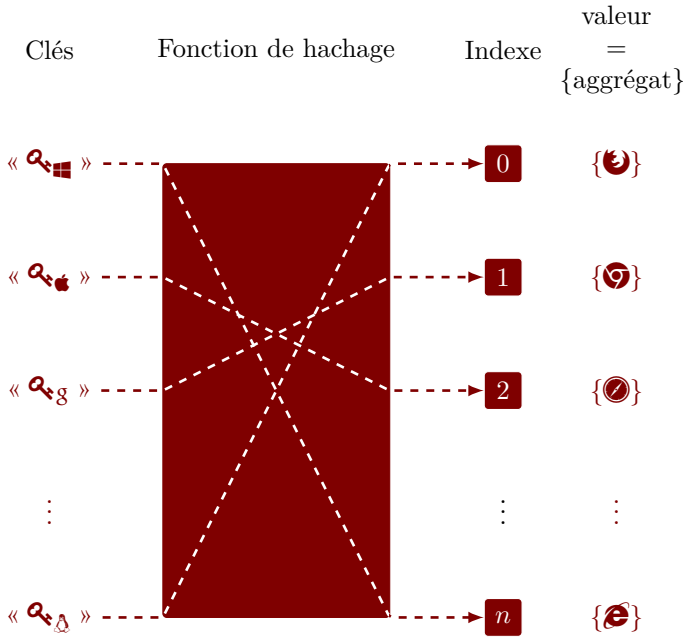


FIGURE 4.6 – Illustration du modèle de représentation des SGBDs NoSQLs orientés clés-valeur.

Les SGBDs NoSQL de type Clé-Valeur présentent un éventail de caractéristiques substantiellement profitable aux problématiques liées aux données d'un EDS :

En premier lieu, ces systèmes n'imposent aucune restriction quant au format de la valeur associée à chaque clé d'un couple $(clé, valeur)$. En interne, celle-ci est interprétée comme une simple donnée de type **B**inary **L**arge **OB**ject (BLOB) au sein duquel des données de toutes natures peuvent être stockées (e.g. données textuelles, données numériques, images, objets complexes, etc.). Les données cliniques étant notoirement très hétérogènes de ce point de vue (e.g. textes cliniques, résultats d'analyses biologiques, imageries médicales, séjours, etc.) les SGBDs

Clé-Valeur apportent l'assurance de pouvoir techniquement, aisément et avec flexibilité, assurer le stockage de ces données.

Il en est par ailleurs, de même, en ce qui concerne leur hétérogénéité structurelle. La définition d'un schéma de données n'étant pas davantage requis, la valeur n'est structurellement aucunement contrainte. L'intégration de nouvelles informations cliniques non gérées auparavant (quelque soit la structure des objets informatiques qui les représentent) peut ainsi être effectuée.

De plus, les EDSs sont fortement sujets à une constante alimentation en informations compte tenu que celles-ci sont essentiellement produites lors des prises en charge de patients. L'**A**pplication **P**rogramming **I**nterface (API) [7] simpliste et flexible que propose ce type de SGBD permet une maintenance facile et efficiente.

Enfin, les SGBDs Clé-Valeur sont sans doute les SGBDs NoSQL les plus évolutifs et assurant les meilleurs performances. La légèreté de la structure de données à base de table de hachage et du modèle Clé-Valeur permet une scalabilité horizontale et assure des accès unitaires aux valeurs extrêmement rapide. Celles-ci permettent également une faible consommation de RAM. La volumétrie conséquente des EDSs étant l'obstacle principal à l'accès aux données qu'ils contiennent et plus généralement à la réalisation d'une RI, ces systèmes présentent en conséquence un atout considérable dans le cadre de cette problématique de thèse.

L'un des inconvénients majeurs de ces systèmes est néanmoins son incapacité à fournir nativement des méthodes permettant de rechercher des couples $(clé, valeur)$ en fonction du contenu de la valeur qui est « opaque » vis à vis du système. Ces solutions imposent donc le développement d'outils annexes (e.g. index inversés) permettant la réalisation des fonctionnalités de RI. De plus, même si la modélisation des valeurs n'est a priori pas nécessaire, il est dans la pratique indispensable de définir une représentation ou une séparation cohérente de l'information en terme de couple $(clé, valeur)$ qui rend possible l'implémentation de ces fonctionnalités. Compte tenu de la structure non seulement complexe mais aussi variable des informations cliniques, la modélisation de celle-ci à l'aide du modèle simpliste Clé-Valeur n'est pas triviale.

4.2.2.2.2 (🖿) SGBD NoSQL orientés Document

Le modèle de représentation des données des SGBDs orientés document est très proche de celui des SGBDs orientés clés-valeur. Ces systèmes stockent également des couples $(clé, valeur)$ et la valeur est également un agrégat accessible par l'intermédiaire de la clé (Figure 4.7). La différence majeure de ces SGBDs avec le SGBDs NoSQLs orienté clé-valeur réside néanmoins dans le fait que la structure de l'agrégat est visible par les SGBDs NoSQLs orienté document. Ces SGBDs fournissent donc une API permettant de référencer les différents champs de l'agrégat au sein des requêtes et/ou de renvoyer des sous-ensembles constituant ce dernier. Dans la plupart des cas, les agrégats sont définis à l'aide des formats de données analysables par le SGBD tels que le format JSON ou XML.

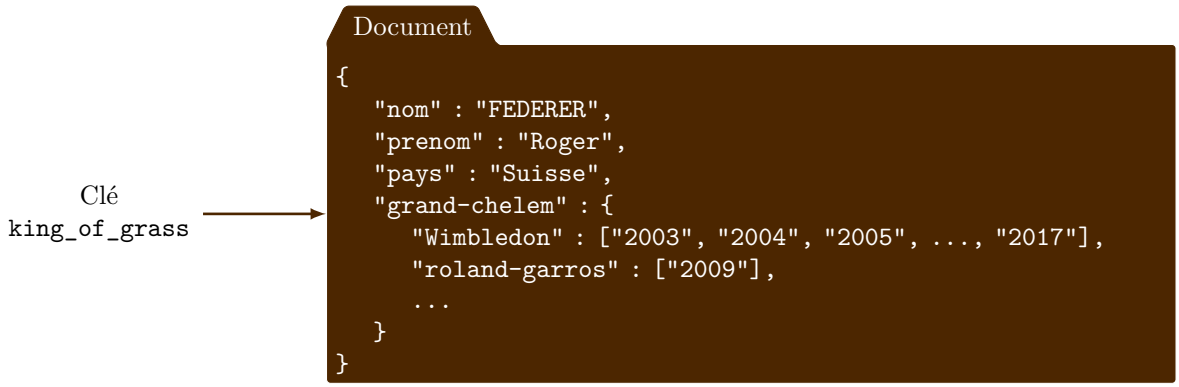


FIGURE 4.7 – Illustration du modèle de représentation des SGBDs NoSQLs orientés document.

7. : « **I**nterface de **P**rogrammation **A**pplicative »

Les SGBDs NoSQL orientés Document possèdent l'avantage de pouvoir structurer pleinement les informations au sein d'un document décrit hiérarchiquement à l'aide de propriétés éventuellement imbriquées. Dans le contexte d'un EDS cela permet notamment de faire face à l'hétérogénéité des données cliniques. Ces SGBDs permettent de plus l'exécution de requêtes expressives. Toutes les propriétés peuvent être recherchées. En somme, d'un point de vue purement fonctionnel, les SGBDs NoSQL orientés Document offrent de bonnes perspectives de RI. Les performances de ces systèmes, notamment en ce qui concerne celles des requêtes, sont néanmoins moins performantes que les SGBDs orientés Clé-Valeur ou Colonne. De plus, les documents de ces systèmes constituent des unités indépendantes. Aucun mappage relationnel n'est nativement possible entre les documents et/ou les propriétés. Ainsi, la complexité structurelle potentielle des documents s'avère également être un inconvénient majeur dans le cadre de la maintenance de données interconnectées telles que les données cliniques. Bien que cette caractéristique soit commune aux SGBDs NoSQL orientés Clé-Valeur ou Colonne, ces derniers associent généralement des valeurs plus granulaires et atomiques aux différentes clés.

Les SGBDs NoSQL orientés Document sont plus aboutis en terme de requêtage et offrent davantage de fonctionnalités de RI. Ces dernières ne permettent cependant pas de s'accommoder entièrement de la complexité de la RI au sein de données cliniques. Ils reposent, en outre, sur un modèle plus lourd à implémenter et n'offrent pas de performances comparables au modèle plus léger Clé-Valeur.

4.2.2.2.3 (▥) SGBD NoSQL orientés Colonne

Parmi les solutions NoSQLs, ce sont celles qui se rapprochent le plus des SGBDRs. Le SGBDs orientés colonnes se distingue avant tout des SGBDs orienté données par la logique interne de sérialisation des données, l'unité de stockage des données étant la colonne et non plus le tuple (i.e. la ligne).

Exemple 5 :
Par exemple, un SGBD orienté clé-valeur sérialisera probablement les informations relatives aux joueurs de tennis Rafael NADAL et Roger FEDERER ainsi :

```
{1,Rafael,NADAL,Espagne}

{2,Roger,FEDERER,Suisse}

...
```

alors que la sérialisation de ces mêmes données pour un SGBD orienté colonnes sera probablement plus proche de :

```
clé                 :  {1,2,...}
Colonne « prénom »  :  {Rafael,Roger,...}
Colonne « Nom »     :  {NADAL,FEDERER,...}
Colonne « Pays »    :  {Espagne,Suisse,...}
```

Ce type de sérialisation améliore les performances dans les contextes où les écritures sont peu fréquentes et les lectures des valeurs de colonnes de plusieurs lignes fréquentes. Le modèle de représentation de l'information des SGBDs NoSQLs orientés colonnes peut être vu comme une map clé-valeur à deux niveaux. Le niveau 0 permettant d'accéder à l'agrégat à partir de la clé $row_{\mathbf{Q}}$ l'identifiant et l'agrégat lui même possédant une structure clé-valeur et permettant d'accéder aux différentes colonnes de la donnée par l'intermédiaire d'une clé $column_{\mathbf{Q}}$:

$$\overbrace{Map : \langle row_{\mathbf{Q}},\ \overbrace{Map : \langle column_{\mathbf{Q}},\ columnValue \rangle}^{\text{Niveau 1 : accès aux colonnes}} \rangle}^{\text{Niveau 0 : accès aux données}}$$

D'un point de vue moins générique, les SGBDs NoSQLs orientés colonnes organisent leurs données en ***colonnes*** éventuellement partitionnées en sous-ensembles de colonnes appelés ***super colonnes***. Ces super colonnes et colonnes sont elles mêmes regroupées au sein d'une ***famille de colonnes***. L'accès à une famille de colonnes tout comme l'accès à une colonne se fait par l'intermédiaire de clés. Un illustration de ce modèle est donnée Figure 4.8.

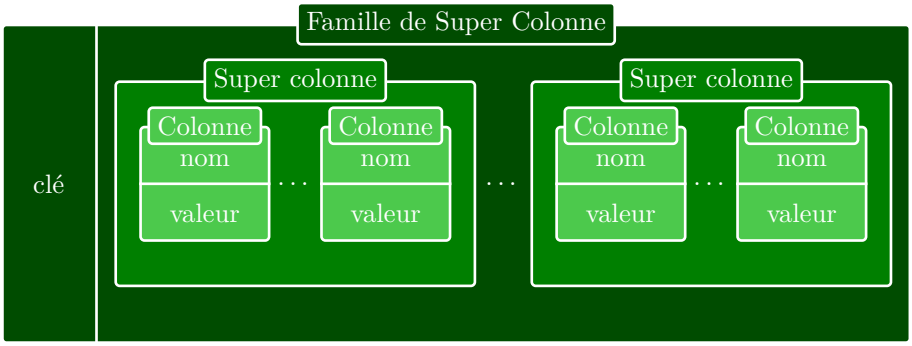


FIGURE 4.8 – Illustration du modèle de représentation des SGBDs NoSQLs orientés colonne.

Les SGBDs orientés colonnes imposent donc une modélisation basique des données en terme de famille de colonnes, super-colonnes et colonnes. Il permettent néanmoins un ajout très aisé de colonnes et apportent donc à la fois une flexibilité sur la structure des agrégats et un accès performant aux données. Comparé aux SGBDs orientés Clé-Valeur, les SGBDs orientés Colonne ajoutent un niveau de structuration en permettant la définition de famille de colonnes, super colonnes et colonnes. Les SGBDs ont l'avantage de permettre de requêter les données en fonction des valeurs de ces éléments et fournissent donc nativement des mécanismes utiles à la RI. En revanche, la valeur reste, in fine, stockée sous forme d'un BLOB opaque au système. Les SGBDs NoSQL orientés Colonne bénéficient donc globalement des mêmes qualités de scalabilité et de bonne mise à l'échelle que leurs homologues orientés Clé-Valeur. Ils offrent certaines fonctionnalités de recherche, cependant la conception des colonnes est un pré-requis critique à ces dernières.

Dans le cadre de la RI au sein d'un EDS, la structuration de base des données que proposent ces SGBDs semble insuffisante vis à vis de l'importante hétérogénéité des données que ces EDSs contiennent. Ainsi l'emploi d'un tel système impliquerait non seulement la résolution des mêmes problématiques que les SGBDs orientés Clé-Valeur mais imposerait également une approche de RI hybride. Plus précisément, l'information clinique ne pouvant pas être pleinement modélisée, ces fonctionnalités seraient alors assurées par les capacités de recherches natives du SGBDs orientés Colonne d'une part, et par l'utilisation de fonctionnalités annexes personnalisées destinées à la recherche des valeurs d'autre part.

4.2.2.2.4 (🔗) SGBD NoSQL orientés Graphe

Les SGBDs NoSQLs orientés Graphe ne reposent pas sur un modèle orienté agrégat. Une description de ce type de SGBD est néanmoins donnée ici par souci d'homogénéité. Ces solutions NoSQLs se concentrent sur la modélisation de la structure des données et sont particulièrement adaptées aux données sémantiquement complexes et fortement interconnectées. Le modèle de représentation exploité par ces SGBDs est inspiré de la théorie des graphes et bien que l'implémentation du concept de graphe varie parfois d'une solution à l'autre le modèle de représentation des SGBDs NoSQLs orienté Graphe peut être globalement assimilé à un ***multigraphe attribués, étiquetés, orientés***[8] à savoir que les données sont représentées à l'aide de :

nœuds (ou sommet) : chaque nœud représente une entité.

relations (ou arrêtes) : ces relations relient les nœuds entre eux. Elles sont typées à l'aide

8. 🇬🇧 : « Property Graph Model »

d'un libellé (i.e. étiquetées) et sont asymétriques[9] (i.e. orientées). Deux nœuds peuvent être connectés par plusieurs relations y compris si celles-ci possèdent la même étiquette (i.e. multigraphe).

propriétés (ou attributs) : les nœuds et arrêtes peuvent posséder un nombre variable d'attributs (i.e. graphe attribué) permettant de les décrire de manière plus détaillée. Ces attributs se présentent sous la forme de couples $(clé, valeur)$ ou la *clé* correspond au nom de la propriété et *valeur* à la valeur assignée à cette propriété.

Une illustration de ce type de graphe est donnée Figure 4.9 p. 105.

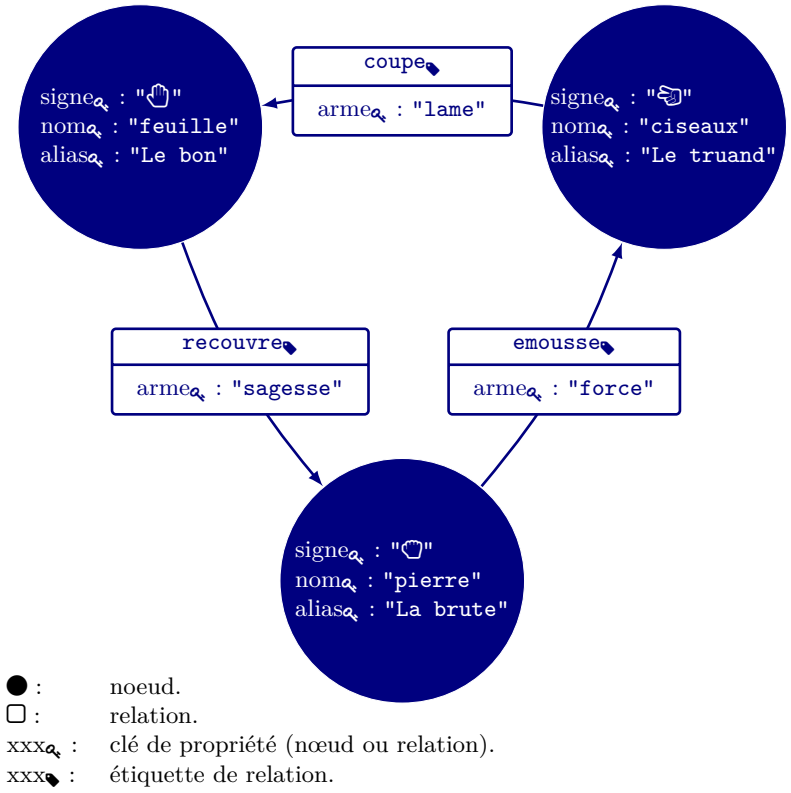


FIGURE 4.9 – Illustration du modèle de représentation des SGBDs NoSQLs orientés Graphe.

Les SGBDs orientés Graphe visent en premier lieu à permettre la représentation d'un réseau d'information organisé. Ils rendent possible la définition d'objets distincts et l'établissement de relations entre ces objets. Ils sont donc particulièrement adaptés à la représentation des informations cliniques qui sont, par nature, composées conceptuellement de multiples entités.

Ces solutions proposent, de plus, des APIs efficaces permettant de requêter exhaustivement les graphes de données en les parcourant avec précision. Ces APIs se matérialisent le plus souvent sous forme de véritables langages de requêtes spécifiques tels que Cypher[10] [142] ou encore Gremlin[11] [143]. Ces langages bénéficient, en outre, de fonctionnalités de recherche héritées de la théorie des graphes (e.g. chemin le plus court, degré de relation, etc.). Dans un contexte de RI au sein d'un EDS, ces dernières permettent non seulement de satisfaire aux besoins de base d'accès à l'information clinique mais ouvrent également des perspectives de RI plus étendues.

9. Le fait qu'un nœud A soit relié à un nœud B par une relation $\mathcal{R}$ quelconque (i.e. $A\mathcal{R}B$) n'implique pas nécessairement que B et relié à A par cette même relation. Par exemple la relation « est père de » est naturellement asymétrique alors que la relation « est de la même famille » est elle naturellement symétrique.

10. url : `https://neo4j.com/docs/cypher-manual/current/#cypher-intro`

11. url : `http://tinkerpop.apache.org/docs/3.3.3/upgrade/#_tinkerpop_3_3_3`

Aucun de ces langages n'est en revanche un standard à l'image du SQL en ce qui concerne la famille des SGBDRs.

Du point de vue des performances, certaines solutions proposent des mécanismes d'indexation des différentes propriétés des nœuds et des relations. Cependant, en raison de leur modèle non orienté-agrégat, les SGBDs orientés Graphe ne sont en réalité pas très adaptés aux architectures distribuées. Certaines solutions fournissent, malgré tout, cette possibilité par le biais de transaction. Cependant, certaines latences dans l'accès aux données en résultent.

Ces types de SGBDs présentent donc in-fine une scalabilité relative dans le sens où elle dépend de la capacité de la RAM à intégrer la totalité du graphe de données. Leurs utilisations dans un contexte de santé ou les volumétries de données sont considérables apparaît par conséquent peu prudent. Ces solutions restent néanmoins pertinentes dans la cadre de la problématique de cette thèse. De plus, de nouveaux triplestores basés sur des technologies NoSQL et permettant l'utilisation du langage SPARQL commencent à voir le jour. On peut notamment citer le triplestore CumulusRDF basé sur architecture cloud mais qui utilise néanmoins Apache Cassandra comme système de stockage des triplets **RDF** en arrière plan.

4.2.2.3 🔒 ⇒ 🔓 Latences d'accès vs. Contraintes d'intégrité relâchées

Les mécanismes d'intégrité référentielle et la représentation relationnelle des données des SGBDRs offrent des possibilités d'interrogation et de modification cohérente, rigoureuse et puissante des données. L'aspect transactionnel des SGBDRs quant à lui, par le biais de mécanisme assurant un respect strict des propriétés ACID, confère une extrême robustesse à ces systèmes ainsi qu'une sécurité et une cohérence infaillible des données à tout instant. Tous ces mécanismes ont néanmoins un coût considérable en terme de performances d'accès aux données et impliquent une dégradation des performances avec la montée en charge des données (y compris au sein d'une architecture non-distribuée).

Dans le cadre d'une architecture distribuée, les propriétés ACIDs ne peuvent être satisfaites sans la mise en place de synchronisations entre les différents nœuds du système afin de garantir les propriétés d'Atomicité, de Cohérence et d'Isolation. Dans le cas d'une distribution des données, des mécanismes supplémentaires de réplication et de synchronisation des mises à jour des données sont nécessaires afin d'assurer la propriété de durabilité. Ces synchronisations génèrent d'importantes latences et entrent en conflit avec le principe même de distribution des données et des traitements.

Ces incompatibilités inhérentes aux architectures distribuées, ont été définies de manière formelle par Eric Brewer [12] en 2000 lors d'un Symposium sur les principes d'informatique distribués sous la forme d'une conjecture nommée aujourd'hui « théorème CAP » ou « théorème de Brewer ». Une preuve formelle de cette conjecture a été apportée en 2002 par deux chercheurs du Massachusetts Institute of Technology (MIT) : Seth GILBERT et Nancy LYNCH [144].

12. Chercheur en Informatique de l'université de Berkeley, Californie, États-Unis

Propriété 1 (théorème CAP) :
Un système informatique distribué ne peut garantir de manière synchrone que deux des contraintes suivantes :

« *C* » **Consistency (Cohérence) :** *Elle correspond à la propriété de* ***cohérence séquentielle*** *qui se définit formellement comme l'existence d'un ordre total sur tous les accès à la mémoire distribuée qui préserve l'ordre des opérations. Dans la pratique, cette propriété se traduit par une exécution de chaque opération sur le système informatique distribué comme si elles étaient atomiques, instantanées et qu'elle s'exécutaient sur un nœud unique. En d'autres termes, cette propriété permet de garantir la mise à jour de l'état de toutes les copies d'une même donnée sur l'ensemble des nœuds du système de sorte qu'à chaque instant, tous les nœuds retournent une même valeur pour une donnée particulière qui correspond à l'état valide le plus récent de cette donnée. En conclusion tous les nœuds du système « voient les mêmes données au même moment ».*

« *A* » **Availability (Disponibilité) :** *Cette propriété impose que toutes opérations (lecture ou écriture) reçoivent une réponse.*

« *P* » **Partition tolerance (Tolérance au partitionnement) :** *Aucune panne moins importante qu'une coupure totale de réseau ne doit empêcher le système de répondre correctement à toutes les opérations de lecture et/ou d'écriture. En cas de morcellement en sous-réseaux du système distribué, chacune des partitions doit pouvoir fonctionner de manière autonome. Le système continue de répondre correctement indépendamment de la perte au sein du réseau d'un nombre arbitraire de messages entre les différents nœuds.*

Remarque 2 :
Le théorème CAP n'affirme pas que les propriétés C, A et P ne peuvent être toutes trois vérifiées à un instant donné mais qu'elles ne peuvent être garanties à tout instant. En effet, aucun système distribué n'est à l'abri des coupures réseaux entre les différents nœuds et l'éventualité d'un partitionnement de ce dernier doit donc être pris en compte.
Dans le cadre d'un partitionnement avéré, il est alors nécessaire de faire un choix entre assurer la cohérence des données ou leur disponibilité :

- *maintenir la cohérence des données implique de renoncer à l'exploitation de données potentiellement non actualisées suite à la coupure réseau et au partitionnement du système.*
- *inversement, maintenir la disponibilité des données nécessite l'exploitation de ces dernière dont l'actualisation n'est pas assurée afin de pouvoir répondre à toutes les requêtes.*

Ainsi un SGBD ne peut garantir ***à tout instant*** qu'uniquement les couples de propriété :

- CA (Consistency + Availability) ;
- AP (Availability + Partition tolerance) ;
- CP (Consistency + Partition tolerance).

En particulier, les bases de données relationnelles sont CA. Quant aux SGBDs NoSQLs non-relationnels, ils sont AP pour une majorité d'entre eux (afin de garantir de meilleures performances) ou CP.

Certains SGBDs ont la possibilité de modifier la politique de gestion de la concurrence et peuvent être configurés afin d'être soit AP soit CP (e.g. MongoDB, CouchBase, Cassandra, etc.).

L'ensemble des SGBDs NoSQLs peuvent donc être classés selon la politique de gestion de la concurrence qu'ils proposent. Cette classification est souvent illustrée à l'aide d'un triangle dont les sommets correspondent aux différentes propriétés C, A et P (Figure 4.10).

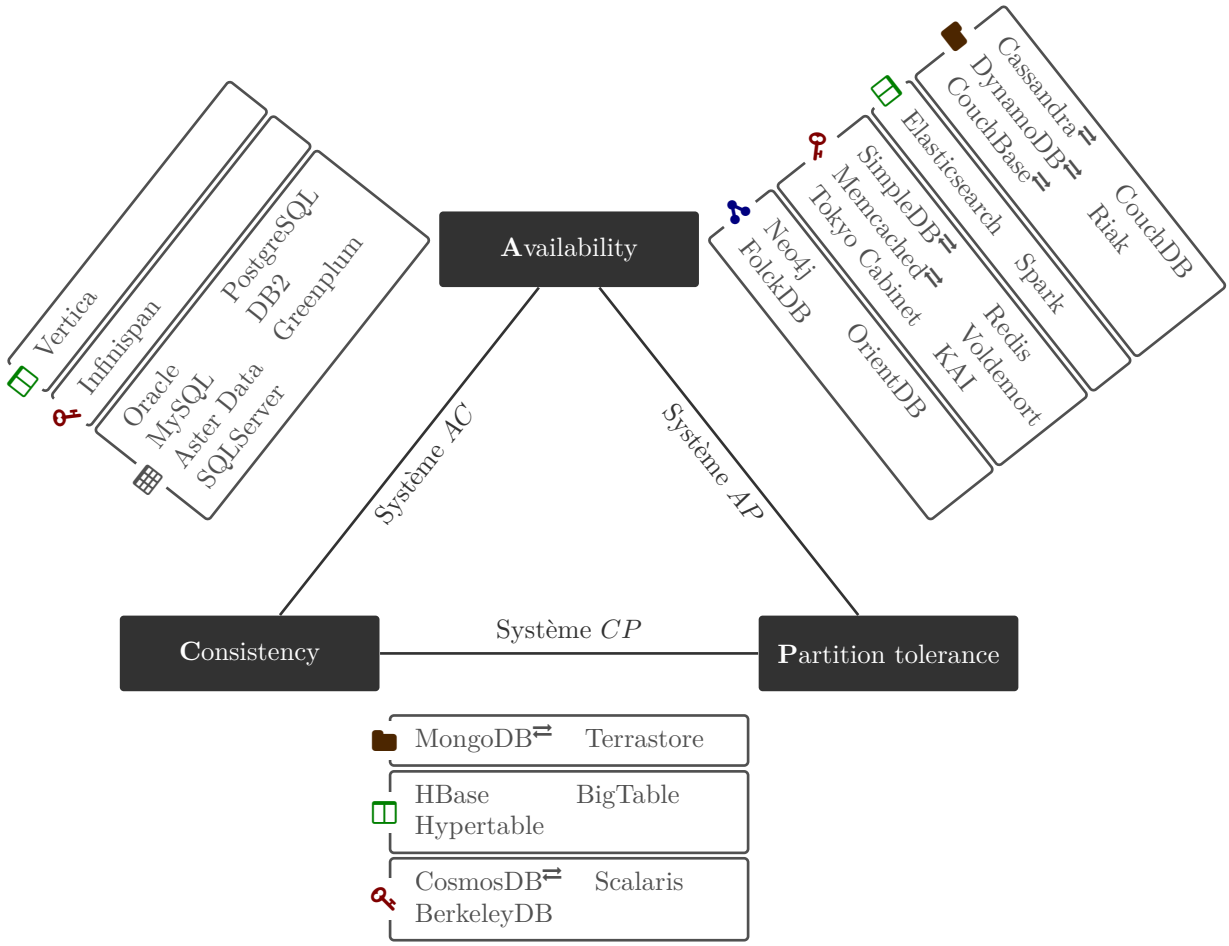


FIGURE 4.10 – Triangle CAP illustrant de manière non exhaustive la politique de gestion de la concurrence de plusieurs SGBDs.

En conclusion, les SGBDs destinés à être utilisés au sein d'une architecture distribuée privilégient fatalement davantage la haute disponibilité des données, et la rapidité d'accès à ces dernières au détriment de la cohérence et de l'exactitude des réponses. Les bases de données NoSQLs opèrent un relâchement des propriétés ACIDs notamment des contraintes impossibles à maintenir dans un contexte distribué en allégeant les synchronisations liées aux accès concurrents aux données. Les propriétés ACIDs n'étant alors plus adaptées à ces caractéristiques, les ***propriétés BASE*** pour **B**asically **A**vailable, **S**oft–state, **E**ventually Consistent (BASE) ont été proposées afin de caractériser les SGBDs NoSQLs :

Définition 12 (propriétés BASE) :

Basically Available : *Le système doit toujours être accessible et un taux de disponibilité des données doit être garanti quelle que soit la charge de la base de données que ce soit en terme de données ou de requêtes. Compte-tenu du théorème CAP, toute requête obtiendra effectivement une réponse mais l'intégrité des données reçues n'est pas garantie.*

Soft-state : *L'état du système peut changer au cours du temps notamment pendant les mises à jour ou lors d'ajout/suppression de nœuds.*

Eventually consistent : *La cohérence des données n'est pas une priorité à tout instant (verrouillage optimiste). A terme, le système atteindra un état cohérent.*

4.2.3 Synthèse

Dans cette section, les différents types de SGBDs alternatifs existant ont été décrits. Il existe, dans la pratique, de nombreux systèmes qui diffèrent :

— au niveau des structures de données utilisées en interne ;

— au niveau des capacités de modélisation des informations ;

— au niveau des fonctionnalités d'accès proposées ;

— au niveau des politiques de gestion des accès concurrents et des transactions ;

— au niveau de la scalabilité horizontale ;

— au niveau des capacités de persistance des données.

Relativement au nombre important de ces systèmes, peu d'entre eux ont en réalité été envisagés dans le cadre de mon travail de thèse. Leur grande hétérogénéité rend, de plus, coûteuse en temps et spécifique leur expérimentation et leur intégration dans le cadre d'une problématique telle que celle de l'accès à des données cliniques. Les SGBDs NoSQL ont néanmoins l'avantage de garantir une amélioration des performances d'accès aux données par l'intermédiaire d'un stockage de ces dernières en RAM. Cette amélioration se fait néanmoins au détriment de la garantie d'intégrité des données qu'offrent les SGBDRs. Les données cliniques sont pourtant des données à la fois privées et sensibles. Dans le cadre de mon travail de thèse, et plus généralement au sein du SI du D2IM, l'**I**n **M**emory **D**ata **G**rid (IMDG) [13] Infinispan (Infinispan) [14] a été utilisé comme support de stockage des données. Comme je le décris plus amplement dans le chapitre suivant, cette solution NoSQL orientée Clé-Valeur permet de garantir la cohérence et la disponibilité des données (i.e. système *AC*) au même titre que les SGBDRs. En revanche, les systèmes Clé-Valeur ne permettent pas de définir nativement un modèle de données de manière analogue aux SGBDRs. Dans le chapitre suivant, je décris le modèle de donnée générique utilisé par le D2IM ainsi que sa transposition au sein de l'IMDG Infinispan garantissant, ainsi, des fonctionnalités génériques d'accès aux données.

13. : « Grille de Données en Mémoire »
14. url : `http://infinispan.org/`

Chapitre 5

Modélisation et intégration des données de l'EDSS

Sommaire

5.1 À l'origine, un SGBDR 113
5.1.1 Notre modèle générique 113
5.1.2 Différents corpus 117
5.1.3 La course « à la perf' » 118
5.2 Le virage du NoSQL 120
5.2.1 L'In Memory Data Grid qualifié 120
5.2.2 Le changement de paradigme en action 120
5.2.3 La recherche d'information 123

Du fait de la sensibilité des données qu'ils exploitent, les SRIs basés sur les EDSs se doivent de persister leurs données de manière sécurisée et d'en permettre un accès sûr et cohérent. La RI au sein de ces données de structure complexe nécessite en outre des méthodes étendues de requêtage.

Les SGBDRs présentent l'avantage, du fait de leur ancienneté et du modèle relationnel qu'ils implémentent, d'être particulièrement aboutis en terme de requêtage (notamment grâce au langage de requête SQL qu'ils proposent), d'intégrité des données (i.e. respect strict des propriétés ACIDs) et de sécurisation de ces dernières. C'est donc, en premier lieu, sur un SGBDR que s'est porté notre choix pour la modélisation et l'intégration des données de l'EDS du CHU de Rouen.

Ces dernières années, de nombreuses solutions NoSQLs ont vu le jour avec, pour principaux objectifs, de pallier aux limitations des SGBDRs en diminuant les temps d'accès aux données (cf. section 4.2). Les performances étant un critère important dans le cadre de la RI au sein des EDSs, je me suis interessé à la question de la modélisation et de l'intégration des données de santé au sein d'une base de données NoSQL.

Comme précisé en introduction, dans ce travail de thèse, deux preuves de concepts de moteur de recherche ont été réalisées. Ces deux derniers ont pour objectif majeur la RI au sein de l'EDS du CHU de Rouen. Ils se distinguent avant tout par les bases de données sous-jacentes qu'ils exploitent et à partir desquelles ils accèdent aux différentes données de santé :

Le SSE$_{SQL}$: Conçu pour accéder aux données directement à partir d'une base de données relationnelle munie d'un modèle de données générique de type EAV.

Le SSE$_{NoSQL}$: Conçu pour accéder aux données via un cache en mémoire de type IMDG implémenté à l'aide de la solution NoSQL Infinispan. Un SGBDR PostgreSQL est néanmoins toujours utilisé pour assurer la persistance des données de façon sûre.

La caractéristique première de l'EDS du CHU de Rouen est son aspect sémantique. Ce dernier fournit une description sémantique de l'information de santé à laquelle le SSE$_{SQL}$ et le

SSE_{NoSQL} permettent d'accéder. Si les SGBDs sur lesquels ces moteurs de recherche diffèrent technologiquement, ils adoptent néanmoins un même modèle de données hérité du paradigme de représentation de l'information du Web Sémantique. Ce dernier a été développé depuis plusieurs années dans le cadre de travaux de thèse [30].

Ce chapitre s'attache à décrire ce paradigme de représentation de l'information ainsi que les modèles de données qui en résultent. Les différents corpus de données utilisés durant ce travail de thèse sont également décrits.

5.1 À l'origine, un SGBDR

5.1.1 Notre modèle générique

Le modèle de données de base du SI de l'équipe du D2IM est un modèle **générique** de type **E**ntity–**A**ttribute–**V**alue (EAV). Ce modèle résulte notamment de travaux antérieurs [30, 145, 146]. Il peut être qualifié de **générique** dans le sens où il permet une **description générique de l'information**. Bien que n'ayant pas été développé dans le cadre de cette thèse, ce modèle revêt néanmoins une importance capitale dans le contexte de cette dernière. L'idée de base des moteurs de recherche est de « transférer », au niveau de la RI, cette généricité de description de l'information afin de la rendre utilisable et accessible de manière tout aussi générique.

Les modèles de type EAV sont largement exploités notamment pour stocker des données biomédicales. Ils permettent de palier à certains manques de flexibilité du modèle relationnel. Une brève explication de l'idée sous-jacente de ce type de modèle est donnée remarque 3.

> ⚠ **Remarque 3 (Les modèle de type EAV) :**
> *Certaines données complexes et plus particulièrement les données cliniques présentent un nombre conséquent de paramètres et attributs descriptifs. Par exemple, la description d'un séjour d'un patient dans un établissement hospitalier requiert un nombre important d'attributs descriptifs tels que : les dates d'entrée et de sortie du séjour, les modes d'entrée et de sortie, le poids du patient lors du séjour, le statut du séjour, etc. Lorsqu'il s'agit de structurer ces données au sein d'un modèle relationnel classique, cette multitude d'attributs se traduit par des tables pourvues d'un nombre de excessif colonnes. Chaque ligne de la table désignant ainsi une entité quelconque (e.g. un séjour) et chaque valeur de colonne de cette ligne correspondant à un paramètre descriptif de ce séjour (e.g. statut du séjour, date d'entrée, etc.). Certains paramètres descriptifs sont parfois non-applicables à certaines entités (e.g. le poids du patient n'est pas systématiquement renseigné car non pertinent pour certains séjours). Ceci se traduit par un nombre parfois conséquent d'assignations à une valeur nulle de certaines colonnes. Les modèles de type EAV ont pour objectif principal de pallier à ces inconvénients. Au sein d'un tel modèle, les entités ne sont plus décrites à l'aide d'une seule et même ligne composée de plusieurs valeurs de colonnes mais à l'aide d'une table de **faits** contenant une ligne par paramètre descriptif. Chaque fait étant ainsi décrit à l'aide de trois paramètres : l'identifiant de l'entité concernée, le type d'attribut (e.g. date d'entrée, etc.) et la valeur de ce paramètre.*

Lorsque l'on souhaite stocker et gérer des données au sein d'un SGBDR, on distingue classiquement trois modèles de représentation des données :

Le Modèle Conceptuel de Données (MCD) qui donne une vision facilement compréhensible des données y compris pour une personne ne possédant pas de compétences techniques. Dans un MCD, les données et leur organisation sont décrites à l'aide d'un formalisme entité-association. Ils définissent des entités conceptuelles et des relations permettant de décrire les différentes associations entre ces entités. Ils donnent ainsi une vision à la fois intuitive et en adéquation avec la réalité de la situation à laquelle ces données se rapportent. Le MCD définit en réalité une représentation d'un ensemble d'informations sous forme d'un **graphe de données** dont les nœuds correspondent aux entités de ce MCD et où les arcs correspondent aux relations qu'il établit entre les entités.

Le Modèle Logique de Données (MLD) qui lui reprend la description abstraite et intuitive des données caractérisée dans le MCD afin d'en définir une structuration relationnelle. En d'autres termes, il permet décrire le MCD sous forme de tables, de colonnes, de tuples, de clés primaires et étrangères. Il donne donc une structuration relationnelle des données en vue de leur maintenance dans un SGBDR.

Le Modèle Physique de Données (MPD) qui, lui, se distingue du MLD dans le sens où il correspond à une implémentation du MLD au sein d'un SGBDR particulier. Il est struc-

turellement identique mais apporte des informations d'implémentation propres au langage de programmation fournit par ce SGBDR particulier. Il précise notamment le typage de chaque colonnes.

Dans la pratique, le MLD et le MPD peuvent être assimilés de telle sorte que l'on considère, d'une part la **vision conceptuelle que l'on a des données (fournie par le MCD)**, et d'autre part la **vision structurelle de celles-ci utilisée pour les stocker (fournie par MLD et/ou le MPD)**. Bien que ces deux visions peuvent différer dans une modélisation relationnelle classique, la structure du MLD hérite néanmoins souvent de celle du MCD. Par exemple, une table du MLD correspond généralement à une entité ou à une relation du MCD dans la pratique.

Dans une modélisation relationnelle des données, un changement du MCD (i.e. lorsque de nouveaux types d'informations doivent être gérés) impose généralement un changement structurel du MLD (e.g. ajout de tables, de colonnes, etc.). Le modèle de données du SI du D2IM exploité dans le cadre de ce travail, et plus généralement des modèles de données de type EAV, n'imposent en revanche pas de telles modifications structurelles du modèle de données. Ce dernier s'attache, en effet, à définir un ensemble de tables permettant de modéliser génériquement le formalisme entité-association lui même. Il ne modélise donc pas un MCD particulier, qui constitue lui une instance de ce formalisme, et ne modélise pas des informations spécifiques issues d'une situation réelle et concrète (cf. Figure 5.1) même si il permet des les intégrer.

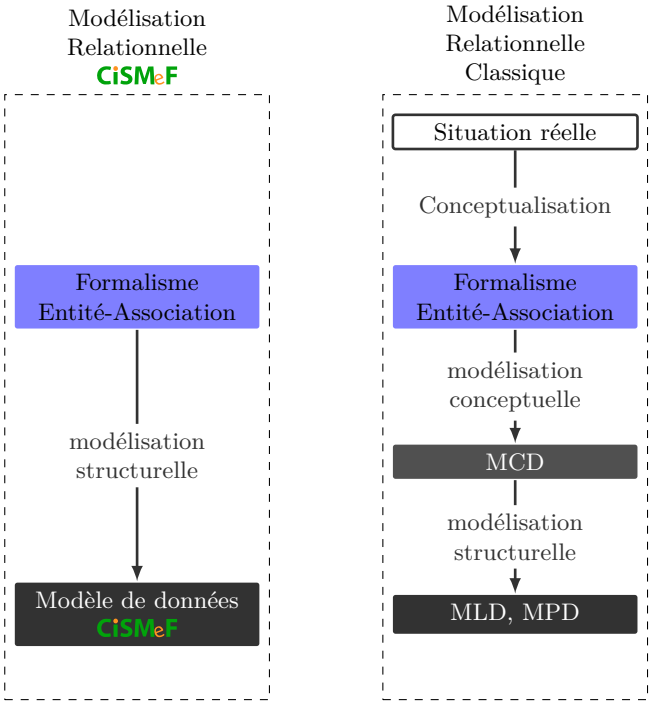


FIGURE 5.1 – Modélisation relationnelle du D2IM et modélisation relationnelle classique.

Le modèle de données du D2IM est indépendant de la donnée elle même. Il ne requiert pas de modifications structurelles même lorsque de nouvelles données doivent être gérées, à partir du moment où leurs représentations à l'aide du formalisme entité-association est possible. De même, les données insérées dans ce modèle définissent intrinsèquement une information sous forme d'un formalisme entité-association et donc un MCD.

Le MLD du D2IM est donné Figure 5.2. Bien qu'il ne s'agisse pas, à proprement parler, d'un MPD, des informations de typage des colonnes sont néanmoins données de manière générique. Ce modèle est composé de 9 tables. Il permet le stockage de n'importe quel type d'information

sans modification structurelle du modèle (i.e. sans ajout de tables, colonnes, etc.). Comme précisé précédemment, ce dernier s'appuie sur une logique de structuration des données basée sur le formalisme entité-association. Plus précisément, il est inspiré du paradigme de représentation de l'information du Web sémantique dans lequel l'information est décrite d'une part, par l'intermédiaire d'une description **RDF** des données et d'autre part, à l'aide d'une Ontologie RDFS et/ou OWL dont le rôle est de structurer le vocabulaire exploité par cette description **RDF**. Á l'image de ce partage, le modèle de données ici décrit se partitionne en deux sous-ensembles de tables :

— Les tables de données.

— Les tables de modèles.

Ces deux ensembles sont respectivement assimilables à la partie Donnée et à la partie Ontologique du Web Sémantique. Ils sont décris dans les sections suivantes. La partie donnée du modèle de données peut également être vue comme le pendant de l'$\mathcal{A}Box$ de la notion de bases de connaissances issues de la théorie des Logiques de Descriptions [147, 148] tandis que la partie modèle permet d'en définir l'ensemble de axiomes Terminologiques ($\mathcal{T}Box$) [146].

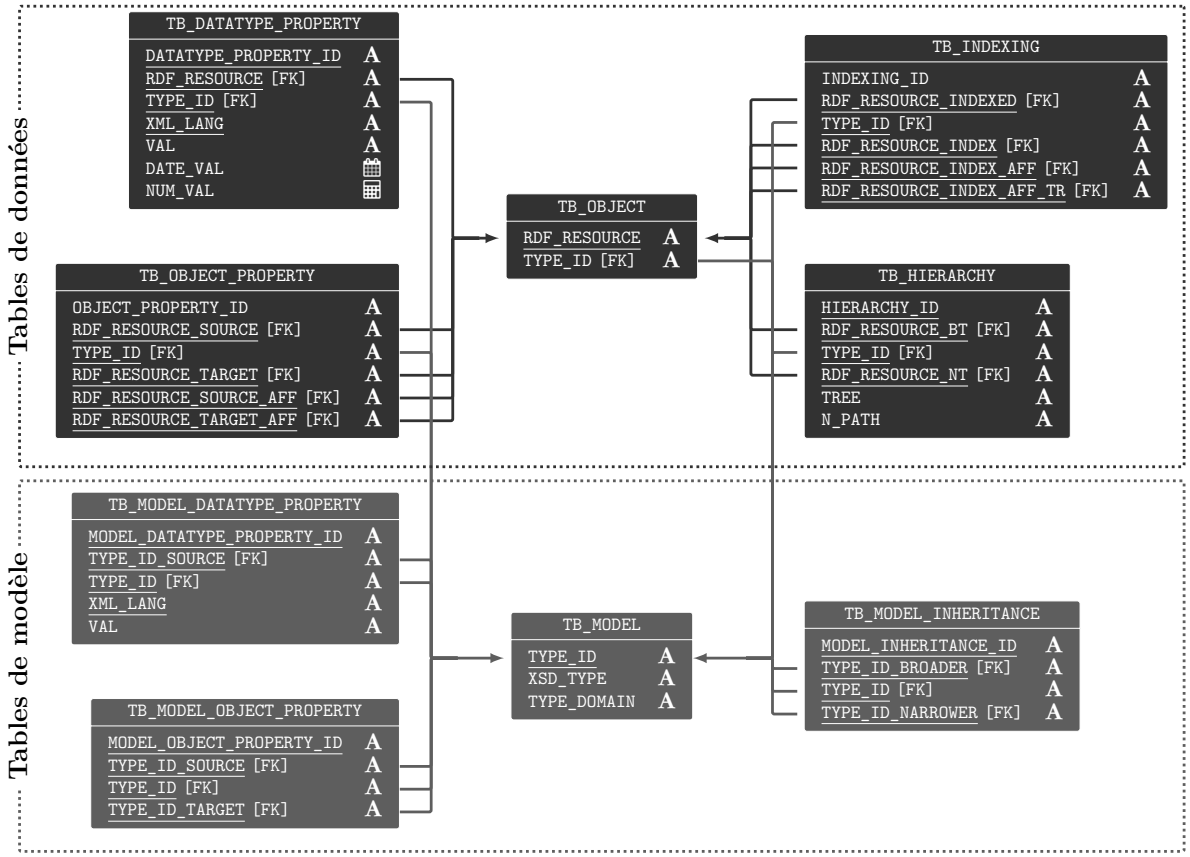


FIGURE 5.2 – Modèle logique de donnée du SI. Les clés primaires sont soulignées (e.g. CLE_PRIMAIRE), les clés étrangères sont identifiées par l'annotation [FK], les champs textuels par **A**, les champs numériques par et les champs de type date par .

5.1.1.1 Les tables de données

Les tables dites « de données » sont les tables :

— TB_OBJECT.

— TB_DATATYPE_PROPERTY.

— TB_OBJECT_PROPERTY.

— TB_INDEXING.

— TB_HIERARCHY.

Ces tables permettent la structuration et le stockage des données concrètes destinées le plus souvent à l'affichage ou à la mise à disposition des utilisateurs par les applications. La structure de ces tables reprend le paradigme de représentation de l'information en triplets (*sujet*, *predicat*, *objet*) du modèle **RDF** tout en faisant physiquement la distinction entre les triplets permettant d'associer une valeur littérale à une entité (i.e. (*objet*, *attribut*, *valeur*)) de celles permettant de relier d'autres entités à cette entité (i.e. (*objet*, *relation*, *objet*)). Ces deux types de propriétés sont respectivement nommées les ***datatype properties*** et les ***object properties*** par le W3C dans le cadre du Web sémantique.

Dans la pratique, la table `TB_OBJECT` permet de « déclarer » une entité (ou un objet) et la table `TB_DATATYPE_PROPERTY` permet d'en définir les attributs. Celle-ci permet en réalité de définir des triplets de colonnes équivalents aux triplets **RDF** de type datatype property :

(`RDF_RESOURCE`, `TYPE_ID`, `VAL`)
(`RDF_RESOURCE`, `TYPE_ID`, `NUM_VAL`)
(`RDF_RESOURCE`, `TYPE_ID`, `DATE_VAL`)
$\Leftrightarrow$ Triplet **RDF** de type datatype property (*objet*, *attribut*, *valeur*)

Les tables `TB_INDEXING`, `TB_HIERARCHY` et `TB_OBJECT_PROPERTY` permettent toutes les trois de définir les relations relatives à cette entité. La table `TB_INDEXING` permet de stocker les relations d'indexation (e.g. indexation d'un compte-rendu avec un concept appartenant à un SOC quelconque), la table `TB_HIERARCHY` est principalement exploitée pour définir les relations hiérarchiques entre les concepts des différents SOCs (e.g. le concept « *Doctorat* » [003912 (TSP)] issu du **T**hesaurus **S**anté **P**ublique (TSP) édité par la BDSP est un « fils » du concept « *Diplôme* » [003832 (TSP)]). Enfin, la table `TB_OBJECT_PROPERTY` permet de définir toutes les autres relations incluant notamment les alignements entre concepts (e.g. « *Doctorat* » [003912 (TSP)] possède un alignement exact avec le concept « *Thèses* » [*M*0359797 (MeSH)]) mais aussi toutes autres sortes de relations structurelles (e.g. relation entre une analyse biologique et un patient, relation entre une ressource bibliographique et un éditeur, etc.). De même, ces trois tables permettent de définir des triplets de type object property par l'intermédiaire de triplets de colonnes :

(`RDF_RESOURCE_INDEXED`, `TYPE_ID`, `RDF_RESOURCE_INDEX`)
(`RDF_RESOURCE_BROADER`, `TYPE_ID`, `RDF_RESOURCE_NAROWER`)
(`RDF_RESOURCE_SOURCE`, `TYPE_ID`, `RDF_RESOURCE_TARGET`)
$\Leftrightarrow$ Triplet **RDF** de type object property (*objet*, *relation*, *objet*)

> ⚠ **Remarque 4 (Notion d'indexation) :**
> *Dans le cadre du SI du* **D2IM**, *la notion d'indexation peut néanmoins prendre une forme légèrement plus complexe qu'une simple association entre une ressource (i.e. colonne `RDF_RESOURCE_INDEXED`) et un concept terminologique (i.e. unique colonne `RDF_RESOURCE_INDEX`). Cette vision de l'indexation, issue de celle adoptée par la terminologie MeSH, effectue premièrement une distinction entre les indexations majeures et mineures. Une indexation majeure est une indexation pour laquelle le concept indexant correspond au sujet principal de la ressource indexée. Dans le cas contraire l'indexation est mineure. Une indexation peut également être affiliée à deux concepts en plus du concept indexant (i.e. indexation post-coordonnée) :*
>
> — *un concept dit* ***qualificatif*** *permettant de spécifier le sens ou un aspect particulier que doit prendre le concept indexant (i.e. colonne `RDF_RESOURCE_INDEX_AFF`) ;*
>
> — *un concept indiquant le* ***type de ressources*** *de la ressource indexée (i.e. correspondant à la colonne `RDF_RESOURCE_INDEX_AFF`).*

5.1.1.2 Les tables de modèle

Les tables de modèle sont les tables :

— `TB_MODEL` ;

— `TB_MODEL_DATATYPE_PROPERTY` ;

— `TB_MODEL_OBJECT_PROPERTY` ;

— `TB_MODEL_INHERITANCE`.

Elles permettent de modéliser la sémantique des objets et relations stockées par les tables de données de manière analogue à la partie ontologique du Web sémantique. Elles permetent donc de définir la sémantique du vocabulaire exploité par les descriptions **RDF** définies dans la partie donnée. Les données stockées constituent ainsi les méta-données. La table `TB_OBJECT` permet de définir un type d'objet, d'attribut ou de relation, `TB_MODEL_DATATYPE_PROPERTY` permet de décrire ces types à l'aide d'attributs (e.g. libellés de relation ou d'attribut) et la table `TB_MODEL_OBJECT_PROPERTY` permet de définir des relations entre les types d'objets, de relations et d'attributs (e.g. un objet de type **ressource bibliographique** possède un attribut de type **auteur**, un objet de type **analyse biologique** est rattaché à un objet de type **patient** par une relation r_1, la relation r_2 est symétrique, etc.).

5.1.2 Différents corpus

Deux corpus de données ont été constitués et intégrés au sein du modèle de données présenté précédemment (Figure 5.2). Ces derniers ont servi de base de réflexion, de modélisation et de développement au SSE_{SQL} initié avec le projet RAVEL [ANR–11–TECS–012]. La migration initiale des données ainsi que leurs mises à jour régulières ont été assurées par un script **E**xtract **T**ransform **L**oad (ETL) également développé dans le cadre de cette thèse. Ces deux corpus sont constitués de données de santé extraites de la base de données relationnelles CDP intégrée au SI du CHU de Rouen. Cette sous-section présente brièvement ces deux corpus de données.

5.1.2.1 Le corpus $_{2\ 000}$

Ce corpus de données est composé des données de santé d'environ 2 000 patients sélectionnés manuellement par un professionnel de santé. Afin de garantir une volumétrie suffisante de données cliniques, seuls des patients ayant au minimum effectué 20 séjours au sein du CHU de Rouen ont été sélectionnés (i.e. des patients susceptibles de présenter une histoire médicale potentiellement particulière). $_{2\ 000}$ constitue historiquement le corpus de base ayant motivé les différents travaux de l'équipe de recherche autour des données cliniques. La Figure 5.3 p. 117 donne un aperçu des types de données inclus dans le corpus $_{2\ 000}$ ainsi que des relations qu'ils entretiennent entre eux.

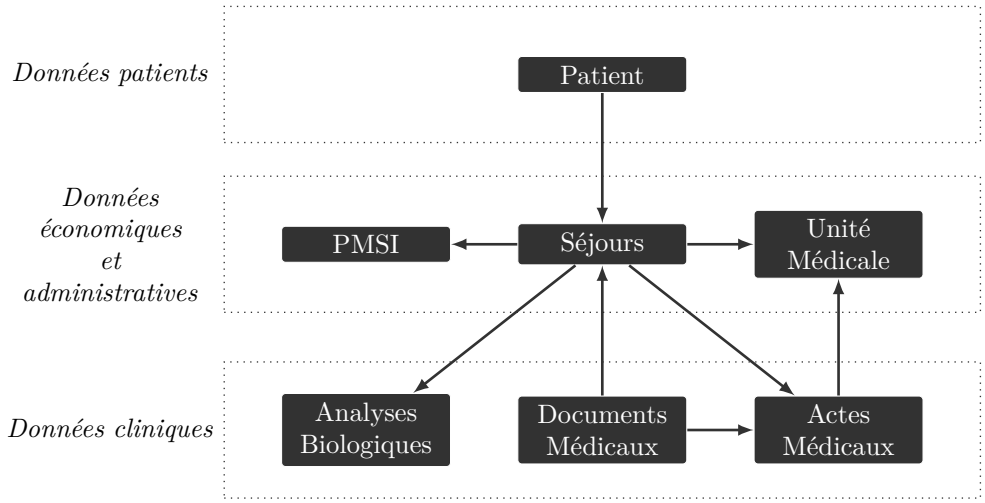


FIGURE 5.3 – Principales données patient inclus dans le $_{2\ 000}$.

On y retrouve notamment les trois niveaux d'information décrits précédemment : le niveau patient (e.g. date de naissance, sexe, etc.), le niveau de prise en charge économico-administratif (e.g. codage PMSI, entrée, sortie, type de séjour, etc.) et le niveau d'information clinique (e.g.

examens biologiques, actes médicaux et chirurgicaux avec leurs codages CCAM, comptes-rendus, etc.). Ce corpus a été anonymisé et validé par l'ASIP Santé et les comptes-rendus ont été indexés automatiquement avec l'ECMT. Il a également été utilisé dans le cadre du projet LERUDI.

5.1.2.2 Le corpus 👥$_{60\ 000}$

Le corpus 👥$_{60\ 000}$, composé d'approximativement 60 000 données patients (i.e. 30 fois plus que 👥$_{2\ 000}$), a été constitué de manière similaire au corpus 👥$_{2\ 000}$ dans un objectif de passage à l'échelle. Les patients inclus dans ce corpus sont les patients du service de dermatologie du CHU de Rouen. En terme de couverture fonctionnelle, le corpus 👥$_{60\ 000}$ couvre sensiblement le même périmètre que le corpus 👥$_{2\ 000}$ excepté l'ajout de certaines données propres à la dermatologie qui n'ont pas été exploitées dans le cadre de ce travail de thèse.

5.1.3 La course « à la perf' »

Le corpus 👥$_{2\ 000}$ a servi de base à la modélisation et au développement du SSE$_{SQL}$. Ce dernier génère des requêtes SQLs permettant de requêter les données patients stockées au sein d'une base de données ORACLE® munie du modèle de données décrit plus haut.

Bien qu'aucune étude quantitative approfondie n'ait été menée sur les performances du SSE$_{SQL}$ dans un contexte de RI au sein de données patients, les performances de ce système dans le cadre du corpus 👥$_{2\ 000}$ ont pu être considérées comme « acceptables ». La Table 5.1 donne un ensemble de cas d'usage ayant initialement servi de base à la modélisation et au développement du SSE$_{SQL}$ notamment dans le cadre du projet RAVEL [ANR–11–TECS–012].

Les temps d'exécution des requêtes ayant permis au SSE$_{SQL}$ de répondre ces cas d'usage étaient de l'ordre de la seconde pour une grande majorité de ces derniers et ne dépassaient pas les 10 secondes y compris pour les requêtes les plus complexes et les plus coûteuses en terme de requêtage SQL (i.e. requête impliquant des critères chronologiques (avant, après, premier, dernier, etc.), recherche plein-texte, explosion hiérarchique). Ces mêmes cas d'usage ont été testés sur le corpus 👥$_{60\ 000}$ afin d'évaluer les performances du SSE$_{SQL}$ avec la montée en charge des données. De nombreux efforts d'optimisation ont été réalisés, majoritairement destinés à l'optimisation des plans d'exécution (i.e. création d'index y compris d'index de domaine pour les recherches textuelles, utilisation des « hints » ORACLE® permettant d'orienter le plan d'exécution). Malgré ces efforts, l'interrogation du système sur le corpus 👥$_{60\ 000}$ a montré des performances peu encourageantes. La majorité des interrogations au moteur de recherche ont mené à des temps d'exécution dépassant l'ordre de la minute.

D'un point de vue technique, le modèle de données générique EAV exploité par le système implique une certaine complexité des requêtes SQLs en comparaison avec les modèles de données relationnelles classiques (ces derniers impliquant notamment l'exploitation d'un nombre important de jointures). Cependant, compte tenu des efforts d'optimisation réalisés, ce dernier ne peut constituer à lui seul la cause principale d'un tel écart de performance entre les deux corpus 👥$_{2\ 000}$ et 👥$_{60\ 000}$. Ce modèle de données étant de surcroît un élément essentiel du SI, l'utilisation d'un SGBDRs a été remise en cause au profit d'une technologie NoSQL.

Afficher les séjours, analyses biologiques, comptes-rendus, etc. d'un patient donné.
Afficher, pour un patient donné, les séjours ayant eu lieu dans une unité médicale donnée.
Afficher les séjours au sein desquels un patient a reçu un diagnostic donné (e.g. « *brûlures couvrant moins de 10% de la surface du corps* » [T310 (CIM–10)]).
Afficher les analyses biologiques d'un type donné (e.g. Calcium) comprise entre x et y.
Afficher les patients présentant une analyse biologique supérieure à la normale (e.g. patient présentant une hypernatrémie).
Rechercher en « plein-texte » un mot ou une expression au sein des comptes-rendus de ces dernier.
Afficher le (resp. la) premier ou dernier (resp. première ou dernière) séjour (resp. analyse biologique) d'un patient donné.
Afficher pour un patient donné l'analyse biologique ayant le plus grand ou plus petit résultat.
Afficher l'analyse biologique d'un type donné ayant le plus petit ou plus grand résultat et ayant eu lieu dans le premier ou dernier séjour.
Afficher les comptes-rendus indexés avec un concept donné ou éventuellement l'un de ses fils (i.e. explosion hiérarchique).
Afficher les actes médicaux d'un type donné ayant eu lieu juste avant ou juste après un autre acte médical donné.

TABLE 5.1 – Ensemble des cas d'usage ayant servi de base à la modélisation et au développement du SSE_{SQL}

5.2 Le virage du NoSQL

5.2.1 L'In Memory Data Grid qualifié

Infinispan [149, 150] est le système de stockage de données NoSQL qui a été retenu afin de palier aux faibles perfomances observées avec le SGBDR ORACLE®. C'est un cache mémoire distribué. Il permet par conséquent de stocker les données dans la mémoire RAM et de les distribuer au sein de plusieurs nœuds. Infinispan fait partie de la famille des **I**n **M**emory **D**ata **G**rids (IMDGs) [1]. Plusieurs critères ont guidé le choix pour cette solution. Premièrement, Infinispan est un logiciel libre et open-source [2] disponible sous la licence « Apache License 2.0 ». Informellement, c'est un magasin de données de type clé-valeur stockant les données au sein de tables de hachage et offrant par conséquence des performances optimales d'accès unitaire aux données. Il offre de plus des mécanismes de persistance des données. Dans un contexte distribué, Infinispan est avant tout et historiquement conçu pour favoriser la cohérence et la disponibilité des données (i.e. Système *AC* (Figure 4.10)) et peut être paramétré finement afin d'assouplir ces contraintes en cas de partitionnement du cluster. Peu de technologies NoSQL fournissent cette politique de gestion de la concurrence néanmoins profitable dans le cadre de la manipulation de données sensibles telles que les données de santé dont la cohérence doit être garantie. En terme de performances, Infinispan est, de plus, une technologie largement exploitée dans la littérature notamment dans le domaine biomédical (e.g. [151, 152]) et plus particulièrement pour ses performances [153].

5.2.2 Le changement de paradigme en action

Le stockage des données d'une base de données Infinispan s'effectue au sein de tables de hachage, ou « maps » permettant une association entre une **clé** et une **valeur** sous forme d'objets Java™ (Java™). Cette structure de données ne permet pas de définir de schéma de manière analogue aux SGBDRs. Afin de conserver la structuration générique de l'information définie par le modèle de données décrit dans précédement, un **P**lain **O**ld **J**ava **O**bject (POJO) [3] a été définie pour chaque table du MLD (Figure 5.2). La liste de ces tables est donnée Table 5.2. Ces dernières permettent de stocker, au niveau applicatif, le contenu des tables auxquelles ils se rapportent. Chaque tuple d'une table pouvant être, intégralement et identiquement structuré, stocké dans une instance de la classe Java™ associée à cette table (i.e. chaque classe Java™ possède un attribut pour chaque colonne de la table à laquelle cette classe se rapporte). Dans leur ensemble, ces objets permettent donc de reproduire l'intégralité de l'information du modèle logique de données en préservant ainsi la modélisation conceptuelle de cette dernière.

Tables	POJO
TB_OBJECT	DBObject.java
TB_DATATYPE_PROPERTY	DatatypeProperty.java
TB_OBJECT_PROPERTY	ObjectProperty.java
TB_INDEXING	Indexing.java
TB_HIERARCHY	Hierarchy.java
TB_MODEL	Model.java
TB_MODEL_DATATYPE_PROPERTY	ModelDatatypeProperty.java
TB_MODEL_OBJECT_PROPERTY	ModelObjectProperty.java
TB_MODEL_INHERITANCE	ModelInheritance.java

TABLE 5.2 – Correspondance entre les tables du modèle physique de données générique et les Classes Java™ qui en permettent leurs représentations au niveau applicatif.

1. : « Grille de Données en Mémoire »
2. : « code Source Ouvert »
3. : « Bon Vieil **O**bject **J**ava »

Ces POJOs sont construits à partir des données présentes dans le SGBDR PostgreSQL de l'EDS. Ils sont ensuite insérés et maintenus au sein de la base de données Infinispan à l'aide d'un SGBD spécifique appelé **NINJAC Is Not Just A Cache** (NINJAC) qui organise les accès à Infinispan au sein du SI du **D2IM**. Les POJOs de type DBObject et Model possèdent la particularité de pouvoir stocker (i.e. d'imbriquer) la liste des POJOs qui les concernent. Ainsi, un DBObject représentant conceptuellement un objet quelconque, permet éventuellement, si celui-ci a été préalablement complété, d'accéder directement à liste des attributs (i.e. la liste des DatatypeProperty) et des relations dans lesquelles cet objet conceptuel est impliqué (i.e. la liste de ObjectProperty et/ou Indexing et/ou Hierarchy). Il est néanmoins possible d'accéder aux POJOs DatatypeProperty, ObjectProperty et Hierarchy sans nécessairement passer par le DBObject auquel il se rapporte. En revanche, le POJO Model est lui systématiquement complété avec les POJOs ModelDatatypeProperty, ModelObjectProperty et ModelInheritance qui le concerne. Ainsi, du coté applicatif, la partie modèle du MLD est intégralement maintenue par l'ensemble des POJOs Model.

Infinispan, tout comme les solutions NoSQL de type clé-valeur d'une manière générale, ne fournit pas de mécanisme permettant de définir de modèle de données à proprement parler. Par conséquent, le modèle de données peut être vu comme résultant de la conjonction entre l'ensemble des maps constituant la base Infinispan d'une part et la structure des POJOs servant comme clés et valeurs dans ces dernières d'autre part. La Table 5.3 donne la liste des maps utilisées par le SI. Quant à la structure des objets utilisés comme clé et valeur, elle est héritée de la structure des tables du MLD.

Le système de gestion NINJAC permet principalement d'alimenter les maps de ce modèle et de mettre à disposition des méthodes permettant d'en extraire les POJOs de manière basique à partir des clés ou de manière un peu plus évoluée en complétant par exemple les DBObject avec les DatatypeProperty, ObjectProperty, Indexing et Hierarchy qui les concernent. Ce modèle est notamment composé de cinq « caches d'entité » donnant accès aux cinq POJOs destinés à la représentation de la partie donnée du MLD et d'un cache destiné à la partie modèle : le cache d'objets donnant accès aux DBObject, le cache d'attributs donnant accès aux DatatypeProperty, les caches de relations, d'indexations et de subsomptions donnant respectivement accès aux ObjectProperty, Indexing et Hierarchy et enfin le cache de modèle en charge de la représentation de la partie modèle du MLD. Les autres caches de la Table 5.4 sont des « caches système » qui servent davantage au fonctionnement interne de NINJAC et sont plus rarement exploités par des applications tierces requérant de l'information.

Cache d'objets		
Retrouver un objet à partir de son identifiant		
`"RDF_RESOURCE"`	⟶	DBObject
identifiant d'un objet.		*Instance Java™ de DBObject représentant l'objet ayant pour identifiant `"RDF_RESOURCE"`.*

Cache d'attributs		
Retrouver les attributs d'un objet à partir de l'identifiant de ce dernier.		
`"RDF_RESOURCE"`	⟶	[DatatypeProperty]
identifiant d'un objet.		*Liste des instances Java™ de DatatypeProperty représentant les attributs de l'objet d'identifiant `"RDF_RESOURCE"` définit dans la table `TB_DATATYPE_PROPERTY`.*

Cache de relations		
Retrouver une relation à partir de l'identifiant de son objet source.		
`"RDF_RESOURCE_SOURCE"`	⟶	[ObjectProperty]
identifiant d'un objet source d'une relation.		*Liste des instances Java™ de ObjectProperty représentant les relations de la table `TB_OBJECT_PROPERTY` dont l'objet source a pour identifiant `"RDF_RESOURCE_SOURCE"`.*

Cache d'indexations		
Retrouver une relation d'indexation à partir de l'identifiant de l'objet indexé.		
`"RDF_RESOURCE_INDEXED"`	⟶	[Indexing]
identifiant d'un objet source d'une relation d'indexation.		*Liste des instances Java™ de Indexing représentant les relations de la table `TB_INDEXING` dont l'objet indexé a pour identifiant `"RDF_RESOURCE_INDEXED"`.*

Cache de subsomptions		
Retrouver une relation hiérarchique à partir de l'identifiant de l'objet père.		
`"RDF_RESOURCE_BROADER"`	⟶	[Hierarchy]
identifiant d'objet source d'une relation hiérarchique.		*Liste des instances Java™ de Hierarchy représentant les relations de subsomption de la table `TB_HIERARCHY` dont l'objet père a pour identifiant `"RDF_RESOURCE_BROADER"`.*

Cache de modèles		
Retrouver un type à partir de son libellé.		
`"TYPE_ID"`	⟶	Model
identifiant d'objet source d'une relation hiérarchique.		*instance Java™ de Model représentant ce type d'objet, d'attribut, de relation ou ce type abstrait (et ayant donc `"TYPE_ID"` pour libellé).*

Légende :

`"xxx"`	Chaîne de caractère contenant le texte `xxx`.
Class	instance de la classe `Class.java`
$[X]$	Collection ordonnée d'éléments de type X.
$\{X\}$	Collection non ordonnée et sans doublons d'éléments de type X.

TABLE 5.3 – Liste des caches de données du SI NoSQL du **D2IM**. Pour chaque $Map : Clé \rightarrow Valeur$: les rôles de la map, de la clé et de la valeur sont explicités.

Cache d'identifiants Retrouver les identifiants des objets d'un type donné.		
`"TYPE_ID"`	$\longrightarrow$	{`"RDF_RESOURCE"`}
Libellé d'un type d'objet contenu dans la table `TB_OBJECT`.		*Ensemble des identifiants des objets de ce type `"TYPE_ID"`.*

Cache de types Retrouver le libellé du type d'un objet.		
`"RDF_RESOURCE"`	$\longrightarrow$	`"OBJECT_TYPE_ID"`
identifiant d'objet contenu dans la table `TB_OBJECT`.		*type de l'objet ayant cet identifiant `"RDF_RESOURCE"`.*

TABLE 5.4 – Liste des saches système du SI NoSQL du D2IM.

5.2.3 La recherche d'information

Comme tous les magasins de données de type clé-valeur, la solution Infinispan repose sur la structure de données de table de hachage. Cette dernière fournit une API de base qui permet une sélection et une insertion de base des données. Cette API permet l'insertion d'un couple $(clé, valeur)$ dans une map (i.e. opération de `put`) ou bien la récupération d'une valeur à partir de sa clé (i.e. opération de `get`). Elle est exploitée par le système de gestion de données NINJAC afin d'assurer la robustesse, la maintenance et l'accès de base aux données du SI.

NINJAC ne fournit, en revanche, pas de fonctionnalités de requêtage avancé. Lorsque un SI repose sur un SGBDR, les fonctionnalités de RI sont assurées par l'utilisation du SQL. Cependant, très peu de solutions NoSQL fournissent la possibilité d'utiliser ce langage de requête. De plus, ceux qui le permettent (e.g. Apache Ignite[4]), n'implémentent pas les fonctionnalités de ce langage sur la partie NoSQL sous-jacente mais sur un SGBDR traditionnel répliquant les données n'assurant ainsi alors pas nécessairement de meilleures performances et impliquant ainsi de plus lourdes contraintes en terme d'espace mémoire.

Infinispan, offre la possibilité d'indexer les données en utilisant Apache **Lucene** (Lucene)[5][154] et fournit plusieurs APIs de requêtage dépendant du contexte d'utilisation :

Ickle : qui est le langage de requête Booléen propre à Infinispan. C'est un sous-ensemble léger de **JP–QL** muni d'une extension pour la recherche plein texte qui peut être utilisée à la fois en mode client ou en mode embarqué sur des données indexées ou non. Ce dernier ne supporte cependant ni le calcul à la volée d'expressions algébriques, ni les jointures entre entités, ni les sous-requêtes.

L'API de requêtage Hibernate : qui permet en mode embarqué de requêter les données indexées en générant des requêtes Lucene.

L'« Infinispan Query Domain–Specific Language (DSL)[6] » : qui fonctionne lui également en mode client (en plus du mode embarqué) en créant programmatiquement des requêtes mais qui ne permet pas d'effectuer de la recherche plein texte et ne supporte pas davantage les jointures.

Aucune de ces APIs ne fournit à elle seule l'intégralité des fonctionnalités nécessaire à la mise en place d'une RI. Elles reposent, de plus, sur des modes de requêtage différents (i.e. constitution

4. url : `https://ignite.apache.org/index.html`
5. url : `https://lucene.apache.org/`
6. : « Language dédié »

de requêtes sous forme de chaînes de caractères vs. de façon programmatique). Enfin aucune méthode native ne permet de gérer efficacement les jointures entre entités qui reste néanmoins une fonctionnalité essentielle à la RI au sein de données de santé.

Afin de garantir l'implémentation de ces fonctionnalités, deux **maps de jointure** sont maintenues par NINJAC. Ces dernières s'ajoutent aux maps de base du SI (Table 5.3) et sont décrites dans la Table 5.5. Elles permettent de garantir la recherche des relations classiques et des relations d'indexation relative respectivement aux tables `TB_OBJECT_PROPERTY` et `TB_INDEXING` du MLD. La map permettant de requêter les relations d'indexation est basée, notamment en ce qui concerne la structure de ses clés, sur la notion d'indexation adoptée par l'équipe et prenant en compte le caractère majeur ou mineur de cette dernière ainsi que les qualificatifs et types de ressources affiliés (cf. remarque 4). D'une manière générale, ces deux maps permettent de retrouver les identifiants cible (resp.source) d'une relation en spécifiant (au niveau de la clé), le type de relation ainsi que l'identifiant source (resp.cible) de cette dernière. Certaines relations donnent effectivement lieu à deux entrées au sein de ces maps afin que la relation puisse être parcourue de manière symétrique.

L'ensemble des maps composant le modèle de données NoSQL du SI permettent de rechercher les différents objets Java™ en exploitant leurs éléments de structure (i.e. leurs identifiants et les relations qu'ils entretiennent entre eux). Ces dernières jouent le rôle d'index classique permettant de retrouver des entités par l'intermédiaire d'identifiants. En revanche, la recherche des entités par l'intermédiaire de leurs contenus (attribut textuel, numérique ou de type date) n'est pas assurée par ces index. Lucene a été utilisé afin de générer des **index inversés** permettant la recherche des entités portant sur les différents champs composant les entités de type DatatypeProperty. Ces index permettent d'effectuer une recherche plein texte portant sur les attributs de type DatatypeProperty.

Nativement, le SGBD NoSQL orienté Clé-Valeur Infinispan ne fournit aucun mécanisme de définition d'une quelconque modélisation des données. Celle-ci est en réalité intrinsèquement fixée par la structure des objets Java™ employés comme valeurs des entrées des différentes maps contenues par ce SGBD. Afin de préserver la modélisation générique définie par le MLD de la Figure 5.2, les valeurs des maps de NINJAC ont été restreintes à des POJOs simulant la structure des différentes tables de ce modèle. En d'autres termes, la modélisation définie à l'aide de tables et de colonnes au sein du SGBDR a été transposée au niveau du SGBD NoSQL NINJAC à l'aide d'objets Java™ possédant une structure équivalente.

Bien que cette modélisation ait pu être réalisée, l'API de base fournie nativement par Infinispan ne permet pas de l'exploiter. Certains outils de requêtage visant à pallier à ce problème existent mais n'autorisent pas une sélection suffisamment exhaustive des données pour répondre à la totalité des besoins d'information relatifs aux données cliniques. Nous avons donc créé des maps spécifiques assimilables à des index inversés qui constituent un ensemble de structures dont l'exploitation fournit l'ensemble des fonctionnalités de recherche requis. Néanmoins, si ces structures offrent des possibilités techniques de recherche des données, chaque besoin d'information requiert une utilisation spécifique et organisée de ces maps. Ceci peut être réalisé grâce à un langage de requête qui fournit une syntaxe permettant à la fois d'exprimer et d'identifier les informations désirées.

Dans le cadre de mon travail de thèse, j'ai ainsi définit le langage $\mathcal{L}_{\text{share}}$ afin de remplir cet objectif. Ce dernier est décrit dans le chapitre suivant.

Cache de jointures relationnelles

Retrouver les objets liés à un autre par un type de relation défini.

`"TYPE_ID"` + `"RDF_RESOURCE_SOURCE"`	⟶	{`"RDF_RESOURCE_TARGET"`}
Clé composée d'un identifiant (i.e. un type) de relation et d'un identifiant d'un objet.		*Ensemble des identifiants des objets liés à l'objet d'identifiant `"RDF_RESOURCE_SOURCE"` par une relation de type `"TYPE_ID_RELATION"`.*

Cache de jointures d'indexations

Retrouver les objets indexés avec un concept terminologique (ou plus généralement un autre objet).

`"TYPE_ID"` + `"RDF_RESOURCE_INDEX"` + `"RDF_RESOURCE_INDEX_AFF"` + `"RDF_RESOURCE_INDEX_AFF_TR"` + $\texttt{MAJEUR}_{0/1}$ + $\texttt{EXPLOSION}_{0/1}$	⟶	{`"RDF_RESOURCE_INDEXED"`}
Clé décrivant permettant de spécifier : — *le type de relation d'indexation désiré ;* — *l'identifiant du concept terminologique (ou de l'objet) indexant les ressources recherchées (e.g. « maladies de l'animal » [D000820 (MeSH)]) ;* — *optionnellement, le qualificatif de l'indexation (e.g. « thérapie » [Q000628 (MeSH)]) ;* — *optionnellement, le type de ressources de l'indexation (e.g. « documents » [TR39 (CiSMeF)]) ;* — *le caractère majeur/mineur désiré pour l'indexation ;* — *si les ressources peuvent être indexées avec un fils hiérarchique du concept indexant indiqué (i.e. explosion hiérarchique).*		*Ensemble des identifiants des objets indexés, par l'intermédiaire de la relation d'indexation de type `"TYPE_ID"`, en majeur ou en mineur (selon la valeur de $\texttt{MAJEUR}_{0/1}$) avec le concept d'identifiant `"RDF_RESOURCE_INDEX"` (ou l'un de ses fils si $\texttt{EXPLOSION}_{0/1}$ est à `true`) et éventuellement le qualificatif `"RDF_RESOURCE_INDEX_AFF"` et le type de ressources `"RDF_RESOURCE_INDEX_AFF_TR"`.*

TABLE 5.5 – Les deux maps de jointure utilisées dans le cadre des tâches de RI afin de parcourir les relations classiques et les relations d'indexations. L'utilité globale de chaque map et les rôles pris par la clé et la valeur sont explicités.

Chapitre 6

Le langage de requête $\mathcal{L}_{\mathcal{G}_<}$

Sommaire

6.1 Description de la syntaxe 130
6.1.1 Clause entité 131
6.1.2 Les types de données 132
6.1.3 Contraintes d'attributs 133
6.1.4 Contraintes sémantiques 138
6.1.5 Propriétés algébriques du langage 146
6.2 Rappels sur les langages formels 152
6.2.1 Alphabet et mots 152
6.2.2 Langages 154
6.2.3 Grammaire 155
6.3 La grammaire $\mathcal{G}_{\mathcal{G}_<}$ 158
6.3.1 Le grammaire réduite $\mathcal{G}^{\times}_{\mathcal{G}_<}$ 158
6.3.2 $\mathcal{G}_{\mathcal{G}_<}$, l'intégrale 162
6.4 Implémentation 165
6.4.1 Le générateur JavaCC™ 165
6.4.2 Exploitation du parseur 166

Le **M**odèle **L**ogique de **D**onnées (MLD) décrit dans la section 5.1 ainsi que sa « traduction » au sein de l'environnement NoSQL (section 5.2) permet non seulement de stocker tout type d'informations sans altération structurelle du modèle de données mais aussi et surtout de maîtriser la vision conceptuelle que l'on souhaite donner à cette information. Ce dernier permet de préserver et de rendre exploitable l'information de santé sous forme d'un **réseau sémantique d'information**[1] représentable à l'aide d'un formalisme entité–association et, de manière plus générique, assimilable à un graphe de données reliant de multiples entités par des relations sémantiques. Ce stockage et cette modélisation générique de l'information ne fournit cependant, en soit, pas de fonctionnalités de RI. Le rôle du travail décrit dans ce mémoire est précisément de tirer partie de l'expressivité sémantique permise par le modèle de données du D2IM dans le cadre de la RI. L'idée de base étant en effet de bénéficier de la structure de graphe de données en adoptant une « philosophie » d'interrogation calquée sur cette sémantique et cette généricité.

Comme évoqué précédemment dans ce mémoire, la sémantique des données de santé fait intervenir de multiples notions, représentées conceptuellement par des entités/objets (e.g. des **patients**, des **séjours**, des **unités médicales**, des **analyses biologiques**, etc.). D'un point de

1. La notion de réseau sémantique est ici employée de manière générique. Elle ne se réfère pas uniquement et spécifiquement à la notion de sémantique classiquement assimilée à l'exploitation de TOs. Ici, l'expression « **réseau sémantique d'information** » désigne plus largement un ensemble d'unités d'informations porteuses de sens et reliées entre elles de manière cohérente d'un point de vue humain. Cette vision structurée de l'information se traduit notoirement par une représentation de l'information en terme d'entités reliées par des relations qui forment ainsi un réseau.

vue médical ou clinique, ces entités ne prennent bien souvent pleinement leurs sens et/ou leurs utilités que lorsqu'elles sont vues de manière interconnectées (e.g. une analyse biologique n'a en soit que peu de valeur informative si l'on ne la rattache pas au patient auquel elle appartient).

La fusion « tout-en-un » des informations relatives à ces diverses entités reste cependant difficile compte tenu de la diversité des cas d'usage d'un EDS qui impose potentiellement des besoins de séparations structurelles de l'information incompatibles. L'hétérogénéité même des types d'information présents rend leurs agrégations techniquement parfois incohérentes et opérationnellement inexploitables. L'emploi soutenu de modèles de données de type **E**ntity–**A**ttribute–**V**alue (EAV) [2], que ce soit dans le cadre de la recherche clinique ou de l'amélioration de la prise en charge des patients, est par ailleurs symptomatique de cette incapacité à modéliser l'information de santé de manière « figée ».

Dans ce contexte, les bénéfices potentiels d'un SRI basé sur un EDS dépendent donc de la capacité de ce système à s'accorder avec cette réalité. En d'autres termes, un tel système doit être en mesure de répondre à des besoins d'information complexes formulés en terme d'atomes d'information qui doivent néanmoins pouvoir être mise en relation.

Lorsqu'il s'agit de requêter les données d'un SGBDR la norme en terme de langage de requête est le SQL. Bien que le SQL soit un langage de requête puissant, offrant de nombreuses possibilités de sélections avancées des données et facile d'utilisation dans l'absolu, il n'en reste pas moins que ce langage de requête a été conçu pour des utilisateurs ayant des connaissances informatiques. Il est donc peu pragmatique d'envisager le SQL comme langage de requête pour l'accès aux données d'un SI. La plupart des SRIs basés sur un EDS reposant sur un SGBDR sont alors couplés à des interfaces graphiques fournissant des fonctionnalités prédéfinies. Classiquement, l'accès à ces dernières se fait à travers des formulaires générant des requêtes SQL préconçues et paramétrables. Ce sont alors principalement ces requêtes prédéfinies qui ont la charge d'effectuer les jointures adéquates entre les différentes entités composant l'information de santé à cibler. Parmi les SRIs existants dans la littérature, on retrouve notamment cette stratégie avec des niveaux de spécificité variables. SMEYEDAT [80] par exemple fournit une interface extrêmement spécialisée permettant de répondre à des cas d'usage ophtalmologiques. STRIDE [1], quant à lui, propose une recherche plus générique permettant, par exemple, de gérer la concomitance de certains événements mais propose néanmoins plusieurs interfaces suivant le cas d'usage (e.g. extraction de patient, extraction d'information clinique, etc.). Enfin i2b2 [68, 69], et l'outil qui lui est dédié (le Workbench Query Tool), proposent un niveau de généricité plus étendu en permettant à l'utilisateur d'influer sur l'étendue des données accessibles via les fonctionnalités de RI de l'outil en structurant de manière adéquate les méta-données. Ces outils restent cependant, malgré tout, globalement centrés autour de la notion de patient. Il sont d'ailleurs majoritairement conçus pour fournir des données agrégées (e.g. moyennes, écarts type, graphiques, etc.) relatives à la liste de patients extraite (cf. chapitre 2). De plus, l'étendue des données sujettes à de la RI est tributaire de la structuration de ces données dans la base de données imposant parfois une modification de la structure conceptuelle des données afin d'en permettre l'intégration et la recherche (e.g. [71])

En somme, le langage SQL permet la mise en place d'une RI efficace mais reste néanmoins un langage technique, muni d'un formalisme faisant intervenir les notions du modèle relationnel. Ce langage ne peut être exploité par les professionnels de santé qu'à travers des interfaces répondant à des problématiques spécifiques qui peuvent être générisées mais qui s'articulent néanmoins autour de la notion de patient. Un des objectifs de mes travaux est de permettre une expression des besoins d'information plus globale et plus générique, ne considérant plus le patient comme le concept central de l'information de santé d'un EDS mais comme une entité parmi d'autres reliée au sein d'un réseau d'information complexe. Cette vision permet notamment de s'accorder avec davantage de cas d'usage et de répondre à divers besoins d'information. Ceci requiert un langage intermédiaire permettant de s'abstraire de la vision structurelle des données au profit d'une vision conceptuelle, en fonction du formalisme entité-association des données, et non basé

2. : « **E**ntité–**A**ttribut–**V**aleur »

directement sur un formalisme technique hérité de la structuration physique des données.
Enfin, un tel langage s'avère essentiel techniquement lorsque le stockage des données est effectué au sein d'un **I**n **M**emory **D**ata **G**rid (IMDG) [3] tel qu'Infinispan qui n'offre, lui, pas la totalité des fonctionnalités requises pour effectuer pleinement une RI efficace (comme vu dans le chapitre précédent) et impose la modélisation et l'implémentation annexe de ces fonctionnalités.

> **Notation 4 :**
> *Une partie non négligeable du travail effectué dans le cadre de cette thèse est la conception d'un langage de requête spécifique. Dans ce mémoire, on notera $\mathcal{L}_{\text{}}$ ce langage, $\mathcal{G}_{\text{}}$ la grammaire formelle qui engendre ce langage et enfin $\mathcal{L}_{\text{}}$–requête les **requêtes** écrites dans le langage $\mathcal{L}_{\text{}}$.*

3. : « Grille de Données en Mémoire »

6.1 Description de la syntaxe

En pratique, un langage de requêtes joue le rôle d'un langage de communication entre un humain et une source de données telle qu'une base de données ou un SI quelconque. Plus précisément, il fournit une syntaxe permettant à un utilisateur humain d'exprimer au travers d'une requête textuelle la nature et les caractéristiques des données qu'il souhaite obtenir du SI.

La syntaxe du SQL s'articule autour de la structure de la base de données. La création de requêtes à l'aide de ce langage nécessite une parfaite connaissance du modèle de la base de données (i.e. tables, colonnes, etc.) ainsi que la maîtrise de certaines notions techniques telle que la notion de jointure par exemple. Ce langage est donc dépendant de la structure concrète des données et permet de constituer des requêtes dont la philosophie de syntaxe est **orientée sur la structure des données**.

Le langage de requête que je propose possède lui, a contrario, une syntaxe **orientée entités**. Celle-ci permet de formuler des requêtes indépendamment de la structure physique des données ciblées par ces dernières. Les éléments de base manipulés par ce langage sont en réalité les diverses entités conceptuelles que représentent implicitement les données et non les données elles mêmes. Ces entités correspondent à celles du MCD. Dans une modélisation relationnelle classique, le MCD correspond approximativement au MLD et peut en être déduit [4]. Dans le contexte de cette thèse, la déduction du MCD est en revanche immédiate (Figure 6.1). En effet, le MLD du SI du **D2IM** généralise le formalisme de représentation entité–association lui-même et non un MCD particulier qui lui, n'en est qu'une instance.

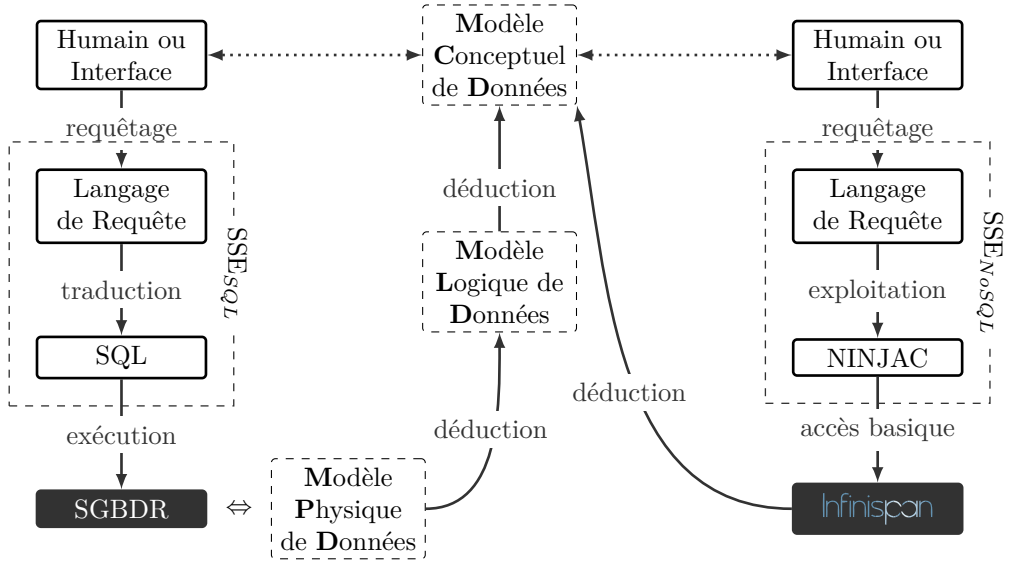


FIGURE 6.1 – Positionnement et rôle du langage de requête dans la chaîne d'interrogation des données de l'EDSS. Les deux preuves de concept exploitant respectivement une base de données relationnelle et une base de données NoSQL Infinispan sont représentées.

Cette section s'attache à décrire de manière pragmatique le langage de requête $\mathcal{L}_{\bullet}$ du SSE$_{SQL}$ et du SSE$_{NoSQL}$. Ayant été initialement imaginé dans un cadre du projet RAVEL [ANR–11–TECS–012] et de la problématique de RI au sein du DPI, ce cas d'usage sera repris comme exemple support à la description de la syntaxe ainsi que de la logique syntaxique du langage de requête. Une partie du MCD exploité durant la modélisation et le développement du langage est donnée Figure 6.2.

Ce dernier se présente sous forme d'un graphe orienté, étiqueté et attribué (i.e. chaque entité peut être muni d'attributs, les arcs sont orientés et possèdent une étiquette).

La syntaxe de base du langage s'articule autour ces notions d'entités, d'attributs d'entités et d'arcs (i.e. de relations) entre ces entités. Bien que le moteur de recherche SSE$_{NoSQL}$ repose

4. Ainsi, il existe par exemple théoriquement une table pour chaque type d'entité

sur ce langage aujourd'hui, ce dernier a été initialement conçu comme un langage intermédiaire entre le SQL et l'humain ou une quelconque interface graphique (Figure 6.1).

Je décris dans cette section le langage $\mathcal{L}_{\text{}}$ imaginé, modélisé et implémenté comme langage d'interaction avec les moteurs de recherche SSE_{SQL} et SSE_{NoSQL}. Les deux sections suivantes (viz. section 6.2, section 6.3) permettent de définir de manière complète et rigoureuse ce langage $\mathcal{L}_{\text{}}$. Les exemples donnés dans cette section font tous référence au graphe de données de la Figure 6.2.

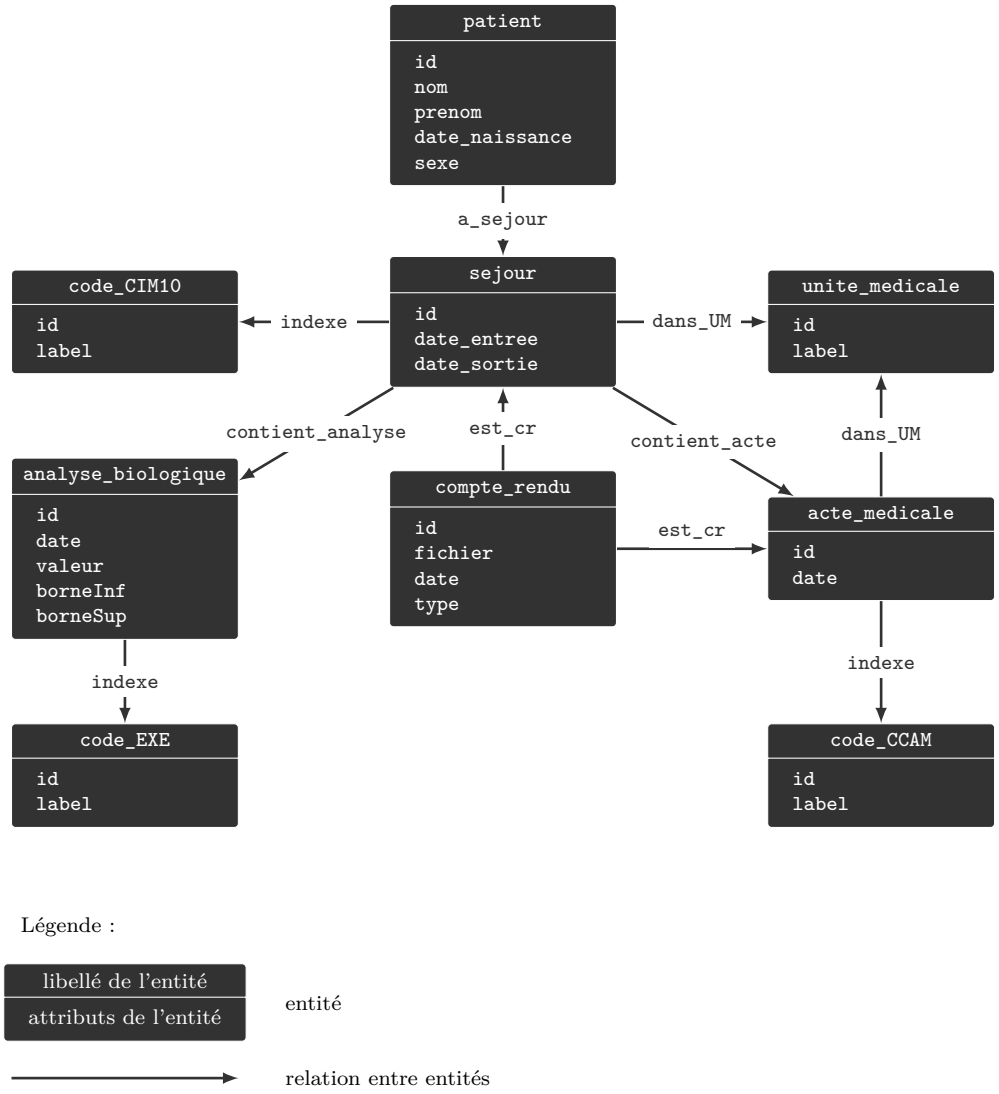


FIGURE 6.2 – Extrait du MCD représenté sous forme d'un graphe de données orienté, étiqueté et attribué. Dans ce graphe, une simplification d'une partie des données de l'EDSS est représentée conceptuellement. L'entité patient est liée à des séjours au sein desquels peuvent être pratiqués des examens biologiques et des actes médicaux. Les actes médicaux et les séjours peuvent être effectués dans des unités médicales distinctes. Les actes sont rattachés à des codes de la CCAM.

6.1.1 Clause entité

Les **clauses entité** sont les « éléments syntaxiques de base » à partir desquels le langage de requête est construit. Elles constituent, d'ailleurs, l'élément minimal requis pour former une requête dans ce langage. Comme il le sera précisé dans la suite, c'est ce formalisme syntaxique de base qui permet, par imbrication récursive, de former des requêtes plus complexes permettant de naviguer au sein du réseau sémantique des données. Une clause entité suit une syntaxe de la forme :

</> Syntaxe 1 (Clause entité) :
Une ***clause entité d'entité*** `ENTITY` *est une expression de la forme :*

`ENTITY(CONSTRAINT)`

où :

`ENTITY` *désigne un libellé d'entité ;*

`CONSTRAINT` *désigne une expression composée de contraintes atomiques.*

Formellement, une clause entité exprime, désigne ou cible un **ensemble d'entités** respectant certaines **contraintes**. La sous-clause `ENTITY` spécifie le (ou les types) d'entité(s) ciblée(s) tandis que la sous-clause `CONSTRAINT` définit un certain nombre de contraintes respectées par cette (ces) entité(s). Plus précisément :

La sous-clause `ENTITY` se matérialise syntaxiquement par un libellé d'entité. Elle a pour but de définir « les entités à requêter » ou plus généralement les entités « ciblées » ou « désignées » par la clause entité dans son ensemble. Dans le cadre de notre exemple (Figure 6.2), `ENTITY` pourrait par exemple correspondre aux types `patient` ou `sejour`. Dans l'absolu, la grammaire du langage de requête accepte qu'une sous-clause `ENTITY`d'une clause entité `ENTITY(CONSTRAINT)` fasse intervenir plusieurs libellés d'entités séparés par des virgules (« `,` ») (cf. exemple 7). La clause entité dans son ensemble cible alors plusieurs entités à la fois. Cependant, l'utilisation de clauses entités désignant plusieurs entités n'est, en pratique, pertinente que dans des cas bien particulier relativement anecdotiques.

> **⚠ Remarque 5 :**
> *Dans un souci de simplicité, si aucune indication n'est fournie, toute clause entité* `ENTITY(CONSTRAINT)` *pourra par la suite être considérée comme ne ciblant qu'une seule et même entité.*

La sous-clause `CONSTRAINT` définit les contraintes qui devront être vérifiées par l'entité ciblée. Cette contrainte est une expression logique construite à l'aide d'une part des opérateurs Booléens `AND` et `OR` et de contraintes de deux types qui seront détaillés par la suite : les **contraintes d'attributs** et les **contraintes sémantiques**.

Dans sa forme la plus simple, une clause entité peut ne pas avoir de contrainte et donc se limiter à une syntaxe de la forme `ENTITY()`. Un moteur de requête implémentant le langage de requête pourra alors interpréter la requête en renvoyant la totalité des entités de type `ENTITY` (cf. exemple 6).

> **✎ Exemple 6 :**
> *la requête* `patient()` *désigne l'ensemble de tous les patients contenus dans la source de données.*

> **✎ Exemple 7 :**
> *la requête* `analyse_biologique,acte_medical()` *désigne l'ensemble de tous les actes médicaux et toutes les analyses biologiques contenus dans la source de données.*

6.1.2 Les types de données

Avant de décrire les différentes formes que peuvent prendre les contraintes, il est nécessaire d'introduire les données typées gérées d'un point de vue purement syntaxique par le langage de requête. Ce dernier prévoit la représentation syntaxique de trois types de données :

les données de type « Numérique » : elles sont représentées par la valeur du numérique en question avec un point séparant la partie décimale de la partie entière. Par exemple `3.14159265`.

les données de type « Chaîne de caractère » : elles sont délimitées par des guillemets anglais « " » (par exemple : `"chaîne de caractère"`) et ne peuvent donc pas contenir de guillemets anglais.

les données de type « Date » : elles sont représentées au format `YYYY-MM-DD HH:mm:ss`. Par exemple, la date du 25 Décembre 2015 à 23 heures 30 minutes et 15 secondes prend la syntaxe `2015-12-25 23:30:15`. La partie de la syntaxe spécifiant les heures, minutes et secondes (i.e. `HH:mm:ss`) est cependant optionnelle. Par exemple, la syntaxe `2015-12-25` est valide et correspond à la syntaxe complète `2015-12-25 00:00:00`.

Le langage de requête fournit syntaxiquement plusieurs **comparateurs** et **opérateurs** pour chacun de ces trois types de données. Ces derniers peuvent être utilisés pour former des contraintes. La Table 6.1 donne la liste de tous ces comparateurs et opérateurs. Comme leurs noms l'indiquent, les opérateurs permettent de composer des données de même type (e.g. lorsqu'il est nécessaire de représenter syntaxiquement la somme ou la différence de deux attributs ou d'un attribut et d'une valeur fixée) et les comparateurs permettent de définir une contrainte basée sur une comparaison (e.g. lorsqu'il s'agit de comparer la valeur d'un attribut d'entité avec une valeur fixée).

Types de données	Opérateurs disponibles	Comparateurs disponibles
Chaînes de caractères	Aucun	`+` egalité `!=` non égalité
Numériques	`+` addition `-` soustraction `*` multiplication `/` division	`=` égalité `!=` non égalité `<` infériorité stricte `<=` infériorité `>` supériorité stricte `>=` supériorité
Dates	`+` addition `-` soustraction (remarque 6)	`=` égalité `<` infériorité stricte `<=` infériorité `>` supériorité stricte `>=` supériorité

TABLE 6.1 – Opérateurs et comparateurs disponibles pour les différents types de données.

> ⚠ **Remarque 6 :**
> *Il est à noter que, le résultat de la composition de deux dates par l'opérateur de soustraction « - » n'est pas supposé être une date dans ce langage de requête mais un numérique correspondant à la différence en jours entre les deux dates soustraites conformément à ce qui est la convention dans le langage SQL.*

6.1.3 Contraintes d'attributs

Étant donnée une clause entité `ENTITY(CONSTRAINT)` (syntaxe 1), la sous-clause de contrainte `CONSTRAINT` peut prendre diverses formes complexes. Cette partie aborde les contraintes d'attributs qui en sont la forme la plus simple et qui, comme leurs noms l'indiquent, permettent de spécifier des contraintes portant sur les valeurs des attributs de l'entité désignée par la sous-clause `ENTITY`.

6.1.3.1 Syntaxe simplifiée

La syntaxe de base permettant de construire une contrainte d'attribut est la suivante :

</> Syntaxe 2 (Contrainte d'attribut atomique) :
Une expression de la forme :

$$\texttt{BLOCK}_1\ \texttt{COMPARATOR}\ \texttt{BLOCK}_2$$

est une ***contrainte d'attribut d'une entité*** `ENTITY` *lorsque :*

1. *elle est utilisée au sein de la clause de contrainte d'une clause entité d'entité* `ENTITY`

$$\texttt{ENTITY(}\ \ldots\ \texttt{BLOCK}_1\ \texttt{COMPARATOR}\ \texttt{BLOCK}_2\ \ldots\ \texttt{)}$$

2. *les blocs* $\texttt{BLOCK}_1$ *et* $\texttt{BLOCK}_2$ *sont au choix :*
   - *une valeur de type numérique, chaîne de caractère ou date ;*
   - *un libellé identifiant un attribut particulier d'une entité.*
3. *le comparateur* `COMPARATOR` *est un comparateur de la Table 6.1.*

> **⚠ Remarque 7 :**
> *En réalité, cette syntaxe est un cas particulier de la syntaxe 3 p. 136. Par souci de clarté, la syntaxe ici présentée est cependant prise comme support d'explication dans un premier temps.*

Dans la pratique, une clause entité `ENTITY(CONSTRAINT)` dont la sous-clause `CONSTRAINT` se résume à une contrainte d'attribut atomique est donc de la forme :

$$\texttt{ENTITY(BLOCK}_1\ \texttt{COMPARATOR}\ \texttt{BLOCK}_2\texttt{)}$$

Un libellé d'attributs est intrinsèquement typé. Par exemple, l'attribut `date_naissance` de l'entité `patient` est intrinsèquement du type date. Même si d'un point de vue purement syntaxique cela ne représente pas une nécessité, une contrainte d'attribut n'est cohérente qu'à condition que les deux sous-clauses $\texttt{BLOCK}_1$ et $\texttt{BLOCK}_2$ soient du même type. Le comparateur `COMPARATOR` peut alors correspondre à n'importe quel comparateur introduit dans la Table 6.1 et disponible pour le type de données commun à ces deux blocs. De même, lorsque qu'un bloc d'une expression du type $\texttt{ENTITY(BLOCK}_1\ \texttt{COMPARATOR}\ \texttt{BLOCK}_2\texttt{)}$ désigne un attribut d'entité, cette expression ne prend de sens que si l'attribut en question est un attribut de l'entité désignée par `ENTITY`.

Deux cas de figures peuvent dans la pratique se présenter. Si dans une clause entité $\texttt{ENTITY(BLOCK}_1\ \texttt{COMPARATOR}\ \texttt{BLOCK}_2\texttt{)}$:

— $\texttt{BLOCK}_1$ (resp. $\texttt{BLOCK}_2$) désigne un attribut de l'entité `ENTITY` et que $\texttt{BLOCK}_2$ (resp. $\texttt{BLOCK}_1$) est une valeur typée alors cette clause entité dans son ensemble permet de cibler l'ensemble des entités de type `ENTITY` dont l'attribut en question (i.e. $\texttt{BLOCK}_1$ (resp. $\texttt{BLOCK}_2$)) vérifie une contrainte relativement à une valeur fixe (i.e. $\texttt{BLOCK}_2$ (resp. $\texttt{BLOCK}_1$)).

> **✎ Exemple 8 (contraintes attribut-comparateur-valeur) :**
> *La contrainte* `sexe = "F"` *permet de spécifier une valeur à l'attribut* `sexe` *de l'entité* `patient`*. La clause entité* `patient(sexe = "F")` *désigne alors l'ensemble des patients de sexe féminin. Dans cette contrainte, relativement à la syntaxe 2 :*
>
> `sexe` *est un libellé d'attribut unique constituant le bloc* $\texttt{BLOCK}_1$ *;*
>
> `=` *constitue le* `COMPARATOR` *(comparateur d'égalité entre deux chaînes de caractère) ;*
>
> `"F"` *est une valeur de type « chaîne de caractère » constituant le bloc* $\texttt{BLOCK}_2$*.*
>
> *Le même type de contrainte peut être construite avec les types de données Date et*

Numérique. Par exemple la clause entité `sejour(date_entree = 2006-09-05)` *désigne l'ensemble des séjours dont la date d'entrée est le* 5 *Janvier* 2006 *(comparaison de deux dates). De même, la clause entité* `analyse_biologique(valeur > 3.5)` *désigne l'ensemble des analyses biologiques dont la valeur de l'analyse est supérieure à* 3.5 *(comparaison de deux numériques).*

— $\texttt{BLOCK}_1$ et $\texttt{BLOCK}_2$ désignent tous deux un attribut de l'entité `ENTITY` alors la clause entité dans son ensemble permet de sélectionner les entités de type `ENTITY` dont la valeur du premier attribut (i.e. $\texttt{BLOCK}_1$) vérifie une contrainte relativement à celle de l'autre (i.e. $\texttt{BLOCK}_1$).

Exemple 9 (contraintes attribut-comparateur-attribut) :
La clause entité `analyse_biologique(valeur < borneInf)` *désigne les analyses dont le résultat est inférieur à la borne inférieure et donc l'ensemble des analyses qui sont « anormalement basses ». Les deux blocs* `valeur` *et* `borneInf` *de la contrainte de cette clause entité désignent tous deux des attributs de l'entité* `analyse_biologique`*. La contrainte de cette clause entité prend donc la forme d'une comparaison entre deux attributs d'entité de type numérique.*

6.1.3.2 Multi-valuation

Lorsqu'il est nécessaire de multi-valuer la valeur d'un attribut d'entité, il est possible de le faire en concaténant l'ensemble des valeurs en question et en les séparant par une virgule « `,` ». La multi-valuation d'une valeur s'apparente à un « `OU` » logique. En d'autre termes, lorsque dans une contrainte d'attribut une valeur multi-valuée est assignée à un attribut, la clause entité contenant cette contrainte désigne alors l'ensemble des entités dont l'attribut vérifie la contrainte avec au moins une des valeurs présente dans la multi-valuation.

Exemple 10 (multi-valuation) :
`patient(prenom="Jacques","Jean","Daniel")` *désigne l'ensemble des patients dont le prénom est « Jacques » ou bien « Jean » ou encore « Daniel ». Il est également possible de multi-valuer une valeur de type date (*`sejour(date_entree=2010-01-01,2010-01-02)`*) ou une valeur de type numérique (*`analyse_biologique(valeur=1.1,2.2,3.3)`*)*

6.1.3.3 Expressions arithmétiques

Les blocs peuvent également prendre la forme d'**expressions arithmétiques** pour créer des contraintes plus évoluées. Ces expressions arithmétiques sont alors construites en composant des attributs et des valeurs de même type par l'intermédiaire des opérateurs arithmétiques disponibles pour ce type de données commun (cf. Table 6.1). Deux exemples sont donnés ci-dessous :

Exemple 11 :
La clause entité `sejour(date_sortie - date_entree >= 10.0)` *désigne l'ensemble des séjours dont la durée est supérieure ou égale à* 10 *jours. La contrainte considérée dans cet exemple est donc* `date_sortie - date_entree >= 10.0` *dans laquelle :*

- *le bloc* `date_sortie - date_entree` *de cette dernière se présente sous forme d'une expression arithmétique effectuant la soustraction de deux dates et permet d'évaluer la différence en jour entre la date d'entrée et la date de sortie d'un séjour ;*
- *le deuxième bloc (le bloc* `10.0`*) est une simple valeur numérique ;*
- *« `>=` » est quant à lui le comparateur « supérieur ou égal à » permettant de comparer deux numériques.*

Exemple 12 :
La contrainte considérée dans cette exemple est la suivante :

```
date_sortie - date_entree >= 10.0
```

Le bloc `date_sortie - date_entree` *de cette dernière se présente sous forme d'une expression arithmétique effectuant la soustraction de deux dates et permet d'évaluer la différence en jour (cf. remarque 6 p. 133) entre la date d'entrée et la date de sortie d'un séjour. Le deuxième bloc (le bloc* `10.0`*) est une simple valeur numérique. « >= » est quant à lui le comparateur « supérieur ou égal à » permettant de comparer deux numériques. En somme, La clause entité* `sejour(date_sortie - date_entree >= 10.0)` *désigne l'ensemble des séjours dont la durée est supérieure ou égale à* 10 *jours.*

Exemple 13 :
Dans la clause entité `analyse_biologique(valeur > 2*borneSup)`*, la contrainte est constituée d'un bloc de gauche se résumant à l'attribut* `valeur` *de l'entité* `analyse_biologique`*, d'un comparateur de numérique (*`>`*) et d'une expression arithmétique comme bloc de droite. Cette dernière correspondant au double de la borne supérieure de l'analyse biologique. La clause entité dans son ensemble désigne l'ensemble des analyses biologiques dont le résultat est deux fois supérieur à la normale.*

6.1.3.4 Syntaxe générique

Le langage de requête accepte en réalité des contraintes suivant une syntaxe plus générique que la syntaxe 2. Une contrainte peut contenir plusieurs blocs de même type séparés par des comparateurs en suivant la syntaxe :

</> Syntaxe 3 :
Dans sa forme la plus générique, une ***contrainte d'attribut*** *est de la forme :*

$$BLOCK_1\ COMPARATOR_{12}\ BLOCK_2\ COMPARATOR_{23}\ BLOCK_3\ \ldots\ BLOCK_n$$

En réalité, comme il le sera précisé dans la suite les possibilités qu'offre cette syntaxe ne sont pas indispensables en terme d'expressivité du langage car l'utilisation d'opérateurs Booléens permet de construire des requêtes équivalentes. Cependant elle permet dans certains cas une plus grande lisibilité et une écriture des requêtes plus naturelle comme il l'est illustré dans l'exemple suivant :

Exemple 14 (syntaxe générique) :
Pour récupérer l'ensemble des analyses biologiques ayant un résultat normal, il est possible d'utiliser la clause entité :

```
analyse_biologique(borneInf<=valeur<=borneSup)
```

Cette dernière est cependant équivalente à la conjonction des deux clauses entités

```
analyse_biologique(borneInf<=valeur)
```

et

```
analyse_biologique(valeur<=borneSup)
```

6.1.3.5 Opérateurs Booléens

L'utilisation des opérateurs Booléens classiques « `AND` » et « `OR` » ainsi que des parenthèses (« `(` » et « `)` ») au sein de la sous-clause de contraintes d'une clause entité est possible. Ils

permettent de lier logiquement plusieurs contraintes que ce soit des contraintes d'attribut ou des contraintes sémantiques comme il le sera expliqué dans la sous-section 6.1.4.

> ⚠ **Remarque 8 (priorité des opérations) :**
> *Les priorités classiques des opérations avec l'aide des opérateurs Booléens et des parenthèses sont évidement respectées. Ainsi les opérations entre parenthèses prévalent sur les opérateurs* `AND` *et* `OR` *et l'opérateur* `AND` *prévaut sur l'opérateur* `OR`*.*

Voici quelques exemples illustrant cette possibilité dans le cadre de l'utilisation de contraintes d'attribut :

> ✎ **Exemple 15 (usage du `AND` et du `OR`) :**
> *La requête :*
>
> ```
> patient(
>   sexe="F" AND date_naissance<=1970-01-01
>   OR sexe="M" AND date_naissance<= 1975-01-01)
> ```
>
> *permet de désigner l'ensemble des patients qui sont :*
>
> — *soit de sexe féminin et nés avant le* 1^{er} *janvier* 1970
>
> — *soit de sexe masculin et nés avant le* 1^{er} *janvier* 1975
>
> .

> ✎ **Exemple 16 (contraintes parenthésées) :**
> *La requête :*
>
> ```
> analyse_biologique(
>   date>=2010-01-01
>   AND (valeur<borneInf OR valeur>borneSup))
> ```
>
> *désigne l'ensemble des analyses biologiques ayant eu lieu après le* 1^{er} *janvier* 2010 *et dont le résultat est « anormal » (i.e. « anormalement bas » ou « anormalement haut »).*

> ⚠ **Remarque 9 (Expressions arithmétiques et opérateurs Booléens) :**
> *La syntaxe générique syntaxe 3 peut simplement s'exprimer à l'aide de l'opérateur Booléen* `AND`*. Par exemple, si l'on reprend l'exemple 14 on a l'équivalence logique :*
>
> `analyse_biologique(borneInf<=valeur<=borneSup)`
> $\Leftrightarrow$
> `analyse_biologique(borneInf<=valeur AND valeur<=borneSup)`

> ⚠ **Remarque 10 (Multi-valuation de valeurs et opérateurs Booléens) :**
> *La multi-valuation de la valeur d'un attribut est équivalente à l'utilisation de l'opérateur Booléen* `OR`*. En effet, si l'on reprend l'exemple 10, on a l'équivalence logique :*
>
> `patient(prenom="Jacques","Jean","Daniel")`
> $\Leftrightarrow$
> `patient(prenom="Jacques" OR prenom = "Jean" OR prenom = "Daniel")`

De même que :

patient(prenom="Jacques","Jean","Daniel" AND nom="Bernoulli")
$\Leftrightarrow$
patient((prenom="Jacques" OR prenom="Jean" OR prenom="Daniel")
AND nom="Bernoulli")

En plus des opérateurs Booléens `OR` et `AND`, le langage de requête permet l'exploitation de l'opérateur Booléen `NOT`. Ce dernier ne peut cependant être utilisé qu'uniquement devant une clause entité et constitue donc davantage un élément de syntaxe d'une clause entité qu'un opérateur Booléen à part entière :

</> Syntaxe 4 (négation d'une clause entité) :
Étant donnée une clause entité `ENTITY(CONSTRAINT)`*, la* ***négation de cette clause entité*** *est l'expression de la forme :*

NOT ENTITY(CONSTRAINT)

La négation `NOT ENTITY(CONSTRAINT)` d'une clause entité désigne et/ou cible l'ensemble des entités de type `ENTITY` qui ne sont pas désignés/ciblés par la clause entité `ENTITY(CONSTRAINT)`.

Ainsi, de manière ensembliste, si l'on note :

— $E_{\texttt{ENTITY}}$ l'ensemble contenant toutes les entités de type `ENTITY` ;

— $E_{\texttt{ENTITY(CONSTRAINT)}}$ l'ensemble des entités de type `ENTITY` désignées par la clause entité `ENTITY(CONSTRAINT)` ;

— $E_{\texttt{NOT ENTITY(CONSTRAINT)}}$ l'ensemble des entités de type `ENTITY` désignées par la négation de la clause entité `ENTITY(CONSTRAINT)`.

alors :

$$\begin{aligned} E_{\texttt{NOT ENTITY(CONSTRAINT)}} &= \complement_{E_{\texttt{ENTITY}}} E_{\texttt{ENTITY(CONSTRAINT)}} \\ &= \overline{E_{\texttt{ENTITY(CONSTRAINT)}}} \\ &= E_{\texttt{ENTITY}} \setminus E_{\texttt{ENTITY(CONSTRAINT)}} \end{aligned}$$

6.1.4 Contraintes sémantiques

Les « contraintes sémantiques » constituent le deuxième type de contraintes qu'il est possible d'utiliser pour former la sous-clause `CONSTRAINT` d'une clause entité `ENTITY(CONSTRAINT)` (cf. syntaxe 1).

Elles offrent la possibilité de contraindre les entités désignées par une clause entité avec des critères portant sur d'autres entités qui lui sont reliées par des relations au sein du graphe de données (i.e. du réseau sémantique).

Si l'on se place dans le contexte de l'EDSS, il peut, par exemple, être intéressant de pouvoir requêter un patient en fonction de contraintes portant sur ses séjours, ses analyses biologiques ou encore ses diagnostics. Les contraintes sémantiques du langage de requête proposés ici, offrent cette possibilité.

Concrètement, une contrainte sémantique n'est autre qu'une clause entité utilisée comme contrainte d'une clause entité mère. Le langage permet donc une **imbrication** des clauses entité selon la syntaxe suivante :

</> Syntaxe 5 :
Une clause entité de la forme

$$ENTITY_2(CONSTRAINT_2)$$

est une ***contrainte sémantique pour une entité*** $ENTITY_1$ *lorsque elle est utilisée au sein de la clause de contraintes* $CONSTRAINT_1$ *d'une clause entité de la forme :*

$$ENTITY_1(CONSTRAINT_1)$$

C'est à dire lorsque elle apparaît au sein d'une syntaxe de la forme :

$$ENTITY_1(\ ... \ ENTITY_2(CONSTRAINT_2) \ ... \ )$$

En pratique, pour qu'une contrainte sémantique ait un sens, il est nécessaire que l'entité désignée par $\texttt{ENTITY}_2$ soit reliée sémantiquement à celle désignée par $\texttt{ENTITY}_1$. Une telle syntaxe désigne alors l'ensemble des objets de type $\texttt{ENTITY}_1$ reliés aux objets de type $\texttt{ENTITY}_2$ par une ou plusieurs relations sémantiques et tel que $\texttt{ENTITY}_2$ vérifie les contraintes définies dans sa clause de contrainte propre (i.e. $\texttt{CONSTRAINT}_2$).

Voici quelques exemples simples de contraintes sémantiques d'une clause entité :

Exemple 17 :
La Figure 6.2 montre que l'entité `patient` *peut être reliée à l'entité* `sejour` *par l'intermédiaire de la relation* `a_sejour`*. Il est alors possible de contraindre l'entité* `patient` *avec une contrainte sémantique portant sur ses séjours en formant par exemple les clauses entité :*

- `patient(sejour(date_entree=2005-05-05))` *qui désigne l'ensemble des patients ayant effectué un séjour le* 5 *mai* 2005*. La clause entité* `sejour(date_entree = 2005-05-05)` *est une contrainte sémantique pour l'entité* `patient` *puisque elle désigne implicitement les séjours liés aux patients par l'intermédiaire de la relation* `a_sejour`*.*
- `patient(sejour(date_sortie-date_entree>10.0))` *qui désigne l'ensemble des patients ayant effectué un séjour de plus de* 10 *jours.*
- `patient(sejour(date_entree>2005-05-05 AND date_sortie-date_entree>10.0)` *qui désigne l'ensemble des patients ayant effectué un séjour de plus de* 10 *jours après le* 5 *mai* 2005*.*

Remarque 11 :
Comme les requêtes de l'exemple 17 le montrent, la syntaxe 5 ne permet pas de spécifier les relations sémantiques à « parcourir » qui est en l'occurrence dans cet exemple la relation `a_sejour`*. Ces dernières sont « implicites ». En réalité, la syntaxe 5 est un cas particulier de la syntaxe générale syntaxe 7 qui permet quant à elle de maîtriser finement les relations à parcourir. Cela reste cependant optionnel ce qui permet un allégement de la syntaxe.*

L'exemple 17 présente trois requêtes permettant de contraindre l'entité `patient` avec l'entité `sejour` qui lui est « directement » liée par l'intermédiaire de la relation `a_sejour`. Il n'est cependant pas indispensable que la relation sémantique reliant une clause entité à sa contrainte sémantique soit directe. Il est par exemple possible de requêter un patient en fonction de ses analyses biologiques sans pour autant passer par l'entité `sejour`. Dans ce cas, le parcours successif des relations `a_sejour` et `contient_analyse` et donc le passage intermédiaire par l'entité `sejour` est là, encore, implicite :

Exemple 18 :
L'entité `patient` *est reliée à l'entité* `analyse_biologique` *par l'intermédiaire de l'entité* `sejour` *et des deux relations* `a_sejour` *et* `contient_analyse`*. Il est donc possible de contraindre l'entité* `patient` *avec une contrainte sémantique portant sur ses analyses biologiques en imbriquant directement l'entité* `analyse_biologique` *sans pour autant faire intervenir l'entité* `sejour` *:*

```
patient(analyse_biologique())
⇔
patient(sejour(analyse_biologique()))
```

Ces deux expressions désignent toutes deux l'ensemble des patients ayant effectué une analyse biologique quelconque. Si de plus la contrainte sémantique `analyse_biologique` *contient une contrainte, celle ci s'applique de telle sorte que :*

```
patient(analyse_biologique(date = 2005-05-05))
```

désigne par exemple l'ensemble des patients ayant fait l'objet d'une analyse biologique le 5 *mai* 2005*.*

6.1.4.1 Utilisation des opérateurs Booléens

Lorsque une clause entité est utilisée comme contrainte sémantique, elle se comporte comme une contrainte à part entière au même titre que les contraintes d'attribut. Il est à ce titre toujours possible d'utiliser les opérateurs Booléens `AND` et `OR` pour lier logiquement des contraintes d'attribut avec des contraintes sémantiques ou plusieurs contraintes sémantiques ensemble. En terme d'expressivité, cette possibilité permet notamment de requêter une entité à l'aide de contraintes portant aussi bien sur l'entité elle même que sur ses entités liées.

Exemple 19 :

```
sejour(
  patient(
    prenom="Jean" AND nom="Bernoulli")
  AND 2005-01-01<date_entree< 2005-02-01)
```

désigne l'ensemble des séjours du patient « Jean Bernoulli » ayant été effectuées au cours des séjours ayant débuté entre le 1^{er} *janvier* 2005 *et le* 1^{er} *février* 2005*. Dans cette clause entité,*

```
patient(prenom="Jean" AND nom="Bernoulli")
```

et

```
2005-01-01<date_entree<2005-02-01
```

sont respectivement une contrainte sémantique et une contrainte d'attribut reliées logiquement par l'opérateur Booléen `AND`*. De même :*

- ```
  patient(
    analyse_biologique(
      date=2005-05-05 AND borneInf<=valeur<=borneSup))
  ```
  désigne l'ensemble des patients ayant effectué une analyse biologique quelconque le 5 *Mai* 2005 *et dont le résultat est normal.*
- ```
  patient(
    analyse_biologique(
      date=2005-05-05 AND (valeur<borneInf OR valeur>borneSup)))
  ```
  désigne l'ensemble des patients ayant effectué une analyse biologique quelconque le 5 *mai* 2005 *et dont le résultat est inférieur ou supérieur à la normale.*

6.1.4.2 Contraintes sémantiques récursives

Une contrainte sémantique est avant tout une clause entité. Elle peut donc à son tour comporter des contraintes que ce soit des contraintes d'attribut ou des contraintes sémantiques. Ainsi, l'imbrication d'une clause entité dans la clause de contrainte d'une clause entité mère peut se faire de manière récursive de telle sorte qu'une syntaxe du type suivant est tout à fait possible :

</> Syntaxe 6 (contraintes sémantiques récursives) :
En tant que contrainte à part entière, toute clause entité $ENTITY_3(CONSTRAINT_3)$ *peut être utilisée comme contrainte sémantique d'une clause entité d'entité* $ENTITY_2$ *même lorsque celle-ci est elle-même contrainte sémantique d'une clause entité d'entité* $ENTITY_1$ *:*

$$ENTITY_1(\ \ldots\ ENTITY_2(\ \ldots\ ENTITY_3(CONSTRAINT_3)\ \ldots\ )\ \ldots\ )$$

Cette récursivité offre au langage de requête une importante expressivité sémantique puisqu'elle permet de requêter des entités en fonction de leurs contextes au sein du réseau sémantique de données dans sa globalité sans se limiter à une relation unique.

Exemple 20 :

—
```
patient(
   analyse_biologique(
      valeur>borneSup
      AND code_exe(
         label="Sodium")))
```
désigne les patients ayant une analyse de « Sodium » supérieur à la normale.

—
```
analyse_biologique(
   date=2015-12-25
   AND patient(
      sexe="M")
   AND code_exe(
      label="Calcium","Sodium"))
```
désigne les analyses biologiques de Sodium et de Calcium effectué le 25 Décembre 2015 par des patients de sexe masculin.

— *La clause entité indentée et colorée suivante :*

```
analyse_biologique(
   code_exe(
      label = "Potassium")
   AND valeur < 3.5
   AND patient(
      sexe="M"
      AND date_naissance<1970-01-01)
   AND sejour(
      date_entree=2015-12-25
      AND unite_medical(
         label = "Cardiologie")))
```

désigne les analyses biologiques de « Potassium » inférieures à 3.5 $mmol/L$ effectuées par des patients masculins nés avant 1970 dans un séjour en « Cardiologie » débutant le 25 décembre 2015. La requête ainsi créée, relie des analyses biologiques aux patients à qui elles appartiennent et aux séjours dans lesquels elles ont été effectuées, le séjour étant lui même relié à l'unité médicale dans laquelle il a été effectué.

Remarque 12 :
Il est à noter que si l'on rattache l'entité `sejour` *à l'entité* `patient` *plutôt qu'à l'analyse biologique la requête prend un sens différent. En effet,*

```
analyse_biologique(
   code_exe(
      label = "Potassium")
   AND valeur < 3.5
   AND patient(
      sexe="M"
      AND date_naissance<1970-01-01
      AND sejour(
         date_entree=2015-12-25
         AND unite_medical(
            label = "Cardiologie"))))
```

désigne alors l'ensemble des analyses biologiques de « Potassium » inférieures à 3.5 *mmol/L effectuées par des patients masculins nés avant* 1970 ***ayant effectué par ailleurs*** *un séjour en « Cardiologie » débutant le* 25 *décembre* 2015.

6.1.4.3 Les options de contraintes sémantiques

Une contrainte sémantique peut en réalité faire l'objet de deux types d'options supplémentaires ce qui fait de la syntaxe 5 un cas particulier de la syntaxe plus générique donnée ci-dessous :

</> Syntaxe 7 :
Dans sa forme la plus générique, une ***contrainte sémantique d'une entité*** $\texttt{ENTITY}_1$ *est une clause entité de la forme :*

$$\texttt{ENTITY}_2\texttt{\{PATH\_OPTS\}[JOINT\_OPTS](CONSTRAINT}_2\texttt{)}$$

utilisée au sein de la clause de contrainte d'une clause entité d'entité $\texttt{ENTITY}_1$ *:*

$$\texttt{ENTITY}_1\texttt{(... ENTITY}_2\texttt{\{PATH\_OPTS\}[JOINT\_OPTS](CONSTRAINT}_2\texttt{) ...)}$$

et dans laquelle :

`{PATH_OPTS}` *est une sous-clause optionnelle correspondant aux* ***options de relations****.*

`[JOINT_OPTS]` *est une sous-clause optionnelle correspondant aux* ***options de jointures****.*

D'un point de vue syntaxique, une contrainte sémantique n'est donc rien d'autre qu'une clause entité à laquelle il est possible de rattacher deux types d'options.

Ces deux types d'options sont décrits ci-après.

6.1.4.3.1 Les options de relations

Considérons le cas de base d'une contrainte sémantique potentiellement munie d'options de relations et de jointures à travers une syntaxe du type :

$$\texttt{ENTITY}_1\texttt{(ENTITY}_2\texttt{\{PATH\_OPTS\}[JOINT\_OPTS](CONSTRAINT}_2\texttt{))}$$

Sans options, cette syntaxe désigne alors simplement l'ensemble des entités $\texttt{ENTITY}_1$ reliées à des entités $\texttt{ENTITY}_2$ vérifiant les contraintes $\texttt{CONSTRAINT}_2$.

Il existe néanmoins potentiellement une multitude de chemins permettant de relier $\texttt{ENTITY}_1$ à $\texttt{ENTITY}_2$ (cf. exemple 22). Les options de relations permettent de spécifier les relations sémantiques à « parcourir » pour relier ces deux entités.

Elles prennent la forme d'un ou de plusieurs « chemins » (i.e. « path » en Anglais) ou de « portions » de ces chemins séparés par des virgules « , ».

Ainsi, une option de relation `PATH_OPTS` possède une syntaxe de la forme suivante :

</> Syntaxe 8 :
*Une **option de relation** `{PATH_OPTS}` d'une contrainte sémantique*

`ENTITY{PATH_OPTS}[JOINT_OPTS](CONSTRAINT)`

est de la forme :

$\texttt{\{PATH}_1\texttt{,PATH}_2\texttt{,...,PATH}_n\texttt{\}}$

où chacun des chemins $\texttt{PATH}_i$ *est de la forme :*

$\texttt{[ENTITY}_w\texttt{]RELATIONSHIP}_a\texttt{[ENTITY}_x\texttt{]RELATIONSHIP}_b\texttt{[ENTITY}_y\texttt{]...[ENTITY}_z\texttt{]}$

et :

- *chaque* $\texttt{ENTITY}_i$ *désigne un libellé d'entité ;*
- *chaque* $\texttt{RELATIONSHIP}_i$ *désigne un libellé de relation.*

Exemple 21 (Exemple de chemins) :
Le chemin désignant la principale liaison sémantique existant entre l'entité `patient` *et l'entité* `sejour` *s'écrit :*

`[patient]a_sejour[sejour]`

Le chemin désignant la principale liaison sémantique existant entre l'entité `patient` *et l'entité* `analyse_biologique` *s'écrit :*

`[patient]a_sejour[sejour]contient_analyse[analyse_biologique]`

Exemple 22 (Exemple d'option de relation) :
Il existe deux chemins principaux reliant l'entité `compte_rendu` *à l'entité* `unite_medicale` *qui sont :*

- `[compte_rendu]est_cr[sejour]dans_UM[unite_medicale]`
- `[compte_rendu]est_cr[acte]dans_UM[unite_medicale]`

La requête

`compte_rendu(unite_medicale(label="Cardiologie"))`

désigne tous les comptes-rendus d'actes et de séjours effectués tous deux dans l'unité médicale de « Cardiologie ». En revanche il est possible d'obtenir uniquement les comptes-rendus de séjours effectués en Cardiologie grâce à la requête :

```
compte_rendu(
  unite_medicale
    {[compte_rendu]est_cr[sejour]dans_um[unite_medicale]}
    (label="Cardiologie"))
```

On peut de même obtenir uniquement les comptes-rendus d'actes effectués en « Cardiologie » :

```
compte_rendu(
  unite_medicale
    {[compte_rendu]est_cr[acte_medicale]dans_um[unite_medicale]}
    (label="Cardiologie"))
```

Les relations entre entités sont considérées comme orientées. Il est néanmoins possible de « parcourir » la relation réciproque ce qui permet, de fait, d'imbriquer une entité $\texttt{ENTITY}_\texttt{b}$ dans une entité $\texttt{ENTITY}_\texttt{a}$ même si la relation existante entre $\texttt{ENTITY}_\texttt{a}$ et $\texttt{ENTITY}_\texttt{b}$ est orientée de $\texttt{ENTITY}_\texttt{b}$ vers $\texttt{ENTITY}_\texttt{a}$. La relation réciproque d'une relation de libellé `relation` et alors notée `relation_INVERSE`.

Exemple 23 :
La Figure 6.2 montre que la relation `a_sejour` *entre l'entité* `patient` *et* `sejour` *est orientée de l'entité* `patient` *vers l'entité* `sejour`*. Le chemin parcouru entre ces deux entités dans la requête* `patient(sejour(...))` *est donc bien le chemin :* `[patient]a_sejour[sejour]`*. En revanche celui parcouru dans la requête* `sejour(patient(...))` *est le chemin*

`[sejour]a_sejour_INVERSE[patient]`

où la relation `a_sejour_INVERSE` *est la relation liant un séjour à un patient uniquement si ce séjour est lié à ce patient par l'intermédiaire de la relation* `a_sejour`*.*

Remarque 13 (absence d'options de relations et gestion des chemins) :
La possibilité qu'offre le langage de requête d'omettre l'option de relation implique un « choix » en ce qui concerne les chemins à parcourir entre une entité mère à sa contrainte sémantique lorsque l'option de relation est omise. Même si dans un tel cas, il peut paraître judicieux d'un point de vue purement informatique de parcourir l'ensemble de tous les chemins existants, cela n'est ni pertinent ni possible en pratique. En effet :

Certaines relations ne sont pas pertinentes

Une relation reliant une entité $\texttt{ENTITY}_1$ *à une entité* $\texttt{ENTITY}_2$ *n'est pas nécessairement pertinente en pratique. Si l'on prend l'exemple de la requête* `patient(sejour(...))`*, le chemin liant l'entité* `patient` *à l'entité* `sejour` *que l'on souhaite implicitement parcourir est le chemin*

`[patient]a_sejour[sejour]`

Il existe cependant d'autre chemins beaucoup moins pertinents tels que :

1. ```
   [patient]
     a_sejour[sejour]
       contient_analyse[analyse_biologique]
         contient_analyse_INVERSE[sejour]
   ```
   qui implique que le séjour lié au patient soit relié à au moins une analyse biologique.
2. ```
   [patient]
     a_sejour[sejour]
       dans_UM[unite_medicale]
         dans_um_INVERSE[acte_medical]
           contient_acte_INVERSE[sejour]
   ```
   qui implique que le patient ait eu au moins un acte médical dans la même unité médicale que le séjour.

Il existe une infinité de chemin reliant deux entités

En effet, il existe dans l'absolu une infinité de chemins reliant par exemple les entités `patient` *et* `sejour` *:*

- `[patient]a_sejour[sejour]`
- ```
  [patient]
    a_sejour[sejour]
      a_sejour_INVERSE[patient]
        a_sejour[sejour]
  ```
- ```
  [patient]
    a_sejour[sejour]
      a_sejour_INVERSE[patient]
        a_sejour[sejour]
          a_sejour_INVERSE[patient]
            a_sejour[sejour]
  ```
- ...

Ainsi, dans la pratique, il n'est pas pertinent de générer automatiquement l'ensemble de tous les chemins existants entre deux entités pour les parcourir exhaustivement par la suite. Pour ces raisons, il a été fait le choix lors des implémentations des moteurs de recherche exploitant ce langage de requête de maintenir un liste des chemins « admissibles » pour chaque couple d'entité utile (générée néanmoins automatiquement).

6.1.4.3.2 Les options de jointures

Les options de jointures présentent quant à elles une syntaxe similaire à celle des contraintes d'attribut classiques. Elles suivent une syntaxe du type :

</> Syntaxe 9 :
*Une clause d'**option de jointure** `[JOINT_OPTS]` d'une contrainte sémantique*

$$\texttt{ENTITY\{PATH\_OPTS\}[JOINT\_OPTS](CONSTRAINT)}$$

est de la forme :

$$\texttt{[OPTION}_1\ \texttt{COMPARATOR}_1\ \texttt{VALUE}_1\texttt{, OPTION}_2\ \texttt{COMPARATOR}_2\ \texttt{VALUE}_2\texttt{, ...]}$$

où :

- $\texttt{OPTION}_i$ *est un mot-clé libre identifiant le type de l'option* i*.*
- $\texttt{VALUE}_i$ *est la valeur de l'option* i*. (valeur typée et gérée par le langage de requête (Table 6.1)).*
- $\texttt{COMPARATOR}_i$ *est un des comparateurs géré par le langage de requête (Table 6.1) et spécifiant la contrainte vérifiée par l'option* i *relativement à la valeur* i*.*

Dans la pratique, les options de jointures servent des objectifs variés. D'une manière générale, elles permettent d'effectuer des actions supplémentaires ou spécifiques lors du processus de jointure entre l'entité de la contrainte sémantique et l'entité mère. Un exemple est donné ci-dessous :

Exemple 24 :
Les séjours sont codés avec des concepts de la CIM–10. Ces derniers sont organisés au sein d'une hiérarchie qu'il est utile de pouvoir parcourir automatiquement. Par exemple, les descendants du code « tumeur maligne du sein » [C50 (CIM–10)] dans la hiérarchie de la CIM–10 sont :

— *« mamelon et aréole »* [*C*50.0 (*CIM–10*)].

— *« partie centrale du sein »* [*C*50.1 (*CIM–10*)].

— *« quadrant supéro-interne du sein »* [*C*50.2 (*CIM–10*)].

— ...

— *« sein, sans précision »* [*C*50.9 (*CIM–10*)].

*Il peut être intéressant de pouvoir désigner automatiquement l'ensemble des séjours codés avec le code C*50 *ainsi que tous les descendants (C*50.0*, C*50.1*, ..., C*50.9*) sans pour autant avoir à les inclure tous dans la requête. Pour ces besoins, une option d'« expansion hiérarchique » a été mise en place dans les différentes implémentations du langage de requête. Ainsi, la requête suivante désigne l'ensemble des séjours codés avec C*50 *ou l'un de ses descendants hiérarchiques :*

```
sejour(cim10Category[EXPL = "DOWN"](label = "C50 tumeur maligne du sein"))
```

⚠ Remarque 14 :
La grammaire et le « parser » (section 6.4) du langage de requête impose une syntaxe à la sous-clause `OPTION` *d'une option de jointure. En revanche, ces derniers n'établissent aucune liste de mots clés admissibles. En l'occurrence ici le fait que* `EXPL` *est une option valide alors que* `EXPLOSION` *ne l'est pas est géré dans l'implémentation des moteurs de recherche exploitant le langage de requête.*

6.1.5 Propriétés algébriques du langage

Dans cette section les différentes propriétés algébriques du langage $\mathcal{L}_{\text{}}$ seront données. Afin de les énoncer clairement ainsi que d'en permettre leurs justifications, un sens ensembliste sera donné aux éléments de synaxe de base du langage $\mathcal{L}_{\text{}}$.

Pour plus de lisibilité, une police de caractère différente sera utilisée pour distinguer les éléments de syntaxe du langage (e.g. `Syntaxe`) de ce qui relève de l'interprétation ensembliste du langage (e.g. « *Ensemble* »). De plus les notations suivantes seront employées :

✲ Notation 5 (Notations ensemblistes) :

— *la distinction sera faite entre les objets et leurs types. Ainsi pour une entité notée* `E`*, on notera E l'ensemble des objets de type* `E`*, l'ensemble E étant ainsi constitué des* ***objets*** *qui* ***instancient*** *le type abstrait* `E`*;*

— *pour tout objet a de type* `A` *et tout objet b de type* `B` *(i.e. pour tout* $a \in A$ *et* $b \in B$*), on notera* $a\mathcal{R}b$ [a] *lorsque qu'il existe une relation reliant a à b.*

a. Cette notation est empruntée de la notion mathématique de **relation binaire** sur un ensemble. L'ensemble des relations sémantiques existantes entre les différents objets peut en effet être modélisée mathématiquement par une relation binaire sur l'ensemble contenant la totalité des objets contenues dans le SI (i.e. l'union des objets instanciant l'ensemble des entités)

✲ Notation 6 (Notations syntaxiques) :
Le langage $\mathcal{L}_{\text{}}$ *propose différents types de contraintes. Une contrainte correspond ici à toute expression qui peut être utilisée comme clause de contrainte d'une clause entité. Il peut alors s'agir d'une contrainte d'attribut d'une clause entité utilisée comme contrainte sémantique d'une clause entité mère ou bien, d'une expression Booléenne formée à partir de ces deux dernières. Pour éviter toute ambiguïté, on se tiendra aux notations suivantes :*

— *on désignera une contrainte quelconque (i.e. sans distinction de type) à l'aide d'un « C » majuscule éventuellement indicé (e.g. C_1, C_2, etc.). Une contrainte C pourra alors désigner indifféremment une contrainte d'attribut, une contrainte sémantique ou une expression Booléenne ;*

— *une contrainte d'attribut sera quant à elle spécifiquement notée à l'aide d'un « c » minuscule ;*

— *enfin, une contrainte sémantique sera explicitée avec l'entité à laquelle elle se rapporte. Elle sera donc notée comme une clause d'entité E(C) où C désigne la contrainte appliquée à cette contrainte sémantique.*

6.1.5.1 Propriétés algébriques des opérateurs Booléens

Les opérateurs Booléens OR et AND du langage $\mathcal{L}_{\leq}$ permettent de constituer de nouvelles contraintes à partir de contraintes existantes ou éventuellement d'agréger des clauses entité à la racine d'une **requête** écrite dans le langage $\mathcal{L}_{\leq}$ ($\mathcal{L}_{\leq}$–requête). Ces derniers possèdent toutes les propriétés « classiques » des opérateurs Booléens habituelles de la RI. On rappelle ci-dessous ces propriétés :

Propriété 2 (Propriétés algébriques des opérateurs Booléens) :
Étant données une entité E et trois contraintes quelconques C_1, C_2 et C_3, alors les propriétés suivantes sont vérifiées :

— ***Idempotence** de toute contrainte pour OR :*

$$E(C_1 \text{ OR } C_1) = E(C_1) \tag{6.1}$$

— ***Idempotence** de toute contrainte pour AND :*

$$E(C_1 \text{ AND } C_1) = E(C_1) \tag{6.2}$$

— ***Associativité** de OR :*

$$E((C_1 \text{ OR } C_2) \text{ OR } C_3) = E(C_1 \text{ OR } (C_2 \text{ OR } C_3)) \tag{6.3}$$

— ***Associativité** de AND :*

$$E((C_1 \text{ AND } C_2) \text{ AND } C_3) = E(C_1 \text{ AND } (C_2 \text{ AND } C_3)) \tag{6.4}$$

— ***Commutativité** de OR :*

$$E(C_1 \text{ OR } C_2) = E(C_2 \text{ OR } C_1) \tag{6.5}$$

— ***Commutativité** de AND :*

$$E(C_1 \text{ AND } C_2) = E(C_2 \text{ AND } C_1) \tag{6.6}$$

— ***Distributivité** de OR par rapport à AND :*

$$E(C_1 \text{ OR } (C_2 \text{ AND } C_3)) = E((C_1 \text{ OR } C_2) \text{ AND } (C_1 \text{ OR } C_3)) \tag{6.7}$$

— ***Distributivité** de AND par rapport à OR :*

$$E(C_1 \text{ AND } (C_2 \text{ OR } C_3)) = E((C_1 \text{ AND } C_2) \text{ OR } (C_1 \text{ AND } C_3)) \tag{6.8}$$

Les contraintes C_1, C_2 et C_3 de la propriété 2 peuvent indifféremment et indépendamment désigner des contraintes d'attribut ou des clauses entité (en tant que contraintes sémantiques) comme illustré dans l'exemple suivant :

Exemple 25 :

$$\texttt{patient}(\overbrace{\texttt{sexe="F"}}^{c_1} \texttt{ AND } (\overbrace{\texttt{analyse\_biologique(date>2017-01-01)}}^{C_2} \texttt{ OR } \underbrace{\texttt{acte\_medicale(date>2017-01-01)}}_{C_3}))$$

$$\Downarrow$$

$$\texttt{patient}(c_1 \texttt{ AND } (C_2 \texttt{ OR } C_3))$$

$$\Downarrow$$

$$\texttt{patient}(c_1 \texttt{ AND } C_2 \texttt{ OR } c_1 \texttt{ AND } C_3)$$

$$\Downarrow$$

$$\texttt{patient}(\overbrace{\texttt{sexe="F"}}^{c_1} \texttt{ AND } \overbrace{\texttt{analyse\_biologique(date>2017-01-01)}}^{C_2} \texttt{ OR } \underbrace{\texttt{sexe="F"}}_{c_1} \texttt{ AND } \underbrace{\texttt{acte\_medicale(date>2017-01-01)}}_{C_3})$$

Toutes les propriétés énoncées dans la propriété 2 restent valables lorsque les opérateurs Booléens sont utilisés entre des clauses entité se trouvant à la racine d'une $\mathcal{L}_{\text{}}$–requête. Ces opérateurs n'étant ainsi pas utilisés au sein d'une clause de contrainte d'une entité.

6.1.5.2 Propriétés algébriques des clauses entité

Les clauses entité présentent également certaines propriétés algébriques. Il est cependant nécessaire de donner au préalable un sens ensembliste à la notion de clause entité avant de pouvoir les définir.

Il ne s'agit cependant pas de modéliser intégralement le langage $\mathcal{L}_{\text{}}$ en terme d'ensemble mais de donner une modélisation de base permettant de comprendre les propriétés algébriques de ce dernier au niveau des expressions atomiques du langage.

Les moteurs de recherche SSE_{NoSQL} et SSE_{SQL} constituent, par ailleurs, une implémentation de la logique ensembliste décrite dans cette partie.

Dans un premier temps on définit l'ensemble E des n entités gérées par le SI avec $n \in \mathbb{N}^*$. Chaque entité est notée $\texttt{E}_i$ pour $i \in [\![1\,;\ n]\!]$. On a alors :

$$E = \{\texttt{E}_1,\ \texttt{E}_2, \cdots,\ \texttt{E}_n\} \tag{6.9}$$

Atomiquement, une clause entité peut se présenter sous différentes formes suivant que sa clause de contraintes est une contrainte d'attribut ou une contrainte sémantique. On distingue donc trois cas :

Cas n° 1 : Étant donnée une entité $\texttt{E}_\texttt{i} \in E$, l'expression $\texttt{E}_\texttt{i}\texttt{()}$ désigne l'ensemble des entités de type $\texttt{E}_\texttt{i}$.

On a ainsi :

$$\texttt{E}_\texttt{i}\texttt{()} \equiv E_i \tag{6.10}$$

Remarque 15 :
Pour tout $\texttt{E}_\texttt{i} \in E$ *:*

$$\texttt{NOT } \texttt{E}_\texttt{i}\texttt{()} \equiv E_i \setminus \texttt{E}_\texttt{i}\texttt{()} = \varnothing \tag{6.11}$$

Cas n° 2 : Étant données une entité $\mathtt{E_i} \in E$ et une contrainte d'attribut c, l'expression $\mathtt{E_i(c)}$ désigne une partie de l'ensemble E_i (i.e. $\mathtt{E_i(c)} \in \mathcal{P}(E_i)$). Cette partie est composée des objets de type $\mathtt{E_i}$ qui vérifient la contrainte c.

Autrement dit :

$$\mathtt{E_i(c)} = \{e \in E_i;\ e \text{ vérifie la contrainte c}\} \tag{6.12}$$

On peut alors définir la négation de l'expression $\mathtt{E_i(c)}$:

$$\mathtt{NOT\ E_i(c)} \equiv E_i \setminus \mathtt{E_i(c)} \tag{6.13}$$

Cas n° 3 : De même, on définit de manière générique pour tout entité $\mathtt{E_i} \in E$ et toute partie $\mathtt{P_j} \in \mathcal{P}(E_j)$ de l'ensemble des objets d'une entité $\mathtt{E_j} \in E$ quelconque, l'expression $\mathtt{E_i(P_j)}$ comme l'ensemble consitué des éléments de E_i en relation avec au moins un élément de $\mathtt{P_j}$.

En d'autres termes :

$$\mathtt{E_i(P_j)} = \{e \in E_i;\ \exists p \in \mathtt{P_j},\ e\mathcal{R}p\} \tag{6.14}$$

On peut alors définir la négation de l'expression $\mathtt{E_i(P_j)}$:

$$\mathtt{NOT\ E_i(P_j)} \equiv E_i \setminus \mathtt{E_i(P_j)} \tag{6.15}$$

Ceci permet, entre autres, de donner un sens aux expressions incluant une contrainte sémantique. Ainsi pour toutes entités $(\mathtt{E_i}, \mathtt{E_j}) \in E^2$, et toute contrainte d'attribut $\mathtt{c}$ on a :

$$\mathtt{E_i(E_j())} = \{e \in E_i;\ \exists x \in \mathtt{E_j()},\ e\mathcal{R}x\} = \{e \in E_i;\ \exists x \in E_j,\ e\mathcal{R}x\} \tag{6.16}$$

$$\mathtt{E_i(E_j(c))} = \{e \in E_i;\ \exists x \in \mathtt{E_j(c)},\ e\mathcal{R}x\} = \{e \in E_i;\ \exists x \in E_j,\ e\mathcal{R}x \text{ et } x \text{ vérifie c}\} \tag{6.17}$$

À l'aide de ces définitions ensemblistes atomiques d'une clause entité, on peut alors définir le sens ensembliste des opérateurs Booléens `AND` et `OR`. Ainsi, pour toute entité $(\mathtt{E_i}, \mathtt{E_j}, \mathtt{E_k}) \in E^3$, tous sous-ensembles $\mathtt{P_j} \in \mathcal{P}(E_j)$ et $\mathtt{P_k} \in E_k$ d'objets de type $\mathtt{E_j}$ et $\mathtt{E_k}$ et toutes contraintes d'attribut $\mathtt{c_1}$ et $\mathtt{c_2}$, on définit :

$$\mathtt{E_i(c_1\ OR\ c_2)} \equiv \mathtt{E_i(c_1)} \cup \mathtt{E_i(c_2)} = \{e \in E_i;\ e \text{ vérifie } \mathtt{c_1} \text{ ou } \mathtt{c_2}\} \tag{6.18}$$

$$\mathtt{E_i(P_j\ OR\ c_1)} \equiv \mathtt{E_i(P_j)} \cup \mathtt{E_i(c_1)} = \{e \in E_i;\ (\exists x \in \mathtt{P_j}, e\mathcal{R}x) \text{ ou } (e \text{ vérifie } \mathtt{c_1})\} \tag{6.19}$$

$$\mathtt{E_i(P_j\ OR\ P_k)} \equiv \mathtt{E_i(P_j)} \cup \mathtt{E_i(P_k)} = \{e \in E_i;\ (\exists x \in \mathtt{P_j}, e\mathcal{R}x) \text{ ou } (\exists y \in \mathtt{P_k},\ e\mathcal{R}y)\} \tag{6.20}$$

$$\mathtt{E_i(c_1\ AND\ c_2)} \equiv \mathtt{E_i(c_1)} \cap \mathtt{E_i(c_2)} = \{e \in E_i;\ e \text{ vérifie } \mathtt{c_1} \text{ et } \mathtt{c_2}\} \tag{6.21}$$

$$\mathtt{E_i(P_j\ AND\ c_1)} \equiv \mathtt{E_i(P_j)} \cap \mathtt{E_i(c_1)} = \{e \in E_i;\ (\exists x \in \mathtt{P_j}, e\mathcal{R}x) \text{ et } (e \text{ vérifie } \mathtt{c_1})\} \tag{6.22}$$

$$\mathtt{E_i(P_j\ AND\ P_k)} \equiv \mathtt{E_i(P_j)} \cap \mathtt{E_i(P_k)} = \{e \in E_i;\ (\exists x \in \mathtt{P_j}, e\mathcal{R}x) \text{ et } (\exists y \in \mathtt{P_k},\ e\mathcal{R}y)\} \tag{6.23}$$

De même on donne un sens aux opérateurs Booléens lorsqu'ils sont utilisés entre clauses entité :

$$\mathtt{E_i(c_1)\ OR\ E_j(c_2)} \equiv \mathtt{E_i(c_1)} \cup \mathtt{E_j(c_2)} \tag{6.24}$$

$$\mathtt{E_i(P_k)\ OR\ E_j(c_1)} \equiv \mathtt{E_i(P_k)} \cup \mathtt{E_j(c_1)} \tag{6.25}$$

$$\mathtt{E_i(P_k)\ OR\ E_j(P_1)} \equiv \mathtt{E_i(P_k)} \cup \mathtt{E_j(P_1)} \tag{6.26}$$

$$\mathtt{E_i(c_1)\ AND\ E_j(c_2)} \equiv \mathtt{E_i(c_1)} \cap \mathtt{E_j(c_2)} \tag{6.27}$$

$$\mathtt{E_i(P_k)\ AND\ E_j(c_1)} \equiv \mathtt{E_i(P_k)} \cap \mathtt{E_j(c_1)} \tag{6.28}$$

$$\mathtt{E_i(P_k)\ AND\ E_j(P_1)} \equiv \mathtt{E_i(P_k)} \cap \mathtt{E_j(P_1)} \tag{6.29}$$

On en déduit alors les propriétés suivantes :

Propriété 3 (propriétés algébriques de la clause entité) :

Étant donnés :

- $E_i \in E$, $E_j \in E$, $E_k \in E$, $E_l \in E$ *quatre entités non nécessairement distinctes ;*
- c_1 *et* c_2 *deux contraintes d'attribut quelconques ;*
- $P_i \in \mathcal{P}(E_i)$, $P_j \in \mathcal{P}(E_j)$, $P_k \in \mathcal{P}(E_k)$ *et* $P_l \in \mathcal{P}(E_l)$ *quatre parties de respectivement* E_i, E_j, E_k *et* E_l *;*

Alors les propriétés algébriques suivantes sont vérifiées :

1. ***Distributivité*** *de la clause entité par rapport à* `OR` *:*

$$E_i(c_1 \texttt{ OR } c_2) \equiv E_i(c_1) \texttt{ OR } E_i(c_2) \tag{6.30}$$
$$E_i(P_j \texttt{ OR } c_1) \equiv E_i(P_j) \texttt{ OR } E_i(c_1) \tag{6.31}$$
$$E_i(P_j \texttt{ OR } P_k) \equiv E_i(P_j) \texttt{ OR } E_i(P_k) \tag{6.32}$$
$$E_i(E_j(c_1 \texttt{ OR } c_2)) \equiv E_i(E_j(c_1) \texttt{ OR } E_j(c_2)) \tag{6.33}$$
$$E_i(E_j(P_k \texttt{ OR } c_1)) \equiv E_i(E_j(P_k) \texttt{ OR } E_j(c_1)) \tag{6.34}$$
$$E_i(E_j(P_k \texttt{ OR } P_1)) \equiv E_i(E_j(P_k) \texttt{ OR } E_j(P_1)) \tag{6.35}$$

2. ***Distributivité partielle*** *de la clause entité par rapport à* `AND` *:*

$$E_i(c_1 \texttt{ AND } c_2) \equiv E_i(c_1) \texttt{ AND } E_i(c_2) \tag{6.36}$$
$$E_i(P_j \texttt{ AND } c_1) \equiv E_i(P_j) \texttt{ AND } E_i(c_1) \tag{6.37}$$
$$E_i(P_j \texttt{ AND } P_k) \equiv E_i(P_j) \texttt{ AND } E_i(P_k) \tag{6.38}$$
$$E_i(E_j(c_1 \texttt{ AND } c_2)) \subseteq E_i(E_j(c_1) \texttt{ AND } E_j(c_2)) \tag{6.39}$$
$$E_i(E_j(P_k \texttt{ AND } c_1)) \subseteq E_i(E_j(P_k) \texttt{ AND } E_j(c_1)) \tag{6.40}$$
$$E_i(E_j(P_k \texttt{ AND } P_1)) \subseteq E_i(E_j(P_k) \texttt{ AND } E_j(P_1)) \tag{6.41}$$

3. ***Première loi de Morgan*** *pour les clauses entité :*

$$\texttt{NOT } E_i(c_1 \texttt{ OR } c_2) \equiv \texttt{NOT } E_i(c_1) \texttt{ AND NOT } E_i(c_2) \tag{6.42}$$
$$\texttt{NOT } E_i(P_j \texttt{ OR } c_1) \equiv \texttt{NOT } E_i(P_j) \texttt{ AND NOT } E_i(c_1) \tag{6.43}$$
$$\texttt{NOT } E_i(P_j \texttt{ OR } P_k) \equiv \texttt{NOT } E_i(P_j) \texttt{ AND NOT } E_i(P_k) \tag{6.44}$$
$$\texttt{NOT } E_i(E_j(c_1 \texttt{ OR } c_2)) \equiv \texttt{NOT } E_i(E_j(c_1)) \texttt{ AND NOT } E_i(E_j(c_2)) \tag{6.45}$$
$$\texttt{NOT } E_i(E_j(P_k \texttt{ OR } c_1)) \equiv \texttt{NOT } E_i(E_j(P_k)) \texttt{ AND NOT } E_i(E_j(c_1)) \tag{6.46}$$
$$\texttt{NOT } E_i(E_j(P_k \texttt{ OR } P_1)) \equiv \texttt{NOT } E_i(E_j(P_k)) \texttt{ AND NOT } E_i(E_j(P_1)) \tag{6.47}$$

4. ***Deuxième loi de Morgan partielle*** *pour les clauses entité :*

$$\texttt{NOT } E_i(c_1 \texttt{ AND } c_2) \equiv \texttt{NOT } E_i(c_1) \texttt{ OR NOT } E_i(c_2) \tag{6.48}$$
$$\texttt{NOT } E_i(P_j \texttt{ AND } c_1) \equiv \texttt{NOT } E_i(P_j) \texttt{ OR NOT } E_i(c_1) \tag{6.49}$$
$$\texttt{NOT } E_i(P_j \texttt{ AND } P_k) \equiv \texttt{NOT } E_i(P_j) \texttt{ OR NOT } E_i(P_k) \tag{6.50}$$
$$\texttt{NOT } E_i(E_j(c_1 \texttt{ AND } c_2)) \supseteq \texttt{NOT } E_i(E_j(c_1)) \texttt{ OR NOT } E_i(E_j(c_2)) \tag{6.51}$$
$$\texttt{NOT } E_i(E_j(P_k \texttt{ AND } c_1)) \supseteq \texttt{NOT } E_i(E_j(P_k)) \texttt{ OR NOT } E_i(E_j(c_1)) \tag{6.52}$$
$$\texttt{NOT } E_i(E_j(P_k \texttt{ AND } P_1)) \supseteq \texttt{NOT } E_i(E_j(P_k)) \texttt{ OR NOT } E_i(E_j(P_1)) \tag{6.53}$$

L'exemple 26 ci-après donne un exemple d'utilisation des ces propriétés algébrique de la clause entité. Il permet notamment de mettre en évidence l'impacte que ces dernières peuvent avoir sur la signification intrinsèque des requêtes et plus spécifiquement en ce qui concerne le

« caractère partiel » des propriétés vis à vis de l'opérateur `AND`.

Exemple 26 :

$$\overbrace{\texttt{patient}}^{P}(\overbrace{\texttt{sexe="F"}}^{c_1}\ \texttt{AND}\ \overbrace{\texttt{analyse\_biologique}}^{A}(\overbrace{\texttt{valeur>x}}^{c_2}\ \texttt{OR}\ \overbrace{\texttt{valeur<y}}^{c_3}))$$

$$\equiv$$

$$P(c_1\ \texttt{AND}\ A(c_2\ \texttt{OR}\ c_3))$$

$$\equiv$$

$$P(c_1\ \texttt{AND}\ A(c_2)\ \texttt{OR}\ c_1\ \texttt{AND}\ A(c_3))$$

$$\equiv$$

$$P(c_1\ \texttt{AND}\ A(c_2))\ \texttt{OR}\ P(c_1\ \texttt{AND}\ A(c_3))$$

$$\equiv$$

$$P(c_1)\ \texttt{AND}\ P(A(c_2))\ \texttt{OR}\ P(c_1)\ \texttt{AND}\ P(A(c_3))$$

$$\equiv$$

```
patient(sexe="F") AND patient(analyse_biologique(valeur>x))
OR patient(sexe="F") AND patient(analyse_biologique(valeur<y))
```

En revanche il n'y a pas d'équivalence si l'opérateur Booléen `OR` *est remplacé par l'opérateur* `AND` *:*

```
patient(sexe="F" AND analyse_biologique(valeur>x AND valeur<y))
```

$$\equiv$$

```
patient(sexe="F") AND patient(analyse_biologique(valeur>x AND valeur<y))
```

$$\subset$$

```
patient(sexe="F") AND patient(analyse_biologique(valeur>x))
        AND patient(analyse_biologique(valeur<y))
```

6.2 Rappels sur les langages formels

Le langage de requête dont la syntaxe à été décrite dans la section précédente (section 6.1) est, dans la pratique, basée sur une **grammaire formelle** dont j'ai réalisé la construction dans le cadre de ma thèse.

Une grammaire formelle est assimilable à un ensemble de règles permettant, par « application » récursive de ces dernières, de construire un ensemble de **mots** qui constitue un **langage formel**.

Cette section rappelle les fondements mathématiques permettant d'aboutir à la notion et à l'objet mathématique de **grammaire formelle**. Ces derniers sont indispensables à la bonne compréhension des règles de grammaire constituant le langage de requête $\mathcal{L}_{\text{<}}$ exploité par le SSE_{SQL} et le SSE_{NoSQL}. De plus, bien que la grammaire formelle de ce langage puisse paraître être une élément purement formel, il constitue en réalité un composant informatique essentiel qu'il m'a été nécessaire d'implémenter dans le cadre du développement des moteurs de recherche SSE_{SQL} et SSE_{NoSQL}.

Afin de ne pas surcharger ce mémoire, cette section se présente davantage sous la forme d'une synthèse et ne donne que succinctement les définitions mathématique utiles à la grammaire formelle du langage $\mathcal{L}_{\text{<}}$. Une reconstitution formelle de ces notions mathématiques bien plus détaillée et accompagnée de nombreux exemples est néanmoins donnée dans l'Annexe D.

Dans cette section, les notations suivantes seront employées :

Notation 7 :

— *On notera pour tout entier naturel* $n \in \mathbb{N}$ *:*

$$I_n = \begin{cases} \varnothing & si\ n = 0 \\ [\![1\,;\ n]\!] & si\ n > 0 \end{cases}$$

où $[\![1\,;\ n]\!]$ *désigne l'ensemble des entiers naturels compris entre* 1 *et* n *(viz.*$[\![1\,;\ n]\!] = \{i \in \mathbb{N} : 1 \leqslant i \leqslant n\}$*).*

— *Pout tout ensemble* E *et tout ensemble* F *on notera* $\mathcal{A}(E, F)$ *l'ensemble des applications de* E *dans* F*.*

— *Pour tout ensemble* E *et tout entier naturel* $n \in \mathbb{N}$ *on notera* $E_n = \mathcal{A}(I_n, E)$ *l'ensemble des applications de* I_n *dans* E*.*

— *Pour tout ensemble* E *on notera* $\mathcal{P}(E)$ *l'ensemble des parties de* E*.*

— *Pour tout ensemble fini* E*, on notera* $card(E)$ *le cardinal de l'ensemble* E*.*

— *Pour tout ensemble* E *et tout ensemble* F *on notera* $f : E \longrightarrow F$ *tout application* f *de* E *dans* F*.*

— *Pour toute fonction* f *d'un ensemble* E *dans un ensemble* F *on notera :*

  — $dom(f) = E$ *l'ensemble de départ (ou domaine) de l'application* f*.*

  — $codom(f) = F$ *l'ensemble d'arrivé (ou le codomaine) de l'application* f*.*

6.2.1 Alphabet et mots

Définition 13 (Alphabet) :
On appelle ***Alphabet*** *tout ensemble* A *fini et non vide.*
Les éléments de A *sont alors appelés* ***symboles*** *ou encore* ***lettres****.*

Un alphabet est donc un ensemble fini contenant les éléments syntaxiques de base à partir desquels les expressions d'un langage sont construites. Un mot sur un alphabet est, quant à lui, formellement vu comme une association entres des positions et des symboles de cet alphabet. En d'autres termes, un mot sur un alphabet n'est autre qu'un n-uplet d'éléments de l'alphabet ou encore une famille d'éléments d'un alphabet indexé sur I_n.

Définition 14 (Mot sur un alphabet) :
Étant donné un alphabet A, on appelle ***mot sur*** *A tout n-uplet d'éléments de A où n est un entier naturel. On appelle donc* ***mot sur*** *A tout application w de I_n dans A où $n \in \mathbb{N}$:*

$$w : \left\{ \begin{array}{ccc} I_n & \longrightarrow & A \\ i & \longrightarrow & w_i \end{array} \right.$$

Remarque 1 :

- *Pour tout alphabet A, il existe un unique 0-uplet d'éléments de A appelé le* ***mot vide*** *et noté ε_A. Le mot vide sur A correspond alors à l'unique application vide $\varepsilon_A : \varnothing \longrightarrow A$.*
- *Pour tout alphabet A, l'ensemble $A_n = \mathcal{A}(I_n, A)$ des applications de I_n dans A est l'ensemble des mots sur A composés de n symboles de A.*

Bien que la notion de mot sur un alphabet A soit défini à l'aide de la notion d'application, on utilisera dans la pratique des notations simples pour désigner un mot. En fonction de la confusion que peut engendrer la notation, un mot w sur un alphabet A composé de n symboles pourra être noté au choix :

$$\begin{array}{lccl} & (w_1, w_2, \ldots, w_n) & \text{avec} & \forall i \in [\![1\,;\,n]\!], w_i = w(i) \\ \text{ou bien :} & w_1\ w_2\ \ldots\ w_n & \text{avec} & \forall i \in [\![1\,;\,n]\!], w_i = w(i) \\ \text{ou encore :} & w_1 w_2 \ldots w_n & \text{avec} & \forall i \in [\![1\,;\,n]\!], w_i = w(i) \end{array}$$

Un mot w sur un alphabet A est un n-uplet d'éléments de A où n est un entier naturel. n correspond intuitivement au nombre de symboles qui composent le mot w et donc à sa longueur noté $|w|$.

La notion d'alphabet, de mot sur un alphabet et de longueur d'un mot étant défini, il est désormais possible de définir l'ensemble A^* des mots sur un alphabet A. Les ensembles A_0, A_1, A_2, etc.correspondent respectivement aux ensembles des mots de longueurs 0, 1, 2, etc. Ainsi, l'ensemble de tous les mots n'est rien d'autre que la réunion de tous ces ensembles.

Définition 15 (Ensemble des mots sur un alphabet) :
Étant donné un alphabet A on note A^* l'ensemble des mots sur A défini par :

$$A^* = \bigcup_{n \in \mathbb{N}} A_n$$

La notion de langage sur un alphabet ne requiert en elle même aucun élément théorique supplémentaire pour être définie. Cependant, la notion de grammaire fait, elle, appel à certaines opérations sur les langages qui nécessitent l'introduction de l'opération de « concaténation » sur les mots d'un alphabet.

On s'attache desormais à définir la loi de composition interne de concaténation définie sur l'ensemble des mots d'un alphabet quelconque. Cette loi permet la construction de nouveaux mots à partir de mots initiaux et par simple concaténation des symboles de ces derniers.

Définition 16 (Concaténation) :
Soit A un alphabet. On note $\bowtie_A$: $A^* \times A^* \longrightarrow A^*$ la loi de composition interne de concaténation sur A^* qui à tout couple $(x, y) \in A^* \times A^*$ de mots sur A associe le mot $x \bowtie_A y$ défini comme l'unique mot de $A_{|x|+|y|}$ tel que que :

$$x \bowtie_A y : \begin{cases} I_{|x|+|y|} & \longrightarrow & A \\ i & \longrightarrow & \begin{cases} x(i) & \text{si } i \leqslant |x| \\ y(i-|x|) & \text{sinon} \end{cases} \end{cases}$$

De manière plus simple, si x et y s'écrivent respectivement $x = x_1 \dots x_n$ et $y = y_1 \dots y_m$ avec n et m deux entiers naturels alors $x \bowtie_A y$ est le mot sur A dont les symboles sont ceux de x concaténés à ceux de y :

$$x_1 \dots x_n \bowtie_A y_1 \dots y_m = x_1 \dots x_n y_1 \dots y_m$$

$$\begin{array}{cccccccccc} x \bowtie_A y & = & x_1 & x_2 & \cdots & x_n & y_1 & y_2 & \cdots & y_m \\ & & \uparrow & \uparrow & & \uparrow & \uparrow & \uparrow & & \uparrow \\ x \bowtie_A y(i) & = & x(1) & x(2) & \cdots & x(n) & y(n+1-n) & y(n+2-n) & \cdots & y(n+m-n) \\ & & \uparrow & \uparrow & & \uparrow & \uparrow & \uparrow & & \uparrow \\ i & = & 1 & 2 & \cdots & n & n+1 & n+2 & \cdots & n+m \\ & & \underbrace{\qquad\qquad\qquad\qquad}_{\text{si } i \leqslant n} & & & & \underbrace{\qquad\qquad\qquad\qquad}_{\text{sinon}} & & & \end{array}$$

Remarque 2 :
La loi de composition interne de concaténation est compatible avec la concaténation de deux mots x et y sur deux alphabets distincts A_1 et A_2. x et de y sont alors vus comme deux mots de l'alphabet $A_1 \cup A_2$ et leur concaténation comme le mot $x \bowtie_{A_1 \cup A_2} y$ sur ce même alphabet. Par la suite on n'indicera donc plus la loi de composition interne de concaténation avec l'alphabet sous-jacent et on notera simplement $x \bowtie y$ au lieu de $x \bowtie_{A_1 \cup A_2} y$ (ou de $x \bowtie_A y$ si x et y sont deux mot sur un même alphabet A).

Il est a noter que l'opération de concaténation sur une alphabet A est associative et que le mot vide ε_A est neutre pour celle-ci (cf. propriété 6). Par la suite, on omettra parfois le symbole $\bowtie$ pour désigner la concaténation de deux mots. Ainsi pour deux mots u et v sur un alphabet quelconque on notera simplement uv pour désigner la concaténation $u \bowtie v$.

6.2.2 Langages

Dans la pratique, un langage sur un alphabet consiste simplement en un sous-ensemble de l'ensemble des mots sur cet alphabet. La définition d'un langage permet ainsi d'établir au sein de l'ensemble de tous les mots qu'il est possible de former sur un alphabet quelconque, une distinction entre des mots « valides » qui appartiennent alors à ce langage et d'autres qui ne le sont pas.

La notion de langage ne définit pas de règles de construction de « mot valide » mais s'intéresse simplement à leur regroupement au sein d'un même ensemble.

Définition 17 (Langage) :
On appelle langage sur un alphabet A toute partie $\mathcal{L}$ de l'ensemble A^*.

La définition suivante introduit l'opération de produit pour les langages. De manière analogue à l'opération de concaténation sur les mots d'un alphabet, le produit de langages permet de construire de nouveaux langages à partir de langages existants. Ces deux notions formelles sont particulièrement utiles pour définir les grammaires formelles et plus précisément pour définir les règles de production d'une grammaire.

Définition 18 (Produit de langages) :
On appelle ***produit de langages*** *la relation qui à tout couple de langage* $(\mathcal{L}_1, \mathcal{L}_2) \in \mathcal{P}(A_1) \times \mathcal{P}(A_2)$ *sur des alphabets* A_1 *et* A_2 *associe le langage* $\mathcal{L}_1 \bullet \mathcal{L}_2$ *sur l'alphabet* $A_1 \cup A_2$ *définit par :*

$$\mathcal{L}_1 \bullet \mathcal{L}_2 = \{x \bowtie y \mid (x, y) \in \mathcal{L}_1 \times \mathcal{L}_2\}$$

Remarque 3 :
Le produit $\mathcal{L}_1 \bullet \mathcal{L}_2$ *n'est donc rien d'autre que le langage contenant tous les mots formés de la concaténation d'un mot de* $\mathcal{L}_1$ *avec un mot de* $\mathcal{L}_2$*.*

6.2.3 Grammaire

On définit la notion de grammaire formelle :

Définition 19 (Grammaire) :
Une grammaire est un quadruplet $\mathcal{G} = (V_T, V_N, S, R)$ *tel que :*

- V_T *est un alphabet appelé* ***vocabulaire terminal*** *et dont les symboles sont appelés* ***terminaux****,*
- V_N *est un alphabet disjoint de* V_T ($V_T \cap V_N = \varnothing$) *appelé* ***vocabulaire non-terminal*** *et dont les symboles sont appelés* ***non-terminaux*** *ou* ***variables****,*
- $V = V_T \cup V_N$ *est appelé* ***vocabulaire*** *ou* ***vocabulaire général*** *de la grammaire* $\mathcal{G}$*,*
- $S \in V$ *est un non-terminal particulier appelé* ***source*** *ou* ***axiome****,*
- R *est une partie finie de* $V^* \bullet V_N \bullet V^* \times V^*$ *dont les éléments sont appelés* ***règles de production****. Une règle de production* $(u, v) \in R \subseteq V^* \bullet V_N \bullet V^* \times V^*$ *est alors noté :*

$$u \to v$$

Remarque 4 :
Les notations suivantes sont conventionnellement employée :

- *Les terminaux sont classiquement notés en minuscule.*
- *Les non-terminaux sont classiquement notés en majuscule.*

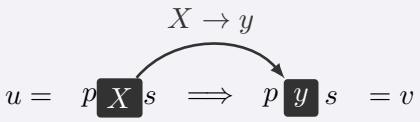
D'un point de vue général, une grammaire peut être vue comme un objet formel permettant de « construire » un langage. Le langage alors reconnu et un langage sur l'ensemble V_T des terminaux et non pas sur l'ensemble V_N des variables qui ne sert que dans le processus interne de construction des mots du langage.

Les « règles de construction » des mots du langage sont données par l'intermédiaire des règles de production (ensemble R). Une règle de production $x \to y$ peut intuitivement être comprise comme la possibilité de remplacer une occurrence du mot x dans un mot quelconque u par le mot y. On parle alors de **dérivation** du mot u.

Par application récursive et aléatoire de ces règles de production à partir de l'axiome S on peut ainsi construire une multitude de mots qui constituent ainsi un langage. On formalise alors la notion de dérivation :

Définition 20 (Dérivation élémentaire) :
*Étant donné une grammaire $\mathcal{G} = (V_T, V_N, S, R)$ et $(u, v) \in (V^*)^2$ (avec $V = V_T \cup V_N$) un couple de mot, on dit que v **dérive directement de** u **par la grammaire** $\mathcal{G}$ et on note alors $u \Rightarrow v$ si et seulement si $u = v$ ou bien si il existe une règle de production $X \to y$ de R et deux mots $(p, s) \in (V^*)^2$ (préfixe et suffixe) tels que :*

- *$u = pXs$ (X est un sous-mot de u)*
- *$v = pys$ (y est un sous-mot de v)*



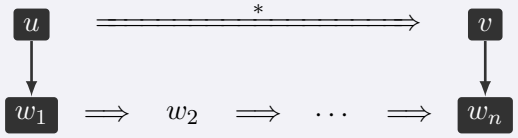
Remarque 5 :
On emploiera également la notation $u \xRightarrow{X \to y} v$ pour signifier que v dérive directement de u par l'intermédiaire de la règle de production $X \to y$.

Par application récursive de dérivations élémentaires il est alors possible de construire de nouveau mots qui dérivent du mot initial.

Ce procédé récursif correspond à la notion de **dérivation** que l'on définit ci-dessous :

Définition 21 (Dérivation) :
*Étant donné une grammaire $\mathcal{G} = (V_T, V_N, S, R)$ et $(u, v) \in (V^*)^2$ (avec $V = V_T \cup V_N$) un couple de mots, on dit que v **dérive de** u **par la grammaire** $\mathcal{G}$ et on note alors $u \overset{*}{\Rightarrow} v$ si et seulement si il existe un entier $n \geqslant 2$ et un n-uplet $w = (w_1, \ldots, w_n) \in \mathcal{A}(I_n, V^*)$ de mots sur V tel que :*

- *$w_1 = u$*
- *$\forall i \in I_{n-1}, w_i \Rightarrow w_{i+1}$*
- *$w_n = v$*



Remarque 6 :
Un mot v dérive d'un mot u si il peut être obtenu par un enchaînement de dérivations élémentaires à partir du mot u.

L'écriture des règles de production d'un grammaire formelle peut s'avérer être une tâche « lourde » ou « redondante ». Un certain nombre de notations existe néanmoins. Ces dernières visent principalement à apporter un peu plus d'expressivité et de flexibilité aux règles de production d'une grammaire formelle même si elles ne sont, sur le plan théorique, pas indispensables.

La remarque 16 suivante introduit certains de ces méta-symboles qui seront par ailleurs utilisés dans la suite de ce mémoire pour décrire la grammaire formelle engendrant le langage $\mathcal{L}_{\text{arbre}}$.

Remarque 16 (Méta-symboles dans les règles de production) :
Pour plus de clarté et de concision dans l'écriture des règles de production d'une grammaire, on emploiera parfois des ***méta-symboles*** *permettant de fusionner certains ensembles de règles de production en une seule. Ces simplifications sont issues du métalangage* ***B****ackus–* ***N****aur* ***F****orm (BNF) et de ses extensions bien que les méta-symboles que l'on emploiera dans ce mémoire ne soient pas syntaxiquement rigoureusement identiques à ceux de cette notation. Ainsi, Étant donnée $\mathcal{G} = (V_T, V_N, S, R)$ une grammaire formelle ou $V = V_T \cup V_N$ est la grammaire générale de $\mathcal{G}$ on pourra notamment employer :*
Le méta-symbole « | » et des parenthèses pour « fusionner » l'ensemble des règles de réécriture possible d'un même mot Y en une seule règle de production :

$$\begin{cases} X \to w_a Y w_b \\ Y \to Y_0 \\ Y \to Y_1 \\ \dots \\ Y \to Y_n \end{cases} \Leftrightarrow \quad X \to w_a(Y_0 \mid Y_1 \mid \cdots \mid Y_n)w_b \quad \text{avec} \quad \begin{cases} (X,Y) \in (V^* \bullet V_N \bullet V^*)^2 \\ n \in \mathbb{N}^* \\ (w_a, w_b) \in (V^*)^3 \\ \forall i \in [\![0\,;\ n]\!], Y_i \in V^* \end{cases}$$

Le méta-symbole « ? » et des parenthèses pour fusionner un ensemble de règles de production rendant intrinsèquement un mot Y optionnel :

$$\begin{cases} X \to w_a Z w_b \\ Z \to Y \\ Z \to \varepsilon \end{cases} \Leftrightarrow \quad X \to w_a\,(Y)?w_b \qquad \text{avec} \quad \begin{cases} (X,Z) \in (V^* \bullet V_N \bullet V^*)^2 \\ (w_a, w_b, Y) \in (V^*)^3 \end{cases}$$

Le méta-symbole « ∗ » et des parenthèses pour fusionner un ensemble de règles de production permettant intrinsèquement à un mot Y d'apparaître un nombre indéfini de fois (éventuellement 0) :

$$\begin{cases} X \to w_a Z w_b \\ Z \to Y Z \\ Z \to \varepsilon \end{cases} \Leftrightarrow \quad X \to w_a\,(Y)^*\,w_b \qquad \text{avec} \quad \begin{cases} (X,Z) \in (V^* \bullet V_N \bullet V^*)^2 \\ (w_a, w_b, Y) \in (V^*)^3 \end{cases}$$

Le méta-symbole « + » et des parenthèses pour fusionner un ensemble de règles de production permettant intrinsèquement à un mot Y d'apparaître au moins une fois puis un nombre indéfini de fois (éventuellement 0) :

$$\begin{cases} X \to w_a Z w_b \\ Z \to Y Z' \\ Z' \to Y \\ Z' \to \varepsilon \end{cases} \Leftrightarrow \quad X \to w_a\,(Y)^+\,w_b \qquad \text{avec} \quad \begin{cases} (X,Z,Z') \in (V^* \bullet V_N \bullet V^*)^3 \\ (w_a, w_b, Y) \in (V^*)^3 \end{cases}$$

Une grammaire permet en somme de construire des mots en dérivant de manière récursive son axiome S. Ces dérivation aboutissent néanmoins à des mots sur le vocabulaire général pouvant contenir des variables appartenant à l'alphabet non terminal et non pas sur le vocabulaire terminal uniquement. Les mots contenant un symbole du vocabulaire non-terminal doivent donc être vus comme des mots intermédiaires dans le processus de dérivation qui a comme but ultime la construction d'un mot composé uniquement de symbole terminaux. Le langage d'une grammaire est donc défini comme l'ensemble des mots sur le vocabulaire terminal qu'il est possible de former par dérivation de l'axiome de la grammaire.

Définition 22 (Langage engendré par une grammaire) :
Étant donné $\mathcal{G} = (V_T, V_N, S, R)$ une grammaire, on appelle ***langage engendré par*** $\mathcal{G}$ *le langage $\mathcal{L}(\mathcal{G})$ sur V_T des mots sur V_T qui dérivent de l'axiome S par la grammaire $\mathcal{G}$:*

$$\mathcal{L}(\mathcal{G}) = \{w \in V_T^* : S \overset{*}{\Rightarrow} w\}$$

6.3 La grammaire $\mathcal{G}_{\blacktriangleleft}$

Dans cette section, je présente la grammaire formelle $\mathcal{G}_{\blacktriangleleft}$ permettant d'engendrer le langage de requête $\mathcal{L}_{\blacktriangleleft}$. Je décris dans un premier temps une « version » réduite $\mathcal{G}^{\times}_{\blacktriangleleft}$ de cette grammaire dans la sous-section 6.3.1. Cette dernière est détaillée et objectivée à l'aide d'un exemple. La grammaire complète $\mathcal{G}_{\blacktriangleleft}$ est, quant à elle, présentée sans détails dans la sous-section 6.3.2.

Un certain nombre d'alphabets sont définis dans cette section. Ces derniers permettent de définir le **vocabulaire terminal** et le **vocabulaire non–terminal** de la grammaire $\mathcal{G}_{\blacktriangleleft}$. Conformément à l'usage, les variables (i.e. les non–terminaux) seront notées en majuscule et les terminaux en minuscule. De plus, afin d'assurer une meilleure lisibilité, des couleurs et des polices de caractère différentes seront utilisées. Ainsi :

— **les variables** seront notées « $VARIABLE$ » ;

— **les terminaux** seront notées « `terminal` ».

Les règles de production feront usage des méta-symboles définis dans la remarque 16 p. 157. Ces derniers seront notés en noir (e.g.« $(\cdots)^*$ »).

6.3.1 Le grammaire réduite $\mathcal{G}^{\times}_{\blacktriangleleft}$

Dans cette section, je présente la grammaire $\mathcal{G}^{\times}_{\blacktriangleleft}$ qui est une version « allégée » de $\mathcal{G}_{\blacktriangleleft}$. Cette dernière permet de se concentrer sur la logique syntaxique du langage $\mathcal{L}_{\blacktriangleleft}$. Elle permet de définir la syntaxe des contraintes d'attribut et des clause entité ainsi que la structure Booléenne d'une requête. En revanche, elle ne permet pas de définir :

— la syntaxe complète des différents types de données (i.e. numériques, dates, chaînes de caractère, libellés d'attribut, d'entité ou de relation) ;

— la syntaxe complète des opérateurs Booléens ;

— la syntaxe complète des opérateurs et comparateurs algébriques.

Ces derniers sont alors désignés par des éléments terminaux de la grammaire. Pour cela, on répartit ces éléments syntaxiques au sein de classes/groupes que l'on traite dans un premier temps comme des terminaux. La grammaire complète est donnée dans la sous-section 6.3.2.

Le vocabulaire terminal et non–terminal de la grammaire sont définis comme suit :

Vocabulaire terminal $T^{\times}_{\blacktriangleleft}$: On partitionne le vocabulaire terminal en quatre sous-ensembles :

$$\begin{aligned} T^{\times}_{0} &= \{\texttt{,},\ \texttt{(},\ \texttt{)},\ \texttt{[},\ \texttt{]},\ \texttt{\{},\ \texttt{\}}\} \\ T^{\times}_{1} &= \{\texttt{not},\ \texttt{and},\ \texttt{or}\} \\ T^{\times}_{2} &= \{\texttt{operator},\ \texttt{comparator},\ \texttt{backReference}\} \\ T^{\times}_{3} &= \{\texttt{numeric},\ \texttt{date},\ \texttt{text},\ \texttt{label},\ \texttt{path},\ \texttt{function}\} \end{aligned}$$

Le vocabulaire terminal $T^{\times}_{\blacktriangleleft}$ de $\mathcal{G}^{\times}_{\blacktriangleleft}$ est alors définit par :

$$T^{\times}_{\blacktriangleleft} = T^{\times}_{0} \cup T^{\times}_{1} \cup T^{\times}_{2} \cup T^{\times}_{3}$$

Vocabulaire non–terminal $N^{\times}_{\blacktriangleleft}$: De même, on définit un ensemble de variable constituant l'ensemble des non–terminaux :

$$N^{\times}_{\blacktriangleleft} = \{Q,\ Q_{\bigstar},\ Q_{\bigstar},\ E_{\bigstar},\ E_{\bigstar},\ E_{\bigstar},\ E_{\bigstar},\ P,\ J,\ J^{c},\ C,\ C^{b},\ C^{b}_{\bigstar},\ C^{b}_{\bigstar},\ C^{b}_{\bigstar},\ C^{b}_{\bigstar}\}$$

La Table 6.2 p. 164 explicite la signification et/ou le rôle de chacune de ces variables au sein du langage de requête $\mathcal{L}_{\blacktriangleleft}$.

Une grammaire « simplifiée » $\mathcal{G}^{\times}_{\blacktriangleleft}$ de $\mathcal{G}_{\blacktriangleleft}$ peut alors être définie :

Définition 23 ($\mathcal{G}_{\text{  }}$ simplifiée) :
La grammaire $\mathcal{G}^{\times}_{\text{}}$ simplifiée est définie par le quadruplet :

$$\mathcal{G}^{\times}_{\text{}} = \left(T^{\times}_{\text{}},\ N^{\times}_{\text{}},\ \mathrm{Q},\ R^{\times}_{\text{}}\right)$$

où :

l'ensemble des terminaux *est l'ensemble $T^{\times}_{\text{  }}$;*

l'ensemble des non-terminaux *est l'ensemble $N^{\times}_{\text{  }}$;*

l'axiome *est l'élément $Q \in N^{\times}_{\text{}}$;*

l'ensemble des règles de production *est l'ensemble $R^{\times}_{\text{  }}$ composé des* 16 *règles de production données ci-dessous :*

		Requête Booléenne	
1	Q	$\rightarrow$	$Q_{\text{}}$
2	$Q_{\text{}}$	$\rightarrow$	$Q_{\text{}}$ (**and**$Q_{\text{}}$ | **or**$Q_{\text{}}$)*
3	$Q_{\text{}}$	$\rightarrow$	($E_{\text{}}$ | **(**$Q_{\text{}}$**)**)
		Clause entité complète	
4	$E_{\text{}}$	$\rightarrow$	(**not**)?$E_{\text{}}$ ($E_{\text{}}$)?**)**
5	$E_{\text{}}$	$\rightarrow$	$E_{\text{}}$ (**and**$E_{\text{}}$ | **or**$E_{\text{}}$)*
6	$E_{\text{}}$	$\rightarrow$	(C | $E_{\text{}}$ | **(**$E_{\text{}}$**)**)
		Clause entité sans sous-clause de contrainte	
7	$E_{\text{}}$	$\rightarrow$	**label** (**,label**)* (J)? (P)?**(**
8	P	$\rightarrow$	**{** (**label** | **path**) (**,** (**label** | **path**))* **}**
9	J	$\rightarrow$	**[**J^c (**,**J^c)* **]**
10	J^c	$\rightarrow$	**label** (**comparator**)$^+$ ($C^b_{\text{}}$ | $C^b_{\text{}}$ | $C^b_{\text{}}$)
		Contraintes d'attribut	
11	C	$\rightarrow$	C^b (**comparator**)$^+$ C^b ((**comparator**)$^+$ C^b)*
12	C^b	$\rightarrow$	($C^b_{\text{}}$ | $C^b_{\text{}}$ | $C^b_{\text{}}$ | $C^b_{\text{}}$) (**operator** ($C^b_{\text{}}$ | $C^b_{\text{}}$ | $C^b_{\text{}}$ | $C^b_{\text{}}$))*
13	$C^b_{\text{}}$	$\rightarrow$	(**backReference**)?**label** (**.function**)?
14	$C^b_{\text{}}$	$\rightarrow$	**numeric** (**,numeric**)*
15	$C^b_{\text{}}$	$\rightarrow$	**date** (**,date**)*
16	$C^b_{\text{}}$	$\rightarrow$	**text** (**,text**)*

Les six règles de production de 11 à 16 permettent de définir la syntaxe des contraintes d'attributs. La règle 11 est notamment assimilable à la syntaxe 3 tandis que la règle 12 définit, elle, la syntaxe des sous-clauses `BLOCK` de la syntaxe 2. Les règles 14, 15 et 16 permettent la représentation syntaxique des différents types de données et prennent en compte la multi-valuation possible de ces valeurs tandis que la règle 13 définit la syntaxe d'un libellé d'attribut d'entité syntaxe 2.

Les règles 7 à 10 permettent de définir le « squelette » d'une clause entité. La règle 7 permet en effet de définir la syntaxe d'une sous-clause `ENTITY` d'une clause entité (i.e. syntaxe 1) en y ajoutant les possibles clauses d'option de chemin (i.e. sous-clause `{PATH_OPTS}` de la syntaxe 7 et de jointure (i.e. sous-clause `[JOINT_OPTS]` de la syntaxe 7) par l'intermédiaire des règles 8 et 9 mais ne permet pas de définir la syntaxe de la sous-clause de contrainte (i.e. sous-clause `CONSTRAINT` de la syntaxe 1).

La définition syntaxique d'une sous-clause de contrainte d'une clause entité est en réalité assurée par les règles 4, 5 et 6 qui permettent respectivement de **définir récursivement** les syntaxes :

— d'une clause entité complète en y ajoutant une sous-clause de contrainte par l'intermédiaire

de la règle 4 ;

— d'une contrainte de clause entité vue comme une agrégation Booléenne de contraintes à l'aide de la règle 5 ;

— d'une contrainte de manière générale qui peut alors être une contrainte d'attribut C ou une contrainte sémantique $E_{\bigstar}$ ou encore une expression Booléenne $E_{\bigstar}$ que l'on peut éventuellement parenthéser.

De même, les règles de production 1 à 3 permettent de définir, de manière tout aussi récursive, la syntaxe globale d'une requête en permettant d'agréger diverses clauses entité en expressions Booléennes à l'aide des opérateurs Booléens et de parenthèses non plus seulement au sein de la clause de contraintes d'une clause entité mais également entre clauses-entité situées à la racine de la requête.

L'exemple ci-dessous montre que cette grammaire permet d'engendrer les requêtes du langage $\mathcal{L}_{\bullet}$ dont la syntaxe a été décrite dans la section 6.1.

Exemple 27 :
La grammaire $\mathcal{G}_{\bullet}$ engendre la requête :

```
patient(date_naissance<1970-01-01)
AND (patient (sexe="M"
          AND analyse_biologique(
               code_exe(label = "Fer")
               AND valeur>31))
     OR
     patient (sexe="F"
          AND analyse_biologique(
               code_exe(label = "Fer")
               AND valeur>29)))
```

En effet, par dérivations successives à partir de l'axiome Q on obtient :

$$
\begin{aligned}
Q &\overset{1}{\Rightarrow} Q_{\bigstar} \\
&\overset{2}{\Rightarrow} Q_{\bigstar} \texttt{ and } Q_{\bigstar} \\
&\overset{3}{\Rightarrow} Q_{\bigstar} \texttt{ and (} Q_{\bigstar} \texttt{)} \\
&\overset{3}{\Rightarrow} E_{\bigstar} \texttt{ and (} Q_{\bigstar} \texttt{)} \\
&\overset{2}{\Rightarrow} E_{\bigstar} \texttt{ and (} Q_{\bigstar} \texttt{ or } Q_{\bigstar} \texttt{)} \\
&\overset{3}{\Rightarrow} \times 2 \quad E_{\bigstar} \texttt{ and (} E_{\bigstar} \texttt{ or } E_{\bigstar} \texttt{)} \\
&\overset{4}{\Rightarrow} E_{\bigstar} \; E_{\bigstar} \texttt{) and (} E_{\bigstar} \texttt{ or } E_{\bigstar} \texttt{)} \\
&\overset{7}{\Rightarrow} \texttt{label (} E_{\bigstar} \texttt{) and (} E_{\bigstar} \texttt{ or } E_{\bigstar} \texttt{)} \\
&\overset{5}{\Rightarrow} \texttt{label (} E_{\bigstar} \texttt{) and (} E_{\bigstar} \texttt{ or } E_{\bigstar} \texttt{)} \\
&\overset{6}{\Rightarrow} \texttt{label (} C \texttt{) and (} E_{\bigstar} \texttt{ or } E_{\bigstar} \texttt{)} \\
&\overset{4}{\Rightarrow} \times 2 \quad \texttt{label (} C \texttt{) and (} E_{\bigstar} \; E_{\bigstar} \texttt{) or } E_{\bigstar} \; E_{\bigstar} \texttt{))} \\
&\overset{7}{\Rightarrow} \times 2 \quad \texttt{label (} C \texttt{) and (label (} E_{\bigstar} \texttt{) or label (} E_{\bigstar} \texttt{))} \\
&\overset{5}{\Rightarrow} \times 2 \quad \texttt{label (} C \texttt{) and (label (} E_{\bigstar} \texttt{ and } E_{\bigstar} \texttt{) or label (} E_{\bigstar} \texttt{ and } E_{\bigstar} \texttt{))} \\
&\overset{6}{\Rightarrow} \times 2 \quad \texttt{label (} C \texttt{) and (label (} C \texttt{ and } E_{\bigstar} \texttt{) or label (} C \texttt{ and } E_{\bigstar} \texttt{))} \\
&\overset{6}{\Rightarrow} \times 2 \quad \texttt{label (} C \texttt{) and (label (} C \texttt{ and } E_{\bigstar} \texttt{) or label (} C \texttt{ and } E_{\bigstar} \texttt{))} \\
&\overset{4}{\Rightarrow} \times 2 \quad \texttt{label (} C \texttt{) and (label (} C \texttt{ and } E_{\bigstar} \; E_{\bigstar} \texttt{)) or label (} C \texttt{ and } E_{\bigstar} \; E_{\bigstar} \texttt{)))}
\end{aligned}
$$

$\overset{7}{\Rightarrow} \times 2$ `label ( C ) and ( label ( C and label ( E☆ ) ) or label ( C and label ( E☆ ) ) )`

$\overset{5}{\Rightarrow} \times 2$
```
label ( C ) and (
label ( C and label ( E☆ and E☆ ) )
or label ( C and label ( E☆ and E☆ ) ) )
```

$\overset{6}{\Rightarrow} \times 2$
```
label ( C ) and (
label ( C and label ( E★ and E☆ ) )
or label ( C and label ( E★ and E☆ ) ) )
```

$\overset{6}{\Rightarrow} \times 2$
```
label ( C ) and (
label ( C and label ( E★ and C ) )
or label ( C and label ( E★ and C ) ) )
```

$\overset{4}{\Rightarrow} \times 2$
```
label ( C ) and (
label ( C and label ( E☆ E☆ ) and C ) )
or label ( C and label ( E☆ E☆ ) and C ) ) )
```

$\overset{7}{\Rightarrow} \times 2$
```
label ( C ) and (
label ( C and label ( label ( E☆ ) and C ) )
or label ( C and label ( label ( E☆ ) and C ) ) )
```

$\overset{5}{\Rightarrow}\overset{6}{\Rightarrow} \times 2$
```
label ( C ) and (
label ( C and label ( label ( C ) and C ) )
or label ( C and label ( label ( C ) and C ) ) )
```

$\overset{11}{\Rightarrow} \times 7$
```
label ( C^b comparator C^b ) and (
  label ( C^b comparator C^b and label (
    label ( C^b comparator C^b )
    and C^b comparator C^b ) )
  or
  label ( C^b comparator C^b and label (
    label ( C^b comparator C^b )
    and C^b comparator C^b ) ) )
```

$\overset{12}{\Rightarrow} \times 14$
```
label ( C^b_⚭ comparator C^b_▦ ) and (
  label ( C^b_⚭ comparator C^b_▤ and label (
    label ( C^b_⚭ comparator C^b_▤ )
    and C^b_⚭ comparator C^b_▦ ) )
  or
  label ( C^b_⚭ comparator C^b_▤ and label (
    label ( C^b_⚭ comparator C^b_▤ )
    and C^b_⚭ comparator C^b_▦ ) ) )
```

Enfin, en appliquant les règles 13, 14, 15 *et* 16 *respectivement pour tous les variables* $C^b_{⚭}$, $C^b_{▦}$ *et* $C^b_{▦}$ *et* $C^b_{▤}$ *on obtient finalement un mot consitué exclusivement de terminaux :*

```
label ( label comparator date ) and (
  label ( label comparator text and label (
    label ( label comparator text )
    and label comparator numeric ) )
  or
  label ( label comparator text and label (
    label ( label comparator text )
    and label comparator numeric ) ) )
```

En remplaçant les terminaux `label`, `numeric`, `text`, `date`, `or`, `and` *et* `comparator` *par des*

valeurs concrètes appropriées, on retrouve bien la requête de départ :

```
label   (      label      comparator     date     )  and  (
  ↓     ↓        ↓            ↓           ↓     ↓    ↓   ↓
patient ( date_naissance      <       1970-01-01  )  AND  (

label   (  label  comparator  text  and        label         (
  ↓     ↓    ↓        ↓        ↓    ↓          ↓            ↓
patient (  sexe       =       "M"  AND  analyse_biologique  (

            label    (  label  comparator  text  )
              ↓      ↓    ↓        ↓        ↓    ↓
          code_exe   (  label      =      "Fer"  )

           and  label  comparator  numeric  )  )
            ↓     ↓        ↓          ↓     ↓  ↓
           AND  valeur      >         31    )  )

                           or
                           ↓
                          OR

label   (  label  comparator  text  and        label         (
  ↓     ↓    ↓        ↓        ↓    ↓          ↓            ↓
patient (  sexe       =       "F"  AND  analyse_biologique  (

            label    (  label  comparator  text  )
              ↓      ↓    ↓        ↓        ↓    ↓
          code_exe   (  label      =      "Fer"  )

          and  label  comparator  numeric  )  )  )
           ↓     ↓        ↓          ↓     ↓  ↓  ↓
          AND  valeur      >         29    )  )  )
```

6.3.2 $\mathcal{G}_{\text{}}$, l'intégrale

Par souci de simplicité, une grand partie des terminaux utilisés dans la grammaire $\mathcal{G}^{\times}_{\text{}}$ présentée dans la section précédente, représente en réalité des « groupes » de terminaux. Par exemple le terminal `text` ne représente pas simplement la chaîne de caractère `"text"` mais toutes les chaînes de caractères qu'il est possible de former (i.e. `""`, `"a"`, `"b"`, ..., `"aa"`, `"ab"`, etc.). En toute rigueur, seul l'alphabet $T_1^{\times}$ constitue réellement un ensemble de terminaux et les alphabets $T_2^{\times}$, $T_3^{\times}$ et $T_4^{\times}$ font partie de l'ensemble des non-terminaux.

Ainsi, d'un point de vue pleinement formel, la grammaire donnée précédemment n'engendre pas strictement les requêtes du langage $\mathcal{L}_{\text{}}$ mais plutôt des requêtes « à trous » dans lesquels les terminaux appartenant à $T_2^{\times} \cup T_3^{\times} \cup T_4^{\times}$ doivent être remplacés par des valeurs concrètes afin de véritablement constituer une requête du langage $\mathcal{L}_{\text{}}$.

Par souci de rigueur, on complète ici la grammaire $\mathcal{G}^{\times}_{\text{}}$ définie à la section précédente afin d'en fournir une définition formelle rigoureuse engendrant pleinement le langage $\mathcal{L}_{\text{}}$.

Définition 24 ($\mathcal{G}_{\text{}}$) :
$\mathcal{G}_{\text{}}$ est la grammaire formelle définie par :

$$\mathcal{G}_{\text{}} = (T_{\text{}}, N_{\text{}}, \mathrm{Q}, R_{\text{}})$$

où :

l'ensemble des terminaux $T_{\text{}}$ *est constitué de l'ensemble des caractères qu'il est possible de saisir sur une machine et dans un codage de caractère déterminé ;*

l'ensemble des variables $N_{\text{}}$ *est défini comme l'ensemble :*

$$\begin{aligned} N_{\text{}} = {} & \{Q,\ Q_{\text{}},\ Q_{\text{}},\ E_{\text{}},\ E_{\text{}},\ E_{\text{}},\ E_{\text{}},\ P,\ J,\ J^c,\ C,\ C^b,\ C^b_{\text{}},\ C^b_{\text{}},\ C^b_{\text{}},\ C^b_{\text{}}\} \\ & \cup \{C,\ C_\alpha,\ C_A,\ D\} \cup \{AND,\ OR,\ NOT\} \cup \{\text{}, \text{}, \text{}\} \\ & \cup \{\text{}, \text{}, \text{}, L,\ ID,\ ID_{\text{}}\} \cup \{P,\ P_E,\ P_R,\ F\} \end{aligned}$$

Le « rôle » ou la fonction syntaxique de chacune de ces variables est détaillé dans la Table 6.2 p. 164 ;

l'élément $Q \in N_{\text{⋖}}$ *est l'axiome de* $\mathcal{G}_{\text{⋖}}$;

l'ensemble des règles de production $R_{\text{⋖}}$ *est constitué de l'ensemble des règles de production suivantes :*

		Requête Booléene	
1	Q	$\rightarrow$	$Q_{\star}$
2	$Q_{\star}$	$\rightarrow$	$Q_{\star}\ (AND\ Q_{\star} \mid OR\ Q_{\star})^*$
3	$Q_{\star}$	$\rightarrow$	$(E_{\bigstar} \mid$ `(` $Q_{\star}$ `)` $)$
		Clause entité complète	
4	$E_{\bigstar}$	$\rightarrow$	$(NOT)?\ E_{\star}\ (E_{\bigstar})?$ `)`
5	$E_{\star}$	$\rightarrow$	$E_{\star}$ `(` $AND\ E_{\star} \mid OR\ E_{\star}$ `)`*
6	$E_{\star}$	$\rightarrow$	$(C \mid E_{\bigstar} \mid$ `(` $E_{\star}$ `)` $)$
		Clause entité sans sous-clause de contrainte	
7	$E_{\star}$	$\rightarrow$	L `(` `,` L `)`$^*\ (J)?\ (P)?$ `(`
8	P	$\rightarrow$	`{` $(L \mid P)$ `(` `,` $(L \mid P)$ `)`* `}`
9	J	$\rightarrow$	`[` J^c `(` `,` J^c `)`* `]`
10	J^c	$\rightarrow$	$L\ (\text{⋖})^+ \left(C^b_{\text{▦}} \mid C^b_{\text{▦}} \mid C^b_{\text{▤}}\right)$
		Contraintes d'attribut	
11	C	$\rightarrow$	$C^b\ (\text{⋖})^+\ C^b \left((\text{⋖})^+\ C^b\right)^*$
12	C^b	$\rightarrow$	$\left(C^b_{\text{⚷}} \mid C^b_{\text{▦}} \mid C^b_{\text{▦}} \mid C^b_{\text{▤}}\right)\ \left(\text{⊕}\ \left(C^b_{\text{⚷}} \mid C^b_{\text{▦}} \mid C^b_{\text{▦}} \mid C^b_{\text{▤}}\right)\right)^*$
13	$C^b_{\text{⚷}}$	$\rightarrow$	$(\text{◀◀})?\ L$ `(` `.` F `)`$?$
14	$C^b_{\text{▦}}$	$\rightarrow$	▦ `(` `,` ▦ `)`*
15	$C^b_{\text{▦}}$	$\rightarrow$	▦ `(` `,` ▦ `)`*
16	$C^b_{\text{▤}}$	$\rightarrow$	▤ `(` `,` ▤ `)`*
		Types de données (équivalent de $T_3^{\times}$*)*	
17	▦	$\rightarrow$	$(D)^+$ `(` `.` $(D)^*$ `)`$?$
18	▦	$\rightarrow$	$D\ D\ D\ D$ `-` $D\ D$ `-` $D\ D$ `(` $\sqcup\ D\ D$ `:` $D\ D$ `:` $D\ D$ `)`$?$
19	▤	$\rightarrow$	`"` $(c)^*$ `"`
20	L	$\rightarrow$	$(ID \mid ID_{\text{⚷}})$
21	ID	$\rightarrow$	$((C_A \mid C_\alpha \mid$ `_` $\mid$ `.` $\mid D))^+$
22	$ID_{\text{⚷}}$	$\rightarrow$	$C_\alpha\ ((C_\alpha \mid C_A \mid D))^+$
23	P	$\rightarrow$	$((P_E \mid P_R))^+$
24	P_E	$\rightarrow$	`[` P_R `]`
25	P_R	$\rightarrow$	$((?$ `?` $\mid$ `_` $\mid C_A \mid C_\alpha \mid D))^+$
26	F	$\rightarrow$	`f` `_` $C_\alpha\ ((C_A \mid C_\alpha))^+$ `()`
		Opérateurs et comparateurs algébriques (équivalent de $T_2^{\times}$*)*	
27	⊕	$\rightarrow$	(`-` $\mid$ `+` $\mid$ `*` $\mid$ `/`)
28	⋖	$\rightarrow$	(`!` $\mid$ `=` $\mid$ `<` $\mid$ `>`)
29	◀◀	$\rightarrow$	`(` `..` `/` `)`$^+$
		Opérateurs Booléens (équivalent de $T_1^{\times}$*)*	
30	AND	$\rightarrow$	(`AND` $\mid$ `and` $\mid$ `ET` $\mid$ `et`)
31	OR	$\rightarrow$	(`OR` $\mid$ `or` $\mid$ `OU` $\mid$ `ou`)
32	NOT	$\rightarrow$	(`NOT` $\mid$ `not` $\mid$ `SAUF` $\mid$ `sauf`)
		Règles de production génériques	
33	C_α	$\rightarrow$	(`a` $\mid$ `b` $\mid \cdots \mid$ `z`)
34	C_A	$\rightarrow$	(`A` $\mid$ `B` $\mid \cdots \mid$ `Z`)
35	D	$\rightarrow$	(`0` $\mid$ `1` $\mid$ `2` $\mid$ `3` $\mid$ `4` $\mid$ `5` $\mid$ `6` $\mid$ `7` $\mid$ `8` $\mid$ `9`)
$\forall$`c` $\in T_{\text{⋖}}$	c	$\rightarrow$	`c`

Variable	Fonction/rôle syntaxique de la variable
Q	une requête complète.
$Q_{\bigstar}$	une agrégation Booléenne de requête partielle.
$Q_{\bigstar}$	une requête partielle éventuellement parenthésée.
$E_{\bigstar}$	une entité (clause entité) complète.
$E_{\bigstar}$	une agrégation Booléenne de contraintes d'entité.
$E_{\bigstar}$	une contrainte d'entité.
$E_{\bigstar}$	une entité sans contraintes (incomplète).
P	une clause d'options de chemin.
J	une clause d'options de jointure.
J^c	une contrainte d'une clause d'option de jointure.
C	une contrainte d'attribut.
C^b	un bloc de contraintes d'attribut.
C^b	un bloc représentant un attribut d'entité.
C^b	un bloc numérique de contrainte.
C^b	un bloc de type date de contrainte.
C^b	un bloc textuel de contrainte.
	une donnée numérique concrète (`numeric`).
	un date concrète (`date`).
	une chaîne de caractère concrète. (`text`).
L	un identifiant/libellé d'attribut, de relation ou d'entité. (`label`).
ID	un identifiant d'attribut, de relation ou d'entité.
ID	un libellé d'attribut, de relation ou d'entité.
P	un chemin (« **P**ath »).
P_E	un identifiant/libellé d'entité apparaissant dans un chemin.
P_R	un identifiant/libellé de relation apparaissant dans un chemin.
F	une fonction.
	un opérateur algébrique (`operateur`).
	un comparateur algébrique (`comparateur`).
	un élément syntaxique de type « **B**ack **R**eference ».
AND	un opérateur Booléen de type « ET ».
OR	un opérateur Booléen de type « OU ».
NOT	un opérateur Booléen de type « SAUF ».
C_α	un caractère alphabétique en minuscule.
C_A	un caractère alphabétique en majuscule.
D	un chiffre (« **D**igit »).
c	un caractère quelconque appartenant à l'ensemble des terminaux.

TABLE 6.2 – Correspondance entre les non-terminaux de $\mathcal{G}_{\bullet}$ et les éléments syntaxiques qu'ils permettent de représenter au sein du langage de requête $\mathcal{L}_{\bullet}$.

6.4 Implémentation

Le langage $\mathcal{L}_{\text{}}$, comme tout langage de requête, est un moyen d'exprimer des besoins d'information à travers une syntaxe « constante et sans ambiguïté » qu'un SRI (en l'occurrence ici le SSE_{SQL} et le SSE_{NoSQL}) est en capacité de traiter (cf. section 6.1 p. 130). Que l'utilisateur accède à cette information à travers une requête qu'il formule par lui même ou bien par l'intermédiaire d'une interface graphique qui se charge de la formuler pour lui, le SRI reçoit, en premier lieu, systématiquement, et quelque soit le contexte, une requête textuelle sous forme d'une **chaîne de caractère brute** (Figure 6.1).

Un tel SRI se doit donc d'être en capacité d'interpréter cette requête textuelle. $\mathcal{L}_{\text{}}$ est cependant un langage basé sur une syntaxe complexe définie par une grammaire formelle (cf. définition 24). Plusieurs modes d'interrogation du SSE_{SQL} et SSE_{NoSQL} ont été implémentés dans le cadre de cette thèse, cependant, l'interprétation d'une $\mathcal{L}_{\text{}}$–requête s'avère néanmoins nécessaire quelque soit le mode d'interrogation utilisé.

La représentation d'une requête écrite dans un langage quelconque en une forme propice à son traitement informatique passe par une étape d'**analyse syntaxique** qui permet de mettre en évidence la structure d'une chaîne de caractère relativement aux règles de production de la grammaire engendrant le langage dans lequel est écrite cette requête.

Dans cette section, le **parseur** du langage $\mathcal{L}_{\text{}}$ rendant possible cette analyse syntaxique pour une $\mathcal{L}_{\text{}}$–requête est présenté. Il fournit en sortie un objet informatique qui, bien que partiellement représentatif d'une $\mathcal{L}_{\text{}}$–requête, constitue la première étape de la chaîne d'interprétation de cette dernière. Dans la suite de ce mémoire, on notera $\text{parser}_{\text{}}$ ce parseur. Parser du langage $\mathcal{L}_{\text{}}$ ($\text{parser}_{\text{}}$) ayant été implémenté avec l'outil **J**ava **C**ompiler **C**ompiler™ (JavaCC™) [5, 6][155, 156], on présente dans un premier temps ce dernier ainsi que les concepts fondamentaux de l'analyse syntaxique.

6.4.1 Le générateur JavaCC™

JavaCC™ est un logiciel libre distribué selon les termes de la licence **B**erkeley **S**oftware **D**istribution (BSD) [7]. Il a été initialement conçu pour faciliter l'implémentation de langage de programmation et son nom (i.e. « Compilateur de Compilateur ») provient de cette fonction première. Cependant, l'utilisation de JavaCC™ ne se limite pas à l'implémentation de langage de programmations.

Plus concrètement, JavaCC™ permet de **générer** un **analyseur syntaxique** (ou « **parser** » en anglais) et un **analyseur lexical** en Java™ à partir d'une description d'un langage.

Un analyseur lexical permet de lire une séquence de caractères et de la partitionner en une séquence d'objet appelés **tokens** [8]. Ces derniers correspondent alors à des sous-séquences de caractères de la chaîne initiale appelés **lexèmes**. Un analyseur lexical permet donc d'identifier les éléments lexicaux symboliques du langage au sein de la chaîne de caractères brute fournie en entrée.

Un parseur consomme la séquence de tokens fournie par un analyseur lexical et permet d'appliquer les règles syntaxiques (i.e. les règles de production) du langage. Il permet notamment d'identifier les séquences valides ou invalides de tokens mais aussi généralement de construire une représentation structurée et cohérente vis à vis du langage de la séquence de caractère fournie en entrée. Une illustration du rôle de l'analyseur lexical et du parser est donnée Figure 6.3.

5. : « **C**ompilateur de **C**ompilateur en **Java**™ »
6. url : `https://javacc.org/`
7. : « Licence **B**erkeley de **D**istribution de Logiciels »
8. : « segments » est probablement une traduction cohérente dans ce contexte.

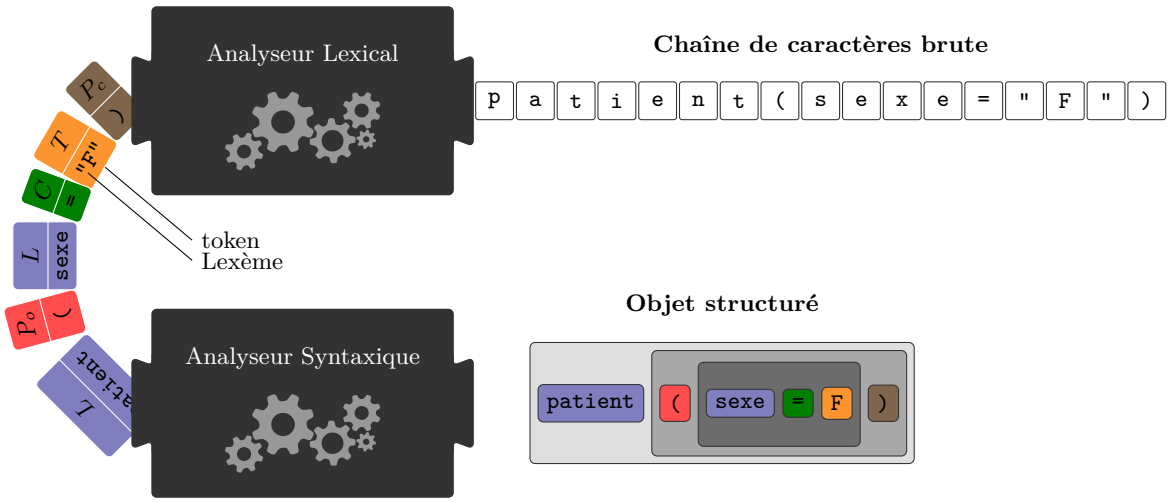


FIGURE 6.3 – Illustration du rôle d'un analyseur lexical et d'un parseur (i.e. un analyseur syntaxique) dans le traitement d'une chaîne de caractères en entrée. La chaîne de caractères traitée est la $\mathcal{L}_{\text{<}}$–requête `patient(sexe="F")`. L'analyseur syntaxique traite cette requête caractère par caractère afin d'identifier une séquence de tokens qui est ensuite analysée par le parseur à l'aide des règles de production de la grammaire $\mathcal{G}_{\text{<}}$ pour en construire un objet structuré.

6.4.2 Exploitation du parseur

Dans le cadre des développements effectués au cours de cette thèse, l'outil JavaCC™ a été utilisé pour générer un parseur pour le langage $\mathcal{L}_{\text{<}}$: le parser$_{\text{<}}$. Ce dernier est utilisé par le SSE_{SQL} et le SSE_{NoSQL} pour **valider la syntaxe des $\mathcal{L}_{\text{<}}$–requêtes** qu'il reçoit en entrée et pour en **construire une représentation traitable informatiquement**.

JavaCC™ prend en entrée un fichier d'extension `.tt` permettant de décrire les règles de production de la grammaire et les règles (lexicales) d'identification des tokens. Ce fichier a été constitué en intégrant les règles de production de la grammaire $\mathcal{G}_{\text{<}}$ données dans la section 6.3. JavaCC™ permet de renseigner les règles de production d'une grammaire en utilisant le méta-langage BNF augmenté des méta-symboles des extensions de ce langage déjà évoqués dans la remarque 16. Les règles de production qui ont été fournies à JavaCC™ sont donc en tout point concordantes avec celles citées dans ce mémoire.

Les règles lexicales (i.e. les tokens) ont quant à elles permis de définir la syntaxe des types de données gérées par le langage $\mathcal{L}_{\text{<}}$ et présentées dans la sous-section 6.1.2.

JavaCC™ autorise l'intégration de code Java™ à la description de chaque règle de production. Cette fonctionnalité a été utilisée pour donner à parser$_{\text{<}}$ la capacité de renvoyer un POJO Java™ appelé SSEQuery fournissant une représentation structurée basique d'une $\mathcal{L}_{\text{<}}$–requête passée en entrée.

Une SSEQuery est assimilable à une simple séquence de POJO Java™ plus basique représentant tous des variables symboliques de la grammaire $\mathcal{G}_{\text{<}}$. L'objet SSEQuery fournit donc un moyen d'identifier les différents **éléments de syntaxe** d'une $\mathcal{L}_{\text{<}}$–requête et d'en permettre la lecture informatiquement.

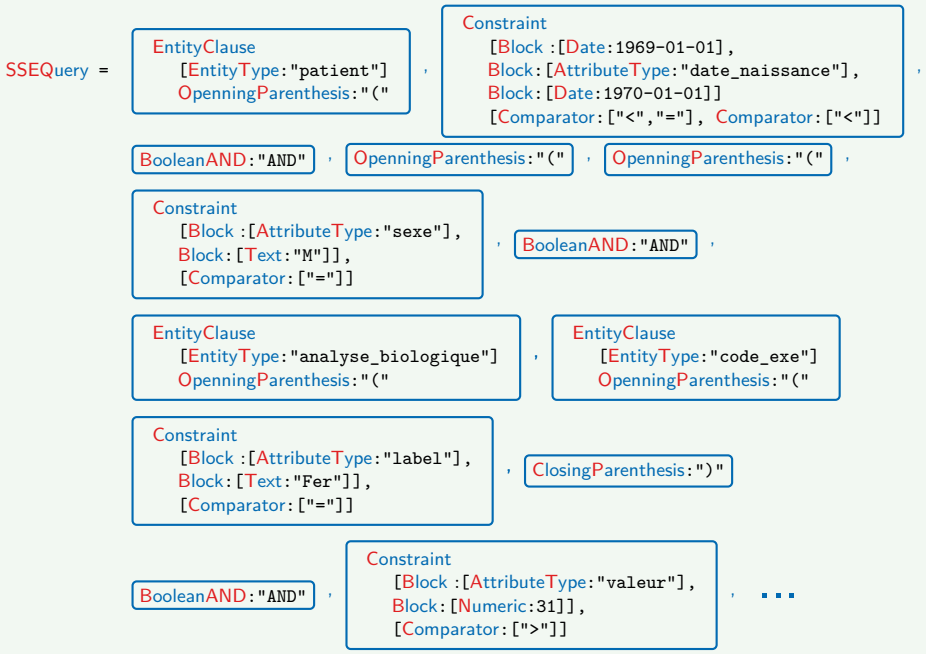
En tant que liste, le POJO SSEQuery ne met cependant pas en évidence la logique Booléenne d'une $\mathcal{L}_{\text{<}}$–requête ni même le caractère « imbriqué » de certaines clauses entité. JavaCC™ ne fournit nativement pas la possibilité de générer un arbre syntaxique.

La structure rigoureuse d'une SSEQuery n'est pas précisée dans ce mémoire, en revanche, l'exemple 28 suivant illustre brièvement la structure d'une SSEQuery obtenue en sortie de parser$_{\text{<}}$.

Exemple 28 :
Étant donnée la $\mathcal{L}_{\text{}}$–requête suivante :

```
patient(
   1969-01-01<=date_naissance<1970-01-01
   AND (
      (
         sexe="M"
         AND analyse_biologique(
            code_exe(label = "Fer")
            AND valeur>31)
      )
      OR
      (
         sexe="F"
         AND analyse_biologique(
            code_exe(label = "Fer")
            AND valeur>29)
      ))
```

L'objet SSEQuery obtenu en sortie de parser$_{\text{}}$ est du type :



Dans ce chapitre, le langage $\mathcal{L}_{\text{}}$ a, dans un premier temps, été décrit précisément de manière pratique et concrète à l'aide de syntaxes informelles et d'exemples. Ceci a permis d'appréhender l'expressivité de ce langage.

Une définition formelle en a ensuite été donnée à l'aide de la grammaire formelle $\mathcal{G}_{\text{  }}$ qui engendre ce langage.

Le parseur parser$_{\text{  }}$ a ainsi pu être construit à partir de celle-ci. Ce dernier permet d'analyser une $\mathcal{L}_{\text{  }}$–requête et d'en extraire un objet structuré SSEQuery traitable informatiquement.
Cependant, les SSEQuerys ne fournissent pas une représentation exhaustive de toute la complexité syntaxique du langage de requêtes $\mathcal{L}_{\text{  }}$. Ils permettent une première étape d'identification des éléments syntaxiques des $\mathcal{L}_{\text{}}$–requêtes mais ne mettent pas en évidence la logique syntaxique

de ces dernières. Des traitements additionnels sont donc nécessaire afin de réaliser un moteur de recherche apte à exécuter de tels requêtes.

De plus, le langage $\mathcal{L}_{\text{  }}$ a été défini pour exprimer des besoins d'information relatifs à des données cliniques provenant de l'EDSS du CHU de Rouen. Cependant, le SI du **D2IM** met à disposition divers outils de santé qui n'entrent pas nécessairement dans le cadre de ce cas d'usage et notamment des outils de RI documentaire et bibliographique en santé.

Le langage $\mathcal{L}_{\text{}}$ et, a fortiori, le moteur de recherche qui l'implémente se doit d'être intégré de manière cohérente au sein du SI du **D2IM** afin d'assurer les fonctionnalités de rétrocompatibilité nécessaire au bon fonctionnement de l'ensemble des outils.

Dans le chapitre suivant, l'intégration du $\mathcal{L}_{\text{}}$ au sein du SI du **D2IM** sera présenté. Cette intégration vise à faire cohabiter divers langages de requêtes d'objectifs variés. L'implémentation des moteurs SSE_{SQL} et SSE_{NoSQL} sera ensuite présenté.

Chapitre 7

La recherche d'information par la pratique

Sommaire

7.1 Modes de requêtage 171
7.1.1 Le langage $\mathcal{L}_{\text{DocCISMeF}}$ 173
7.1.2 Le langage $\mathcal{L}_{\mathbb{B}}$ 176
7.1.3 Le Langage Naturel 178
7.1.4 Le langage $\mathcal{L}_{\#}$ 179
7.2 Logique interne 187
7.2.1 Structure arborescente 187
7.2.2 Le Semantic Search Engine SQL (SSE_{SQL}) 190
7.2.3 Le Semantic Search Engine NoSQL (SSE_{NoSQL}) 194

Le chapitre précédent a permis de décrire la modélisation théorique et l'implémentation du langage $\mathcal{L}_{\text{share}}$. Dans ce chapitre, son exploitation à des fins de RI est abordée. Elle se matérialise par le développement des deux moteurs de recherche, le **S**emantic **S**earch **E**ngine **SQL** (SSE_{SQL}) et le **S**emantic **S**earch **E**ngine **NoSQL** (SSE_{NoSQL}). Le langage $\mathcal{L}_{\text{share}}$ constitue le cœur de ces deux moteurs dans le sens où toutes leurs logiques d'exécution reposent sur la syntaxe de ce langage.

Aujourd'hui, seul le SSE_{NoSQL} est encore maintenu. Comme expliqué précédemment, ce moteur est utilisé dans divers contextes opérationnels et notamment au sein de notre SI comme moteur de recherche sous-jacent aux outils de RI bibliographiques DocCISMeF et LiSSa mais également dans le cadre du projet de création de l'EDSS confié par le CHU de Rouen à notre équipe de recherche. Bien que le SSE_{SQL} ait été initialement le premier à voir le jour, ce dernier a été abandonné afin de palier aux faibles performances des temps de traitements résultant de l'exploitation du SGBDR ORACLE (cf. chapitre 5).

Bien que basé sur des SGBDs différents, le SSE_{SQL} et le SSE_{NoSQL} ont en commun de pouvoir tous deux interpréter et exécuter des $\mathcal{L}_{\text{share}}$–requêtes. Néanmoins, la logique d'exécution qu'ils emploient diffèrent. Cette différence s'explique principalement par le fait que le SSE_{SQL} traduit ces requêtes en SQL alors que le SSE_{NoSQL} les traduit en une succession d'ordres rendus disponibles par l'API de requêtage du SGBD NINJAC.

Les fonctionnalités du SSE_{SQL} sont plus étendues que celles du SSE_{NoSQL}. Fonctionnellement, le SSE_{NoSQL} ne peut en effet couvrir la totalité de l'expressivité du langage de requête $\mathcal{L}_{\text{share}}$ contrairement au SSE_{SQL}. Ceci s'explique principalement par la « puissance » du langage SQL qui offre nativement les fonctionnalités génériques nécessaires à l'implémentation des fonctionnalités de RI. Le cœur du SSE_{SQL} à donc pour principal rôle de **transcrire les $\mathcal{L}_{\text{share}}$–requêtes en requêtes SQL**.

En revanche, dans le cadre de l'exploitation de la base de données NoSQL Infinispan, les

fonctionnalités nécessaires à la RI ont dû être développées y compris au niveau de la donnée elle-même par le biais du SGBD NINJAC et notamment de la maintenance des maps de jointure et des index Lucene permettant une recherche plein texte. Bien que, la transcription d'une $\mathcal{L}_{\bullet}$–requête en requête SQL n'est pas triviale, la logique interne du SSE_{NoSQL} suppose une plus grande atomicité dans le sens où une $\mathcal{L}_{\bullet}$–requête est exécutée **contrainte par contrainte par le SSE_{NoSQL} lui-même (en faisant appel à NINJAC) et non d'un seul bloc par un SGBD tierce**.

Le SSE_{SQL} et le SSE_{NoSQL} proposent différentes fonctionnalités et notamment des modes d'interrogations différents ainsi que la possibilité de trier les ressources obtenues en sortie.

7.1 Modes de requêtage

Le SSE$_{NoSQL}$ est utilisé pour assurer différents types de RI. Il s'attache premièrement, à permettre une **RI documentaire** classique dans le cadre de DocCiSMeF et LiSSa. Dans un second temps, il assure une RI plus complexe au sein des données patient de l'EDSS du CHU de Rouen. Le langage de requête $\mathcal{L}_{\text{graphe}}$ est adapté à ce deuxième cas d'utilisation mais reste inutilement complexe en ce qui concerne la RI documentaire. Afin d'adapter le SSE$_{NoSQL}$ à ces différents contextes d'utilisation, cinq modes d'interrogation ont été proposés. Parmi ces cinq modes, trois correspondent à la possibilité d'utiliser trois langages de requête logiques différents. Pour chacun de ces langages, les outils permettant son exploitation et sa manipulation informatique ont été développés :

Le langage $\mathcal{L}_{\text{graphe}}$: muni du parser$_{\text{graphe}}$ permettant de générer un objet Java™ SSEQuery représentant une $\mathcal{L}_{\text{graphe}}$–requête.

Le langage $\mathcal{L}_{\text{DocCiSMeF}}$: muni à la fois :

- du parser$_{\text{DocCiSMeF}}$ permettant de générer une objet Java™ DCQuery représentant informatiquement les requêtes écrites dans ce langage que l'on notera $\mathcal{L}_{\text{DocCiSMeF}}$–requêtes ;
- d'un convertisseur $\mathcal{L}_{\text{DocCiSMeF}} \rightarrow \mathcal{L}_{\text{graphe}}$ permettant de « traduire » une $\mathcal{L}_{\text{DocCiSMeF}}$–requête en une $\mathcal{L}_{\text{graphe}}$–requête équivalente.

Le langage $\mathcal{L}_{\mathbb{B}}$: muni :

- du parser$_{\mathbb{B}}$ permettant de générer un objet Java™ BQuery représentant informatiquement les requêtes écrites dans ce langage que l'on notera $\mathcal{L}_{\mathbb{B}}$–requête ;
- d'un convertisseur $\mathcal{L}_{\mathbb{B}} \rightarrow \mathcal{L}_{\text{DocCiSMeF}}$ permettant de « traduire » une $\mathcal{L}_{\mathbb{B}}$–requête en une $\mathcal{L}_{\text{DocCiSMeF}}$–requête équivalente.

Le moteur de recherche peut également être interrogé par l'intermédiaire de requêtes écrites en langage naturel. Deux modes d'interrogation sont disponibles pour interpréter ces requêtes :

Par une conversion $\mathcal{L}_{\#} \rightarrow \mathcal{L}_{\text{graphe}}$: Cette méthode permet de définir des syntaxes génériques et paramétrables et d'identifier ces syntaxes au sein des requêtes en langage naturel fournies en entrée du SSE$_{SQL}$. Les requêtes interprétables par cet outil sont donc intrinsèquement écrites dans un langage spécifique que l'on notera par la suite $\mathcal{L}_{\#}$. L'outil est ainsi assimilable à un convertisseur $\mathcal{L}_{\#} \rightarrow \mathcal{L}_{\text{graphe}}$ de $\mathcal{L}_{\#}$–requête en $\mathcal{L}_{\text{graphe}}$–requête.

Par l'intermédiaire de l'ECMT : Il permet d'annoter sémantiquement les requêtes en langage naturel et d'en construire une $\mathcal{L}_{\text{DocCiSMeF}}$–requête lorsqu'il s'agit d'exploiter le moteur de recherche dans un contexte de RI documentaire.

Dans la pratique, seules les $\mathcal{L}_{\text{graphe}}$–requêtes sont en interne réellement gérées par le moteur de recherche. Son interrogation à l'aide des différents types de requête évoqués dans cette section passe donc par diverses étapes de conversion en $\mathcal{L}_{\text{graphe}}$–requêtes équivalentes. L'organigramme de programmation donné dans la Figure 7.1 indique le processus de gestion d'une requête reçue en entrée du moteur de recherche jusqu'à sa traduction en $\mathcal{L}_{\text{graphe}}$–requête. La gestion d'une requête se fait en deux étapes majeures :

Étape n° 1 : Identification du type d'une requête (i.e.$\mathcal{L}_{\text{graphe}}$–requête, $\mathcal{L}_{\mathbb{B}}$–requête, etc.).

Étape n° 2 : Conversion de cette requête en $\mathcal{L}_{\text{graphe}}$–requête puis en SSEQuery.

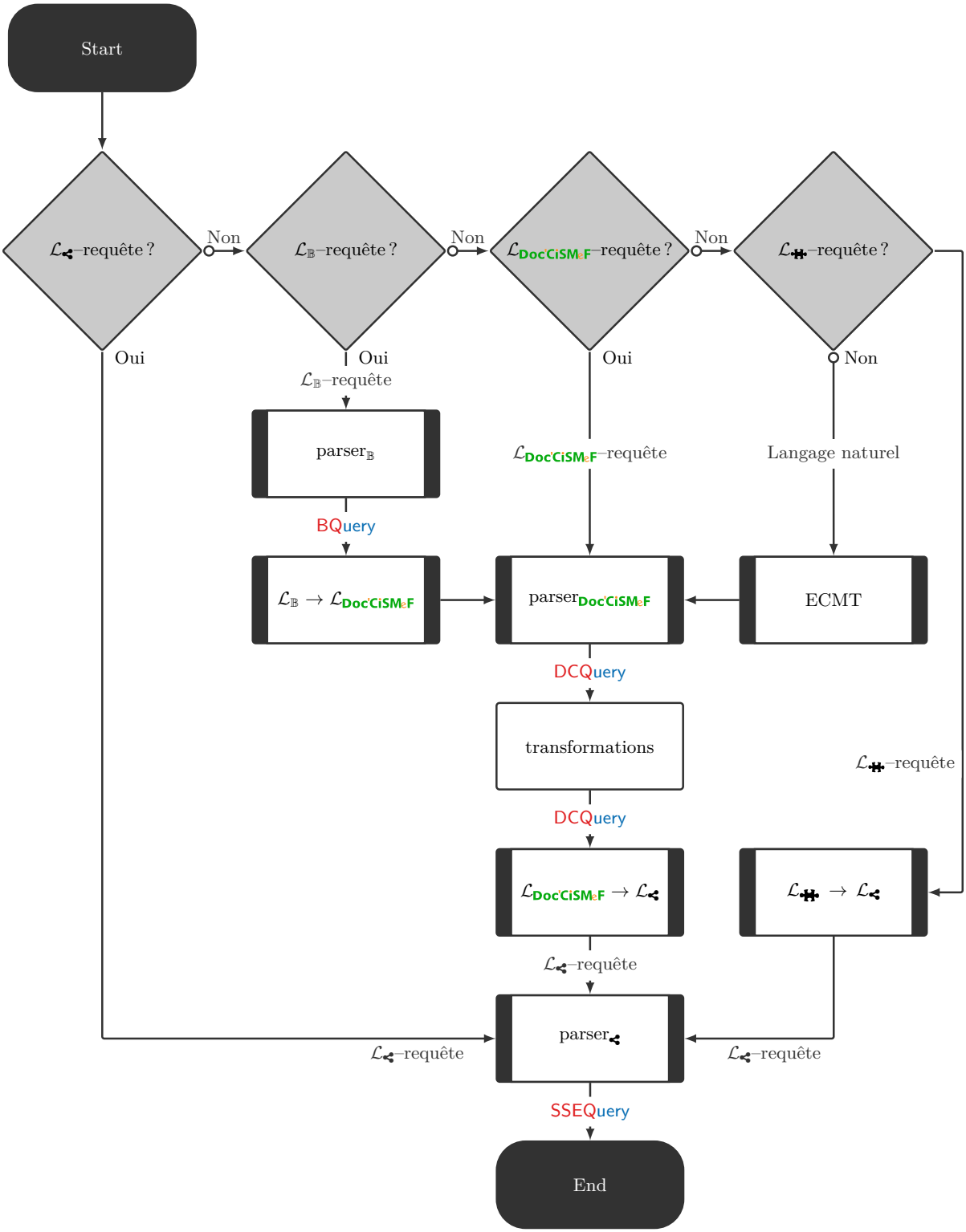


FIGURE 7.1 – Organigramme de programmation représentant le processus de gestion d'une requête donnée en entrée du moteur de recherche. Les cinq modes d'interrogation possibles (viz. $\mathcal{L}_{\text{share}}$–requête, $\mathcal{L}_{\text{DocCiSMoF}}$–requête, $\mathcal{L}_{\mathbb{B}}$–requête, $\mathcal{L}_{\text{puzzle}}$–requête et requête en langage naturel) sont représentés. Le format d'une requête est identifié afin de pouvoir se ramener de manière systématique à une $\mathcal{L}_{\text{share}}$–requête.

Le langage $\mathcal{L}_{\text{DocCISMeF}}$ est un langage pivot pour différents processus de conversion (i.e. étape n° 2). La conversion en une $\mathcal{L}_{\text{share}}$–requête d'une $\mathcal{L}_{\mathbb{B}}$–requête ou d'une requête en langage naturel pur s'effectue au préalable, par une conversion intermédiaire en une $\mathcal{L}_{\text{DocCISMeF}}$–requête. En revanche, la conversion des $\mathcal{L}_{\text{puzzle}}$–requêtes en $\mathcal{L}_{\text{share}}$–requêtes est autonome et s'effectue sans conversion intermédiaire. Ceci s'explique principalement par la couverture fonctionnelle de chacun de ces langages. La Figure 7.2 illustre l'ensemble des modes d'interrogation du SSE_{NoSQL} en fonction du contexte de RI pour lesquels il sont destinés et adaptés et en fonction de la complexité à l'aide de ces langages du point de vue de l'utilisateur.

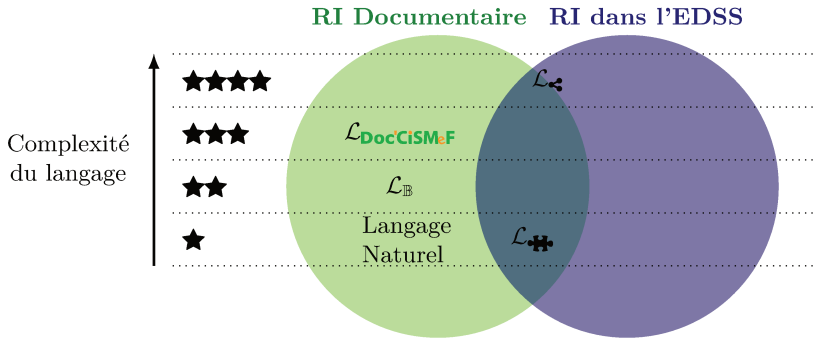


FIGURE 7.2 – Représentation des différents modes d'interrogation du SSE_{NoSQL} en fonction de leur complexité d'utilisation pour un utilisateur et du contexte de RI pour lequel ils sont adaptés.

Le langage naturel, le langage $\mathcal{L}_{\mathbb{B}}$ et le langage $\mathcal{L}_{\text{DocCISMeF}}$ sont en effet des langages dont le formalisme, la philosophie et l'expressivité sont destinés à une utilisation dans un contexte de RI documentaire alors que le langage $\mathcal{L}_{\text{share}}$ est lui davantage destiné à une utilisation dans une contexte de RI au sein des données patient.

Le langage $\mathcal{L}_{\text{DocCISMeF}}$ peut être considéré comme le langage logique de référence pour la RI documentaire au sein du SI du D2IM tandis que le langage $\mathcal{L}_{\text{share}}$ est celui de la RI au sein de l'EDSS. Si l'on ne considère que les langage de requête logique, ces deux langages correspondent par ailleurs aux deux langages les plus complexes en terme de syntaxe et possédant la plus grande expressivité disponible pour chacun de ces deux contextes de RI. Dans l'absolu, le langage naturel possède évidement la plus grande expressivité mais ne constitue pas un langage de requête logique. De plus sa traduction en requête logique en vue de son exécution au sein des moteurs sémantiques ne permet pas une aussi grande finesse d'accès aux données et informations que les langages $\mathcal{L}_{\text{share}}$ et $\mathcal{L}_{\text{DocCISMeF}}$.

Dans les sections suivantes, les différents modes d'interrogation sont présentés.

7.1.1 Le langage $\mathcal{L}_{\text{DocCISMeF}}$

Tout comme le langage $\mathcal{L}_{\text{share}}$, le langage $\mathcal{L}_{\text{DocCISMeF}}$ est un langage de requête Booléen basé sur une grammaire formelle. Ce dernier est muni du $\text{parser}_{\text{DocCISMeF}}$ généré à l'aide de l'outil JavaCC™. La définition et l'implémentation de ce langage n'ont cependant pas été réalisées dans le cadre de cette thèse.

Historiquement, le langage $\mathcal{L}_{\text{DocCISMeF}}$ est le langage de référence du moteur de recherche DocCISMeF. Bien que le moteur de recherche applicatif sous-jacent à cet outil ait aujourd'hui été remplacé par le SSE_{NoSQL}, le langage $\mathcal{L}_{\text{DocCISMeF}}$ est dans la pratique toujours privilégié par les documentalistes du D2IM pour accéder et rechercher des ressources documentaires et bibliographiques à l'aide de DocCISMeF et de LiSSa.

La syntaxe du langage $\mathcal{L}_{\text{DocCISMeF}}$ est largement inspirée du langage de requête proposé par l'outil Ovid® (Ovid®) [1] qui donne accès à des bases de données bibliographiques en ligne, à des

1. url : `http://www.ovid.com/site/index.jsp`

revues universitaires et à d'autres produits, principalement dans le domaine des sciences de la santé. Sa philosophie de requêtage est également très proche de celle du langage de requête proposé par le moteur de recherche de données bibliographiques PubMed.

De même que les langages de requête de PubMed et Ovid, la syntaxe du langage $\mathcal{L}_{\text{DocCiSMeF}}$ repose sur l'exploitation d'élément syntaxique permettant de créer des contraintes sur des **champs de recherche**[2]. Un champ de recherche n'est rien d'autre qu'un attribut ou une méta-donnée d'une ressource documentaire ou bibliographique qu'il est possible d'exploiter dans le cadre de la RI sur ces ressources. Ces champs de recherche peuvent par exemple correspondre au titre, à la date de publication ou encore au nom de l'auteur d'une ressource. Dans le cas du langage $\mathcal{L}_{\text{DocCiSMeF}}$, les contraintes de champs de recherche sont de la forme suivante :

</> Syntaxe 10 (contrainte de champs de recherche) :
*Un **contrainte de champs de recherche** est de la forme :*

$$\texttt{VALEUR.code[OPTION}_1\texttt{][OPTION}_2\texttt{],...}$$

où :

`code` : *est un libellé généralement court désignant un champ de recherche d'une ressource documentaire ou bibliographique.*

`VALEUR` : *la valeur que doit prendre ce dernier champ de recherche.*

$\texttt{[OPTION}_1\texttt{][OPTION}_2\texttt{],...}$: *est un ensemble de clauses optionnelles d'options permettant de préciser des spécificités dans le processus de recherche de ressources.*

La contrainte de champ de recherche `VALEUR.code[OPTION]` désigne alors l'ensemble des ressources dont l'attribut désigné par `code` a pour valeur `VALEUR`.

En plus de ces contraintes de champ de recherche, le langage $\mathcal{L}_{\text{DocCiSMeF}}$ permet l'utilisation des parenthèses (i.e. « `(` » et « `)` ») et des opérateurs Booléens `ET`, `OU` et `SAUF`. On ne dressera pas dans ce mémoire la liste de la totalité des champs de recherche disponibles pour ce langage[3] cependant, quelques exemples de requêtes qu'il est possible de former avec ce langage sont donnés ci-dessous :

Exemple 29 :
*Le code champ `.mc`, signifiant **M**ot **C**lé permet par exemple de rechercher les ressources indexées avec un concept :*

`maladie.mc` : *Désigne l'ensemble des ressources indexées avec les concepts de **maladie** toutes TOs confondues ou l'un des descendants de ces concepts.*

`maladie.mc[TER_MSH]` : *Désigne l'ensemble des ressources indexées avec le concept « maladie » [D004194 (MeSH)] de la terminologie MeSH ou l'un de ses descendants.*

`maladie.mc[TER_MSH][NOEXPL]` : *Désigne l'ensemble des ressources indexées avec le concept « maladie » [D004194 (MeSH)] sans nécessairement l'être avec l'un des ses descendants (`NOEXPL` ⇔ pas d'explosion hiérarchique du terme).*

De même le code champ `.ti` désigne le titre d'une ressource :

`asthme.mc[TER_MSH] OU asthme.ti` : *Désigne l'ensemble des ressources indexées avec « maladie » [D004194 (MeSH)] ou contenant le mot « asthme » dans le titre.*

`asthme.mc[TER_MSH] ET ((nourrisson.ti OU bébé.ti) SAUF enfant.ti)` : *désigne l'ensemble des ressources indexées avec le concept « maladie » [D004194 (MeSH)] et contenant dans le titre le mot « nourrisson » ou « bébé » mais pas le mot « enfant ».*

2. : Ce terme étant une traduction de l'expression « Search Field » utilisé par le moteur de recherche PubMed

3. url : La plus grande partie des codes champs disponibles sont cependant décrit à l'url suivante : `http://www.chu-rouen.fr/cismef/aide/liste-des-abreviations-des-champs-utilises-dans-doccismef-et-lissa/`

Enfin, le code champ `.an` *permet par exemple de cibler des ressources publiées à une année précise :*

`asthme.mc[TER_MSH] ET 2017.an :` *Désigne les ressources traitant de l'asthme publiées en 2017.*

`asthme/diagnostic.mc[TER_MSH] ET diagnostic.ti ET 2015->2018.an :` *Désigne les ressources traitant du diagnostic de l'asthme, dans lesquelles le mot « asthme » apparaît dans le titre et publiées entre 2015 et 2018.*

Dans le cadre de la RI documentaire classique, le type de ressource renvoyé par un moteur de recherche est toujours le même. Il peut correspondre à des ressources documentaires (e.g. DocCISMeF) ou bien à des ressources bibliographiques (e.g. LiSSa). Quoi qu'il en soit, le « type d'objet » obtenu en sortie ne varie pas contrairement au cas d'usage de la RI au sein de l'EDSS qui, elle, peut renvoyer divers types d'objets (e.g. patient mais aussi séjour, analyse biologique, etc.).

Ainsi, le formalisme du langage $\mathcal{L}_{\text{DocCISMeF}}$ n'inclut pas pleinement d'éléments syntaxiques permettant de gérer la notion d'entité. Tous les codes champs désignent des attributs d'un type d'objet mais ce type d'objet est **implicitement** défini par l'application ou le contexte dans lequel ces codes champs sont employés et non directement exprimés au sein de la requête.

Par exemple, le type de d'objet renvoyé par DocCISMeF étant des ressources documentaires et celui de LiSSa des ressources bibliographiques, le code champ `.re` correspond à la description des ressources documentaires pour DocCISMeF alors qu'il désigne l'abstract d'une ressource bibliographique pour LiSSa. De même le code champ `.vol` ne renvoie rien pour DocCISMeF alors qu'il correspond au volume des ressources bibliographiques pour LiSSa. Le requêtage d'une telle contrainte ne s'effectue donc pas nécessairement de la même manière suivant que la requête a été soumise à l'application DocCISMeF ou LiSSa.

En somme, le langage $\mathcal{L}_{\text{DocCISMeF}}$ est basé sur une philosophie et un formalisme **orienté attribut**. Ce dernier, permet abstraitement de considérer que les différents codes champs correspondent à différents attributs (e.g. titre, auteur, date de publication, etc.) se rapportant tous à une même et unique entité définie implicitement (i.e. l'entité de sortie du système) et quelque soit la réalité de la modélisation concrète ou conceptuelle des données sous-jacentes. Ce paradigme est la principale différence avec le langage $\mathcal{L}_{\text{<}}$ qui lui est **orienté entité** et permet d'exprimer au sein des requêtes à la fois les attributs et les entités conceptuelles auxquelles ils se rapportent.

Afin d'assurer la rétro-compatibilitée du moteur de recherche SSE_{NoSQL} avec le langage de requête $\mathcal{L}_{\text{DocCISMeF}}$, un outil de conversion $\mathcal{L}_{\text{DocCISMeF}} \rightarrow \mathcal{L}_{\text{<}}$ a été développé. Ce dernier prend en paramètre une $\mathcal{L}_{\text{DocCISMeF}}$–requête mais également le type d'objet implicite définissant le contexte et permettant une « traduction » de la $\mathcal{L}_{\text{DocCISMeF}}$–requête en $\mathcal{L}_{\text{<}}$–requête cohérente. Une illustration de ce principe est donnée Figure 7.3 p. 175.

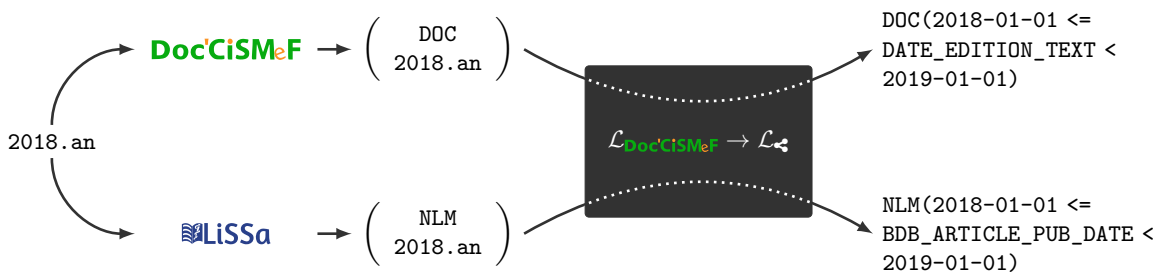


FIGURE 7.3 – Illustration « boîte noire » du convertisseur $\mathcal{L}_{\text{DocCISMeF}} \rightarrow \mathcal{L}_{\text{<}}$. La même requête `2018.an` est passée en entrée de DocCISMeF et LiSSa. Le convertisseur $\mathcal{L}_{\text{DocCISMeF}} \rightarrow \mathcal{L}_{\text{<}}$ est appelé avec la requête d'une part et le type d'entité attendu en sortie d'autre part (i.e. `DOC` pour DocCISMeF et `NLM` pour LiSSa). La $\mathcal{L}_{\text{<}}$–requête obtenue en sortie est alors différente.

7.1.2 Le langage $\mathcal{L}_{\mathbb{B}}$

Le langage $\mathcal{L}_{\mathbb{B}}$ est un langage de requête Booléen minimaliste. Il a été conçu afin de permettre l'écriture de requêtes Booléennes sans pour autant faire usage des codes champs du langage $\mathcal{L}_{\text{DocCiSMeF}}$ qui nécessitent une certaine maîtrise de la structuration de l'information sous-jacente. Tout comme le langage $\mathcal{L}_{\text{  }}$, ce langage est engendré par une grammaire formelle $\mathcal{G}_{\mathbb{B}}$ que j'ai implémentée dans le cadre de mon travail de thèse. La grammaire $\mathcal{G}_{\mathbb{B}}$ est explicitée par la définition 25 :

Définition 25 (La grammaire $\mathcal{G}_{\mathbb{B}}$) :
$\mathcal{G}_{\mathbb{B}}$ est la grammaire formelle définie par :

$$\mathcal{G}_{\mathbb{B}} = (T_{\mathbb{B}}, N_{\mathbb{B}}, \mathrm{Q}, R_{\mathbb{B}})$$

où :

l'ensemble des terminaux $T_{\mathbb{B}}$ *est constitué de l'ensemble des caractères qu'il est possible de saisir sur une machine dans un codage de caractère déterminé.*

l'ensemble des variables $N_{\mathbb{B}}$ *est défini comme l'ensemble :*

$$N_{\mathbb{B}} = \{Q, L, L_{\bullet}, L_{\circ}, W, \mathbb{B}\}$$

Le « rôle » ou la fonction syntaxique de ces variables est détaillé dans la Table 7.1

l'élément $Q \in N_{\mathbb{B}}$ *est l'axiome de* $\mathcal{G}_{\mathbb{B}}$.

l'ensemble des règles de production $R_{\mathbb{B}}$ *est constitué de l'ensemble des règles de production suivante :*

Ensemble $R_{\mathbb{B}}$ des règles de production			
1	Q	$\rightarrow$	$(L\ (\ \mathbb{B}\ Q\ )^* \mid$ `(` Q `)` $(\ \mathbb{B}\ Q\ )^*\ )$
2	L	$\rightarrow$	$(\ L_{\bullet} \mid L_{\circ}\ )$
3	$L_{\bullet}$	$\rightarrow$	`"` $L_{\circ}$ `"`
4	$L_{\circ}$	$\rightarrow$	$W\ (W)^*$
5	W	$\rightarrow$	$(C)^+$
6	$\mathbb{B}$	$\rightarrow$	(`AND` | `ET` | `OR` | `OU` | `NOT` | `SAUF`)
$\forall$ `c` $\in T_{\mathbb{B}} \setminus \{\sqcup,$ `(`, `)`$\}$	C	$\rightarrow$	`c`

Variable	Fonction/rôle syntaxique de la variable
Q	une $\mathcal{L}_{\mathbb{B}}$–requête.
L	un mot clé (i.e. un **L**ibellé) d'une $\mathcal{L}_{\mathbb{B}}$–requête.
$L_{\bullet}$	un mot clé entre guillemets.
$L_{\circ}$	un mot clé simple.
W	un mot d'un mot clé.
$\mathbb{B}$	un opérateur Booléen.
C	un caractère pouvant servir au sein d'un mot.

TABLE 7.1 – Fonction syntaxique jouée par chaque variable de la grammaire $\mathcal{G}_{\mathbb{B}}$ dans une $\mathcal{L}_{\mathbb{B}}$–requête.

Cette grammaire ne sera pas détaillée. Les $\mathcal{L}_{\mathbb{B}}$–requête ne sont cependant rien d'autre que des requêtes permettant de lier logiquement des libellés en langage naturel à l'aide d'expressions parenthésées et des opérateurs Booléens `ET`, `OU` et `SAUF`. Deux exemples de requêtes qu'il est possible de former à l'aide de ce langage sont données dans l'exemple 30.

Exemple 30 :

`myopathie des ceintures OU dystrophies musculaires` *sera interprétée comme une recherche visant à retrouver les ressources faisant état de « myopathie des ceintures » **ou bien** de « dystrophies musculaires ». Cette requête est différente de la requête `myopathie des ceintures dystrophies musculaires` qui n'effectue elle aucune distinction et tend à être interprété comme une conjonction de « myopathie », « ceintures », « dystrophies » et « musculaires ».*

`guide de bonnes pratiques ET asthme ET (enfant OU nourrisson)` *sera interprétée comme une recherche des « guides de bonne pratique » traitants de l'« asthme » chez l'« enfant » ou bien chez le « nourrisson ».*

L'implémentation de ce langage de requête a été effectuée avec la même stratégie que celle du langage $\mathcal{L}_{\text{share}}$. Le générateur JavaCC™ a été utilisé pour générer un parser$_{\mathbb{B}}$ qui permet de valider une $\mathcal{L}_{\mathbb{B}}$–requête et d'en construire une représentation sous forme d'un objet java BQuery.

Un convertisseur $\mathcal{L}_{\mathbb{B}} \rightarrow \mathcal{L}_{\text{DocCiSMeF}}$ a de plus été développé. Le langage $\mathcal{L}_{\mathbb{B}}$ est en effet destiné à une utilisation dans un contexte de RI documentaire et/ou bibliographique (Figure 7.2). Le langage $\mathcal{L}_{\text{DocCiSMeF}}$ sert donc de langage pivot au langage $\mathcal{L}_{\mathbb{B}}$ (langage de référence pour ce type de RI dans le SI) en vue d'une conversion en $\mathcal{L}_{\text{share}}$–requête. Le convertisseur $\mathcal{L}_{\mathbb{B}} \rightarrow \mathcal{L}_{\text{DocCiSMeF}}$ fait appel à l'annotateur sémantique ECMT pour convertir indépendamment chaque mot du langage naturel en « sous-$\mathcal{L}_{\text{DocCiSMeF}}$–requête ». Une illustration de ce principe est donnée Figure 7.4.

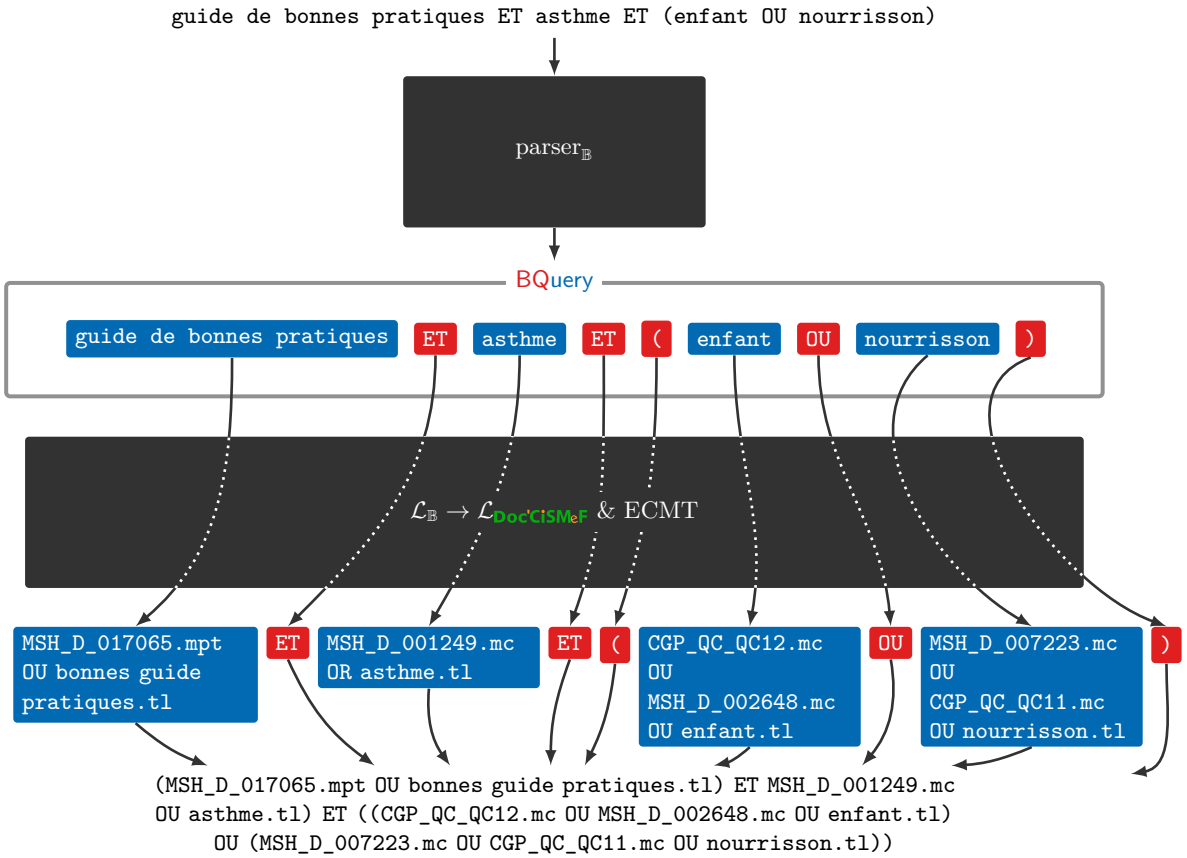


FIGURE 7.4 – Processus de transformation d'une $\mathcal{L}_{\mathbb{B}}$–requête en $\mathcal{L}_{\text{DocCiSMeF}}$–requête. La requête `guide de bonnes pratiques ET asthme ET (enfant OU nourrisson)` est d'abord transformée en un objet java à l'aide du parser$_{\mathbb{B}}$. Les mots clés sont analysés par le convertisseur $\mathcal{L}_{\mathbb{B}} \rightarrow \mathcal{L}_{\text{DocCiSMeF}}$ qui permet, à l'aide de l'annotateur sémantique ECMT de construire, pour chacun, une sous-$\mathcal{L}_{\text{DocCiSMeF}}$–requête.

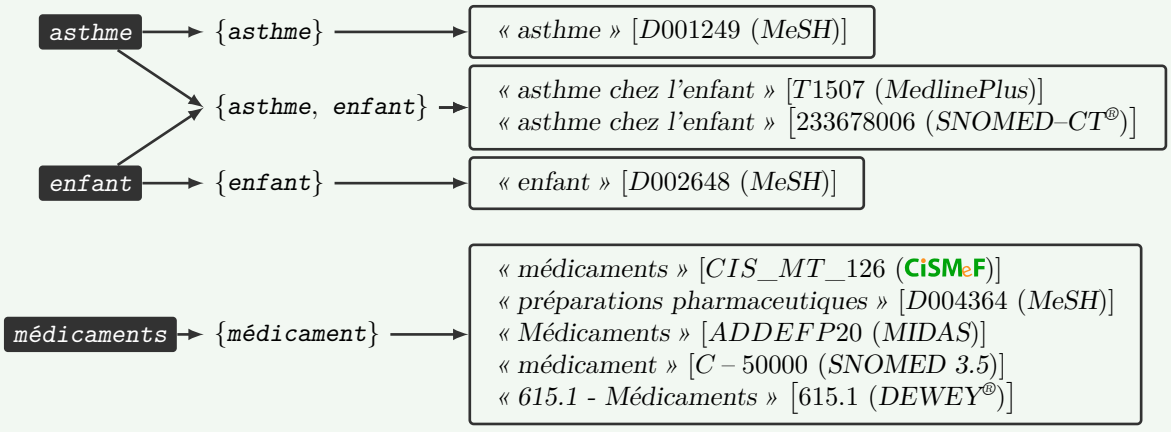
7.1.3 Le Langage Naturel

L'interrogation du moteur SSE_{NoSQL} en langage naturel est destiné à la RI dans le cadre de DocCISMeF et LiSSa (de même que les langages $\mathcal{L}_{\text{DocCISMeF}}$ et $\mathcal{L}_{\mathbb{B}}$). La RI au sein des données patient possède également un mode d'interrogation en langage naturel (via le langage $\mathcal{L}_{\text{puzzle}}$) cependant ce dernier est beaucoup moins abouti et « systématique ».

Tout comme pour le langage $\mathcal{L}_{\mathbb{B}}$, les requêtes en langage naturel sont traduites dans le langage pivot de la RI documentaire et bibliographique, c'est à dire en $\mathcal{L}_{\text{DocCISMeF}}$–requêtes. Cette transformation est effectuée grâce à l'annotateur sémantique ECMT [36]. Ce dernier repose sur le portail terminologique HeTOP fournissant un accès à plus de 70 TOs.
Le rôle premier de l'ECMT est d'**identifier**, ou plutôt de **reconnaître**, **au sein d'un texte en langage naturel, les mots et expressions correspondant à des concepts inclus dans** HeTOP. Techniquement, l'ECMT repose sur l'algorithme du sac de mots et inclut divers traitements de **T**raitement **A**utomatique du **L**angage (TAL) et notamment des opérations de normalisation, de tokénisation et de racinisation [4] (ou désuffixation).

Dans le contexte ici présenté, le texte fourni est de longueur réduite puisqu'il s'agit d'une requête. Une fois que les concepts du portail HeTOP présents dans cette requête sont identifiés, ECMT propose un algorithme permettant de constituer une $\mathcal{L}_{\text{DocCISMeF}}$–requête à partir de l'ensemble des concepts identifiés. Ce dernier fonctionne de manière assez basique en essayant de constituer une $\mathcal{L}_{\text{DocCISMeF}}$–requête qui couvre logiquement la totalité des mots de la requête. Un exemple est donné ci-dessous afin d'illustrer le principe de formation d'une $\mathcal{L}_{\text{DocCISMeF}}$–requête.

Exemple 31 :
Soit la requête `asthme chez l'enfant médicament`*. Après normalisation, tokénisation et racinisation, cette requête est interprété par l'ECMT comme une requête composée de trois mots : «* `asthme` *», «* `enfant` *» et «* `médicament` *». L'ECMT permet d'identifier plusieurs concepts médicaux à l'aide de différentes combinaisons de ces trois mots :*



L'algorithme dédié à la constitution d'une $\mathcal{L}_{\text{DocCISMeF}}$–requête agrège alors ces concepts au sein d'une requête de telle sorte que la totalité des trois mots « `asthme` *», «* `enfant` *» et «* `médicaments` *» soient couverts :*

$$\{\texttt{asthme}, \texttt{enfant}, \texttt{médicament}\} = \begin{cases} \{\texttt{asthme}\} \cup \{\texttt{enfant}\} \cup \{\texttt{medicament}\} \\ ou \\ \{\texttt{asthme,enfant}\} \cup \{\texttt{médicament}\} \end{cases}$$

Ainsi l'ECMT fournit la $\mathcal{L}_{\text{DocCISMeF}}$–requête :

(

4. : « stemming »

```
      (MSH_D_001249.mc OU asthme.ti)                    ⇔ {asthme}
      ET (MSH_D_002648.mc OU enfant.ti)                 ⇔ {enfant}
      (MID_CO_ADDEFP20.mc OU SNO_NO_C-50000.mc          ⇔ {médicament}
      OU MSH_D_004364.mc OU CIS_MT_126.mt
      OU DEW_CD_615.1.mc OU medicament.ti)
      ET (chez.ti)
)
OU                                                      ou
(
      (MED_T_T1507.mc OU SCT_CO_233678006.mc            ⇔ {asthme,enfant}
      OU asthme chez enfant.ti)
      (MID_CO_ADDEFP20.mc OU SNO_NO_C-50000.mc          ⇔ {médicament}
      OU MSH_D_004364.mc OU CIS_MT_126.mt
      OU DEW_CD_615.1.mc OU medicament.ti)
)
```

7.1.4 Le langage $\mathcal{L}_{\#}$

Dans le cadre de la RI au sein des données patient, le langage de requête de base disponible est le langage $\mathcal{L}_{\leq}$. Ce dernier reste cependant **difficile à exploiter**. Ce dernier requiert en effet à la fois une certaine formation afin d'en **maîtriser la syntaxe** mais aussi une **bonne connaissance de l'organisation conceptuelle des données**. Même si les professionnels de santé tendent à avoir une certaine intuition de cette structuration des données (celle-ci dérivant majoritairement du processus de prise en charge des patients), ils n'ont en revanche pas, dans le cadre de leur travail, le temps de se former à une nouvelle syntaxe.

Afin de pallier à cette difficulté, il a été imaginé dans le cadre de cette thèse une **méthode d'interrogation en langage naturel**.

Les outils de RI classiques proposent généralement de retourner un ensemble de documents à l'aide d'une requête composée d'une séquence de mots-clés. Ce procédé repose principalement sur la recherche d'une correspondance partielle ou exacte entre les mots-clés de la requête et les méta-données ou les concepts indexant les documents en question. Cette approche est cependant peu adaptée à la RI au sein des données patient puisque la sémantique inhérente à ces données est plus « sophistiquée » et qu'une simple séquence de mot-clés est généralement insuffisante pour l'exprimer.

La méthode décrite dans cette section permet, quant à elle, de **réécrire des requêtes en langage naturel en $\mathcal{L}_{\leq}$–requête**. Pour se faire, cette méthode **identifie récursivement des « patrons** [5] **» ou « motifs »** de requête en langage naturel pouvant par la suite être transposés génériquement en $\mathcal{L}_{\leq}$–requête. L'ensemble des requêtes en langage naturel pouvant être « reconnues » (ou réécrites) est ainsi limité et pré-établi. Il est alors possible de considérer que ces requêtes particulières constituent un langage noté $\mathcal{L}_{\#}$ en tant que partie de l'ensemble de toutes les requêtes en langage naturel possible.

Le langage $\mathcal{L}_{\#}$ n'est cependant pas défini dans la pratique comme un langage engendré par une grammaire formelle comme c'est le cas pour les langages, $\mathcal{L}_{\leq}$, $\mathcal{L}_{\text{DocCiSMeF}}$ ou $\mathcal{L}_{\mathbb{B}}$. La définition de ce langage passe par l'exploitation d'une méthode d'**extraction d'information**. Ce type de méthode consiste à extraire des types d'information pré-définis à partir d'un texte. Il existe cependant quatre courants majeurs d'extraction d'information exploitant en réalité différentes techniques :

— Des techniques de TAL ;

— Des techniques exploitant des règles ;

5. : « patterns »

— Des techniques d'apprentissage automatique notamment à l'aide de Classifieur ;

— Des techniques de filtrage par motif plus connues sous le terme anglais correspondant « pattern-matching ».

La méthode proposée dans cette section est une méthode reposant sur des techniques de filtrage par motif combiné avec du TAL basique.

7.1.4.1 Principe de base

La Figure 6.2 représente les données de patients sous forme d'un graphe de données à priori intelligible pour les professionnels de santé. En effet, cette organisation de l'information est issue du processus de prise en charge des patients au sein des établissements de santé dans lesquels ils évoluent professionnellement. Les entités présentes dans le graphe (e.g. patients, séjours, actes médicaux, etc.) ainsi que les relations sémantiques qu'entretiennent ces entités entre elles sont par conséquent potentiellement familières à ces professionnels de santé. Une formulation sous forme d'une requête en langage naturel d'un besoin d'information basé sur ces données fait donc généralement intervenir ces notions (i.e. les entités et leurs relations).

Considérons, par exemple, la requête suivante :

« *tous les patients homme ayant effectué un séjour de 10 jours dans l'unité de cardiologie* »

Cette requête, formulée en langage naturel, constitue clairement à une question restant à la portée d'un professionnel de santé et susceptible d'être formulée par ces derniers. Pour autant, la mise en correspondance de cette requête avec le modèle de données de la Figure 6.2 et assez aisée comme il l'est objectivé ci-dessous :

« ***patient*** » correspond à l'entité `patient` de la Figure 6.2.

« ***homme*** » correspond à son attribut `sexe`.

« ***séjour*** » correspond à l'entité `sejour` de la Figure 6.2.

« ***10 jours*** » peut être exprimé à l'aide des attributs `date_entree` et `date_sortie` de cette entité.

« ***patient [...] ayant effectué un sejour*** » correspond à la relation `a_sejour` entre l'entité `patient` et l'entité `sejour`.

« ***unité*** » correspond à l'entité `unite_medical` de la Figure 6.2.

« ***cardiologie*** » correspond au libellé `label` de cette entité.

« ***un sejour [...] dans l'unité [...]*** » correspond à la relation `dans_UM` entre l'entité `sejour` et l'entité `unite_medicale`.

Le principe de base de la méthode décrite ici est de définir des **expressions régulières** permettant d'identifier et de tagger [6] les différents mots et/ou expressions afin de les mettre en correspondance avec le MCD sous-jacent aux données (dans le cas présent le MCD est défini par la Figure 6.2).

Il est à noter que cette méthode a été développée dans le cadre du SSE_{SQL}. Les exemples donnés dans toute cette section sont donc basés sur le MCD de la Figure 6.2 [7].

6. : « annoter »

7. Le MCD aujourd'hui utilisé dans le cadre du SSE_{NoSQL} diffère en revanche légèrement de ce dernier et est susceptible de subir encore davantage de modifications à l'avenir avec le projet de l'EDSS du CHU de Rouen

7.1.4.2 La méthode

Le processus de cette méthode se décompose globalement en trois étapes qui sont les suivantes :

1. le système accepte en entrée une requête q en langage naturel ;
2. la requête q est traitée par un **tagger** dont le rôle est d'identifier la structure de celle-ci en attribuant aux différents mots et expressions qui la compose, des **tags**[8] génériques désignant leurs **rôles structurels** vis à vis du MCD sous-jacent ;
3. les tags génériques identifiés et les libellés concrets qu'ils étiquettent sont alors utilisés pour construire une $\mathcal{L}_{\text{share}}$–requête.

Le tagger repose sur cinq itérations. Chacune de ces itérations consiste en une reconnaissance de patron qui est réalisée grâce à un ensemble d'expressions régulières prédéfinie (cet ensemble étant bien entendu différent pour chaque itération).

Ces expressions régulières visent à identifier au sein de la requête q des catégories spécifiques d'éléments ou des structures grammaticales pré-définies afin de les étiqueter. Les différentes itérations du tagger sont, de plus, récursives de telle sorte qu'une itération $n+1$ exploite les tags identifiés à l'itération n.

Ainsi, les itérations identifient des structures de plus en plus génériques à mesure que les itérations se succèdent. Les patrons identifiés par l'itération n° 5 sont donc les plus génériques et abstraits tandis que les patrons reconnus par l'itération n° 1 sont eux, très concrets, pragmatiques et très proche de la requête q elle-même.

Parmi les cinq itérations, les trois premières peuvent être considérées comme relativement basiques comparées aux deux suivantes.

Les itérations n° 1, n° 2 et n° 3 permettent simplement de tagger des éléments clés de la requête q.

Les itérations n° 4 et n° 5 s'attachent elles, davantage à identifier la structure de la requête q qui permet de lier ces éléments clés entre eux.

Afin de détailler le rôle de chaque itération dans les explications suivantes, on s'appuiera sur l'exemple de la requête q suivante :

« *Analyse biologique de polynucléaires neutrophiles supérieure à la normale du patient 71* »

La Figure 7.5 donne une représentation graphique de tout le processus du tagger appliqué à cette requête et donc de expression en lagage naturel reconnues au sein de cette requête tout au long des cinq itérations.

Une description textuelle des rôles de chacunes de ces cinq itérations est également fournie.

8. : « étiquettes »

Définition 26 (Itération n° 1 « Tagging d'entité ») :
Il a pour but d'identifier les différentes entités du MCD sous-jacent auxquelles la requête q fait référence. Par exemple, dans la Figure 7.5 l'expression « Analyse Biologique » et le mot « patient » font référence respectivement aux entités `analyse_biologique` *et* `patient` *du MCD de la Figure 6.2.*

Définition 27 (Itération n° 2 « Tagging de concept ») :
Il consiste en une indexation multi-terminologique d'entité. Il vise à identifier des concepts terminologiques au sein de la requête q. Ces derniers peuvent définir par exemple de types de diagnostics grâce à des code CIM–10 (indexation d'une entité entité de type diagnostic), des types d'actes médicaux avec des codes CCAM ou encore, comme dans le cas présent, des types d'analyses biologiques (e.g. Sodium, polynucléaires neutrophiles, Potassium, etc.). Le tagging de concept est réalisé par l'intermédiaire de l'annotateur sémantique ECMT et constitue l'essentiel de l'utilisation de méthode de TAL évoquée précédemment.

Définition 28 (Itération n° 3 « Tagging de contraintes ») :
Cette itération permet d'analyser des expressions plus complexes qui correspondent à des contraintes sur les entités identifiées dans l'itération n° 1. Ces contraintes reposent principalement sur les attributs de ces entités. par exemple dans une requête du type « […]patient homme[…] », le mot « homme » fait référence à l'attribut `sexe` *de l'entité* `patient`*. De même, les attributs de type date des entités* `analyse_biologique` *et* `sejour` *peuvent modéliser des caractéristiques telles que la date d'une analyse biologique ou la durée d'un séjour. Dans l'exemple de la Figure 7.5, « supérieure à la normale » correspond aux attributs* `valeur`*,* `borneInf` *et* `borneSup` *de l'entité* `analyse_biologique` *identifiée à l'itération n° 1 et « 71 » correspond à l'attribut* `id` *de l'entité* `patient`*. Cependant, une contrainte peut également correspondre à un codage. Par exemple, dans la Figure 7.5, l'expression « de polynucléaires neutrophiles » est taggée comme une contrainte de codage de l'entité* `analyse_biologique` *identifiée à l'itération n° 1.*

Définition 29 (Itération n° 4 « Tagging de phrase entité ») :
À partir de cette itération, toutes les entités et contraintes ont été mises en évidence (i.e. taggées). Une phrase entité correspond à une sous phrase de la requête initiale concernant une seule entité. Le tagging de phrase entité permet donc de rattacher logiquement les contraintes identifiées par l'itération n° 3 aux entités auxquelles elles se rapportent qui ont, elles, été identifiées à l'itération n° 1. Cette étape regroupe donc de manière cohérente divers tags qui se doivent d'être considérés comme un ensemble.

Définition 30 (Itération n° 5 « Tagging d'interconnexions ») :
Le tagging d'interconnexions met en évidence les liens existants entre les différentes entités référencées dans la requête q. Ceci est rendu possible par la reconnaissance de la construction grammaticale liant les différentes phrases entité identifiées à l'itération précédente. Dans l'exemple de la Figure 7.5, le mot « du » entre la phrase entité relative à l'analyse biologique et celle relative au patient, permet de mettre en évidence la relation existante entre ces deux phrases entité. En l'occurrence ici la requête dans son ensemble requiert des analyses biologiques du patient.

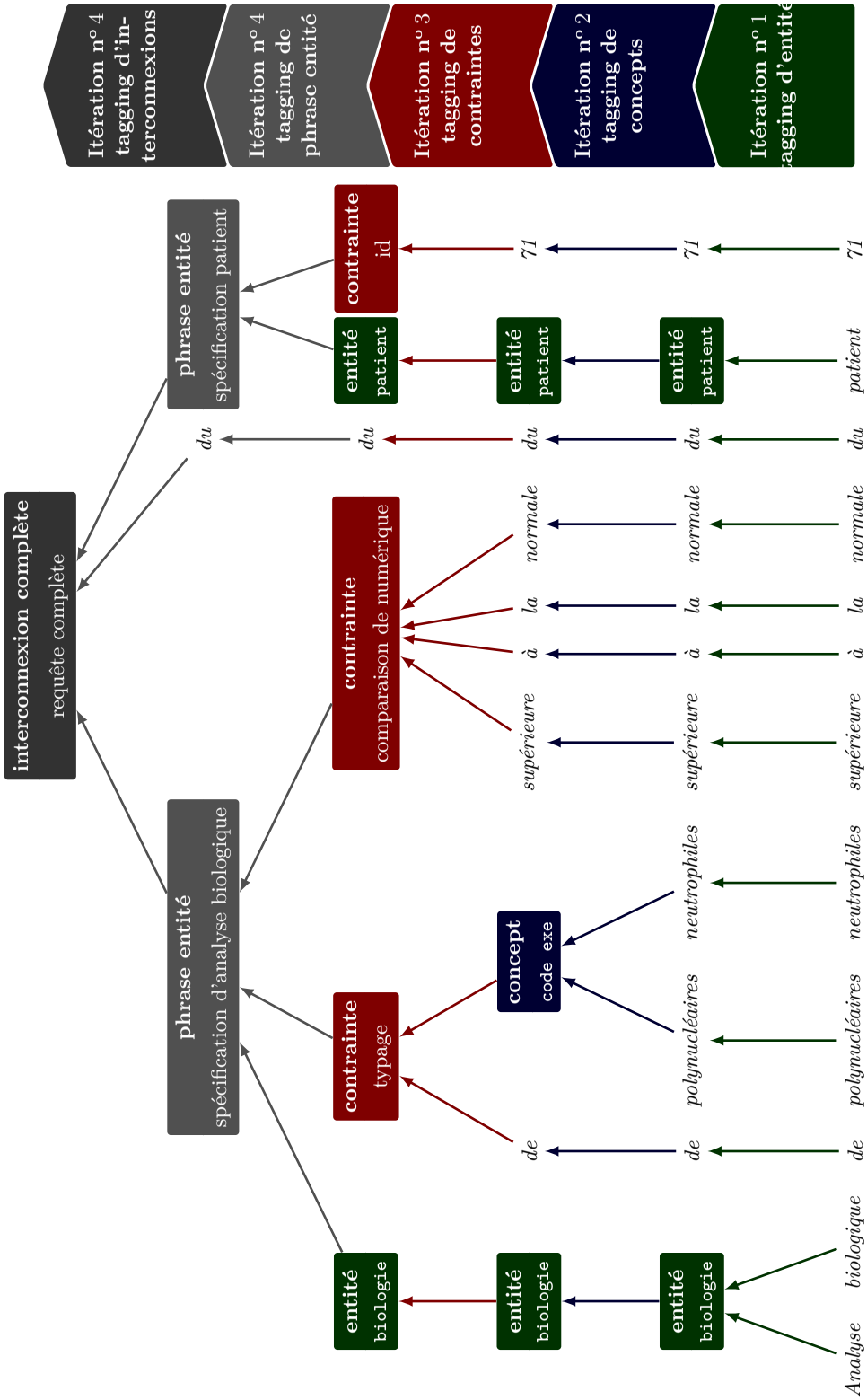


FIGURE 7.5 – Représentation graphique des différentes phases de réécriture de la requête « *Analyse biologique de polynucléaires neutrophiles supérieure à la normale du patient 71* » effectuée par le tagger. Chaque itération (hormis la première) consomme les tags identifiés par la ou les itérations précédentes afin d'identifier de nouveaux patrons plus complexes.

7.1.4.3 Génération de la $\mathcal{L}_{\text{<}}$–requête

Chaque tag identifié par le tagger est à-même de générer une partie de la $\mathcal{L}_{\text{<}}$–requête finale. Un tag identifié récursivement à l'aide de tags ayant été généré par des itérations précédentes construit ainsi sa propre $\mathcal{L}_{\text{<}}$–requête en fusionnant les requête issues de ces précédents tags.

Par exemple, la $\mathcal{L}_{\text{<}}$–requête du tag correspondant à la phrase entité relative au patient est la requête `patient(id="DM_PAT_71")`. Elle est construite en fusionnant la requête `patient()` et la requête `id="patient"` générées respectivement par le tag d'entité issu du mot « *patient* » et le tag de contrainte issu du mot « *71* ».

La génération de la $\mathcal{L}_{\text{<}}$–requête finale est donc réalisée également de manière récursive. La Figure 7.6 illustre cette génération récursive de la $\mathcal{L}_{\text{<}}$–requête correspondant à la même requête en langage naturel que précédemment.

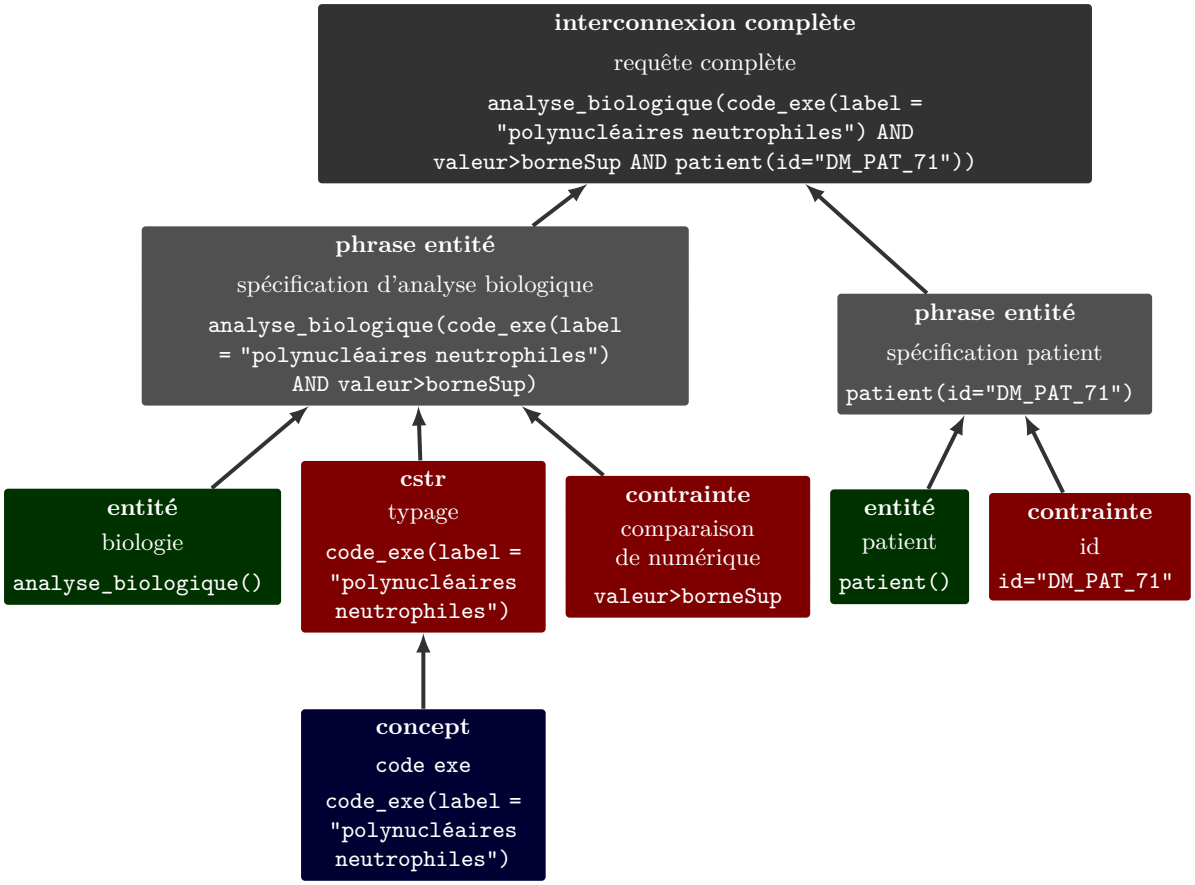


FIGURE 7.6 – Représentation de la génération récursive de la $\mathcal{L}_{\text{<}}$–requête correspondant à la requête « *Analyse biologique de polynucléaires neutrophiles supérieure à la normale du patient 71* ».

7.1.4.4 Dans la pratique

La méthode décrite dans cette section a été, dans la pratique, implémentée afin de fonctionner avec le SSE_{SQL}. Une preuve de concept en a notamment été développé et évaluée [157] sur sa capacité à répondre à un certain nombre de cas d'usages du projet RAVEL [ANR–11–TECS–012].

Les requêtes en langage naturel utilisées dans le cadre de cette étude sont données dans la Table 7.3. La Table 7.2 indique le nombre d'expressions régulières qu'il a été nécessaire d'écrire afin d'implémenter cette preuve de concept.

Types de tagging	**nombre de patrons implémentés**
tagging d'entités	25
tagging de contraintes	34
tagging de phrases entité	24
tagging d'interconnexions	134

TABLE 7.2 – Nombre de patterns (expression régulière) manuellement écrites et disponibles pour chacune des cinq itérations du tagger. Pas de tagging de concept (utilisation de l'ECMT).

Requêtes en langage naturel	t **(ms)**
Les patients 44 et 45	188
Les patients 44, 45 et 18	171
Les séjours des patients 44 et 45	94
Les patients avec une polyarthrite rhumatoïde séropositive	250
Analyses de sodium supérieures à la normale	296
Analyses de Polynucléaires neutrophiles du patient d'id 71	375
Potassium du patient 104	141
Calcium ou glucose anormal du patient 4	484
Analyses de calcium ou glucose du patient 4	172
Analyses de calcium <6 du patient 4	203
Analyses de calcium <2.5 du patient 4	188
Analyses de Polynucléaires neutrophiles supérieures à la normale du patient 71	234
Analyses de Polynucléaires neutrophiles <= norm du patient 1902	266
Analyses de calcium ou glucose anormale du patient 4	281
Analyses de calcium ou glucose dont le résultat est dans la norme du patient 4	813
Analyses de sodium et de calcium et de glucose du patient d'id 44	265
Les analyses de glucose des patients de sexe féminin de plus de 100 ans	344
Les séjours avec une prescription de 9213737	109
Les analyses de glucose des patients hommes de plus de 35 ans	235
Les séjours de 2010 du patient 45	109
Les séjours de mars 2010 de patients hommes de plus de 60 ans	250
Les séjours de mars 2010 d'une durée de plus de 5j des patients hommes de plus de 60 ans	344
Les séjours de mars 2010 d'une durée de plus de 5j des patients de plus de 95 ans	312
Séjours du patient 140	94
Patients avec le AIDS	110
Affiche analyses de calcium ou glucose du patient 4	234
Dernier compte rendu d'imagerie médicale	156
Patients avec Remicade	172
Analyses d'hypocalcémie du patient 4	141
Hyperkaliémie du patient 104	140
Patient âgé de plus de 70 ans	141
Hypercalcémie du patient 4	141
Séjours pour un acte de radiographie du thorax	469
Les patients avec une hypertension artérielle	219
Les séjours de mars 2010 avec une prescription de 9213737	203

TABLE 7.3 – Requêtes et types de requêtes en langage naturel issus des cas d'usages du projet RAVEL [ANR–11–TECS–012]. Pour chaque requête le temps d'exécution moyen t de ces dernières sur le SSE_{SQL} est indiqué.

Bien que les $\mathcal{L}_{\text{}}$–requêtes générées par cette méthode soient interprétables par le SSE_{NoSQL}, la preuve de concept n'a pas été ré-adaptée au SSE_{NoSQL}. Bien que cela soit parfaitement possible, la méthode sous-jacente fait l'objet de plusieurs limitations. Elle manque notamment de flexibilité à deux niveaux :

Flexibilité dans l'écriture des patterns : le système permet de réécrire des requêtes en langage naturel en $\mathcal{L}_{\text{}}$–requête. Cependant, cette réécriture ne permet aucun écart avec la syntaxe initialement prévue et toutes les syntaxes possibles doivent être prévues et faire l'objet de l'adaptation des expressions régulières appropriées. Par exemple, si un certain nombre d'expressions régulières visant à permettre la réécriture des requêtes du type « *patient possédant un diagnostic de ...* » n'ont pas été définies de manière assez générique alors le système ne sera pas en capacité de réécrire les requêtes du type « *patient avec un diagnostic de ...* ».

Flexibilité vis à vis du MCD sous-jacent : Le système établit une réécriture des requêtes en $\mathcal{L}_{\text{}}$–requêtes. Les $\mathcal{L}_{\text{}}$–requêtes sont syntaxiquement basées sur le MCD sous-jacent. En revanche la méthode de réécriture est elle, indépendante de ce dernier et n'en a aucune « connaissance ». Ainsi les changements de structuration conceptuelle des données ne sont donc pas répercutés sur les requêtes générées par le système. Ce manque de flexibilité est la principale raison pour laquelle la preuve de concepts n'a pas été adaptée au SSE_{NoSQL} qui repose sur un MCD sous-jacent encore aujourd'hui en « mouvement ».

7.2 Logique interne

Comme déjà précisé précédemment, les moteurs SSE_{SQL} et SSE_{NoSQL} reposent tous deux sur des SGBDs différents. Leurs logiques d'exécution des requêtes diffèrent ainsi également. Même si l'interrogation peut être effectuée en langage naturel ou éventuellement par l'intermédiaire des langages de requête logique $\mathcal{L}_{\text{DocCiSMaF}}$, $\mathcal{L}_{\mathbb{B}}$ ou $\mathcal{L}_{\#}$. Le langage de requête $\mathcal{L}_{\text{share}}$ reste le langage de requête pivot de ces deux moteurs et le seul réellement exécutable par ces deux moteurs.

On s'intéressera dans cette section à la logique interne d'exécution des $\mathcal{L}_{\text{share}}$–requêtes pour chacun des moteurs SSE_{SQL} et SSE_{NoSQL}. La structure arborescente exploitée pour représenter l'intégralité de la structure d'une $\mathcal{L}_{\text{share}}$–requête est, dans un premier temps, présentée. Cette dernière est commune aux deux moteurs mais est exploitée de manière différente. La logique d'éxecution de cette structure par les deux moteurs SSE_{SQL} et SSE_{NoSQL} est alors présentée dans un second temps.

7.2.1 Structure arborescente

Les langages $\mathcal{L}_{\text{DocCiSMaF}}$, $\mathcal{L}_{\mathbb{B}}$ et $\mathcal{L}_{\text{share}}$ sont tous les trois des langages de requête logiques dans le sens où ils sont Booléens. Cependant, le langage de requête $\mathcal{L}_{\text{share}}$ est plus complexe que les deux autres. Ce dernier possède en effet, non seulement une structure Booléenne, mais également une structure sémantique obtenue par imbrications récursives de clauses entité (i.e. contraintes sémantiques).

Les $\mathcal{L}_{\text{share}}$–requêtes obtenues en sortie du parser$_{\text{share}}$ sont cependant assimilables à de **simples listes de tokens** qui ne permettent pas de rendre compte de leurs structures complexes. Ces dernières permettent en effet d'identifier les unités syntaxiques de base du langage $\mathcal{L}_{\text{share}}$ mais ne permettent pas d'identifier les dépendances logiques existantes entre ces différentes unités syntaxiques.

Afin de permettre l'exécution des $\mathcal{L}_{\text{share}}$–requêtes, un **analyseur** a été conçu. Ce dernier permet de construire une structure arborescente représentative de tous les aspects structurels d'une $\mathcal{L}_{\text{share}}$–requête (i.e. la structure Booléenne et la structure sémantique). En d'autres termes, cet analyseur permet de **« transformer » une $\mathcal{L}_{\text{share}}$–requête en un arbre binaire** (i.e. un arbre dans lequel tous les nœuds possèdent au plus deux nœuds fils) permettant non seulement de représenter la logique propre de la $\mathcal{L}_{\text{share}}$–requête mais aussi de naviguer au sein de cette dernière. Dans la suite de ce rapport on appellera $\mathcal{L}_{\text{share}}$–arbre tout **Arbre** Binaire d'une $\mathcal{L}_{\text{share}}$–requête.

On prendra dans cette section la $\mathcal{L}_{\text{share}}$–requête suivante comme exemple support d'explication :

```
PATIENTS(
     PATIENTS.SEXE="1"
     ET SEJOURS(
          EDS_ANA[FILTERS="IND_EDS_ANA_EDS_TYPE_ANA#EDS_BIO_KDD"](
               EDS_ANA.RES>=4)
          ET DIAGNOSTICS(
               T_DESC_ICD10_CATEGORY[EXPL="DOWN"]{IND_PMSI}(
                    id="ICD_CA_I50"))
          ET LIBUMREL_UF_LIBUM(
               id="LIBUM.REAC")
          ET RECORD(
               T_DESC_ATC_CODE[EXPL="DOWN"]{IND_AUTO_RECORD_DESCRIPTOR}(
                    id="ATC_CD_C03CA01")
               OU T_DESC_PHARMA_RACINE[EXPL="DOWN"]{IND_AUTO_RECORD_DESCRIPTOR}(
                    id="PHA_RAC_3549","PHA_RAC_4617","PHA_RAC_4620")
               OU T_DESC_PHARMA_DCI[EXPL="DOWN"]{IND_AUTO_RECORD_DESCRIPTOR}(
                    id="PHA_DCI_410"))))
```

Cette dernière permet de récupérer l'ensemble des patients hommes ayant effectué un séjour dans l'unité médicale de réanimation pour lequel :

— un diagnostic d'insuffisance cardiaque a été posé. Ceci est réalisé à l'aide d'une recherche des séjours codés avec les concepts fils du code « *insuffisance cardiaque* » [$I50$ (CIM–10)] ;

— au moins une analyse biologique de potassium s'est révélée être supérieure à 4 mmol/L ;

— le ou les compte(s)-rendu(s) de ce dernier sont susceptible(s) de mentionner un traitement visant à faire baisser la kaliémie du patient. Ceci est réalisé par une recherche des comptes-rendus indexés automatiquement avec les concepts suivants issus des deux terminologies **A**natomical **T**herapeutic **C**hemical classification (ATC) [9] et Médicaments (PHA) [10] :

Code ATC : « *furosémide* » [*C*03*CA*01 (ATC)]

Code de Dénomination Commune Internationale (DCI) de la PHA : « *furosémide* » [410 (PHA)]

Code de racine pharmacologique de la PHA :

— « *FUROSEMIDE* » [3549 (PHA)]

— « *LASILIX* » [4617 (PHA)]

— « *LASILIX SPECIAL* » [4620 (PHA)]

Cette requête est inspirée du type de requête qu'il est possible de construire avec l'outil **ASIS** développé au sein du **D2IM** depuis 2017. Cette interface permet d'effectuer des recherches au sein des données patients du CHU de Rouen et exploite le SSE_{NoSQL} comme moteur de recherche.

Comme toutes $\mathcal{L}_{\text{<}}$–requêtes, la syntaxe de cette requête basée sur le MCD structurant conceptuellement les données patients au sein de l'EDS. Les différentes entités, relations et attributs apparaissant dans cette requête ne diffèrent malgré tout que très peu structurellement par rapport à celui de la Figure 6.2. Il est donc possible, dans un souci de clarté et de compréhension, de fournir une requête équivalente, écrite dans le MCD de la Figure 6.2. Cette adaptation n'ayant en outre pas d'incidence sur les explications qui seront données par la suite.

Pour plus de lisibilité, cette requête est donnée indentée et colorée (viz. `bleu` pour les entités, `noir` pour les options de jointure et de chemin, `rouge` pour les opérateurs Booléens et `vert` pour les contraintes d'attribut).

Exemple 32 :

```
patient(
      sexe="1"
      ET sejour(
            analyse_biologique[FILTERS="IND_EDS_ANA_EDS_TYPE_ANA#EDS_BIO_KDD"](
                  valeur>=4)
            ET diag(
                  cim10[EXPL="DOWN"]{IND_PMSI}(
                        id="ICD_CA_I50"))
            ET unite_medicale{REL_UF_LIBUM}(
                  id="LIBUM.REAC")
            ET comte_rendu(
                  atc[EXPL="DOWN"]{IND_AUTO_RECORD_DESCRIPTOR}(
                        id="ATC_CD_C03CA01")
                  OU racine[EXPL="DOWN"]{IND_AUTO_RECORD_DESCRIPTOR}(
                        id="PHA_RAC_3549","PHA_RAC_4617","PHA_RAC_4620")
                  OU dci[EXPL="DOWN"]{IND_AUTO_RECORD_DESCRIPTOR}(
                        id="PHA_DCI_410"))))
```

Cette dernière fait apparaître les entités `patient`, `compte_rendu`, `sejour`, `unite_medicale` et `analyse_biologique` qui sont présentes dans le MCD de la Figure 6.2. La nouvelle modélisation conceptuelle des données patients introduit une nouvelle entité permettant de modéliser un diagnostic noté ici `diag`. Ainsi les codes CIM–10 permettant de définir le diagnostic d'un séjour ne sont plus rattachés directement au séjour mais par l'intermédiaire d'une entité de type `diag`

9. url : `https://www.whocc.no/atc_ddd_index/`
10. Codes construits à partir de la base de données publique des médicaments et/ou MedicaBase et/ou VIDAL

qui est, elle, rattachée à l'entité `sejour`. Enfin les entités `racine`, `dci`, `atc` et `cim10Cat` sont les libellés utilisés ici pour désigner respectivement les types de concepts racine pharmacologique de la PHA, code DCI de la PHA, les code ATC et les catégories CIM–10.

La Figure 7.7 donne une représentation de l'arbre généré par l'analyseur pour la requête de l'exemple 32 p. 188.

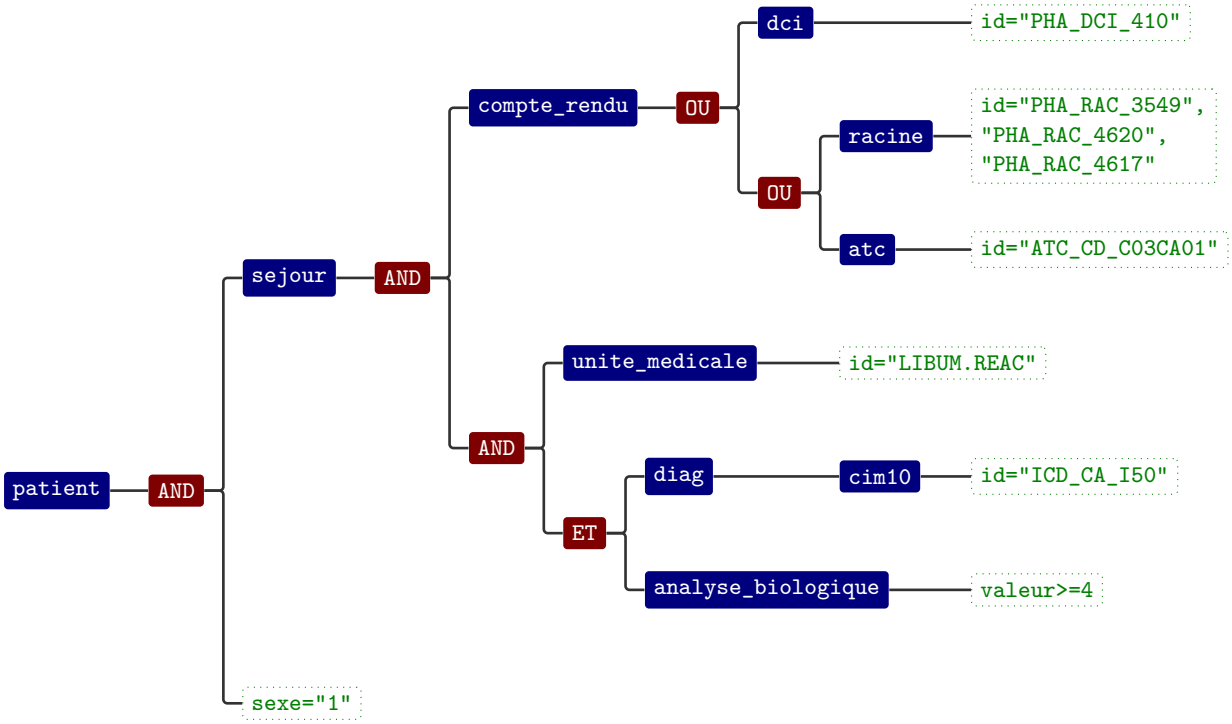


FIGURE 7.7 – Représentation de la structure arborescente de la $\mathcal{L}_{\text{<}}$–requête de l'exemple 32 utilisé en interne du moteur de recherche SSE_{NoSQL} et SSE_{SQL} pour représenter et exécuter cette dernière. Bien que contenus au sein des nœuds d'entité, les contraintes de jointure et de chemins ne sont pas représentés ici pour plus de lisibilité.

Ce dernier est un arbre binaire non entier. Les nœuds de cet arbre peuvent en effet ne pas avoir de nœud fils (feuille de l'arbre) ou en avoir deux mais aussi n'avoir qu'un seul nœud fils. Relativement au langage $\mathcal{L}_{\text{<}}$, tous les nœuds représentent soit un token de la $\mathcal{L}_{\text{<}}$–requête (i.e. une unité syntaxique du langage $\mathcal{L}_{\text{<}}$) construite par le parser$_{\text{<}}$, soit un ensemble de ces tokens. Plus précisément un nœud peut représenter :

Une déclaration d'une clause entité : Ce nœud contient alors l'ensemble des tokens de la $\mathcal{L}_{\text{<}}$–requête relatif à l'entité ciblée ainsi que des contraintes de jointure et de chemin qui lui sont rattachées. Un tel nœud peut éventuellement posséder un nœud fils représentant alors la clause de contrainte de la clause entité ou ne posséder aucun nœud fils et constituer un nœud feuille de l'arbre lorsque la clause entité ne possède pas de clause de contrainte. Ce type de nœud permet notamment la représentation de la structure sémantique de la $\mathcal{L}_{\text{<}}$–requête en permettant de rendre compte de l'imbrication des clauses entité en tant que contrainte d'une clause entité mère.

Un opérateur Booléen `ET` ou `OU` : Dans ce cas le nœud possède exactement deux nœuds fils représentant les deux contraintes liées logiquement par l'opérateur Booléen en question. Ce type de nœud constitue la méthode de base permettant une représentation de la logique Booléenne de la $\mathcal{L}_{\text{<}}$–requête. Les liens existants entre les différents nœuds sont établis dans le respect des règles de priorité des opérateurs Booléens et du parenthésage.

Une contrainte d'attribut d'une clause entité : Dans ce cas le nœud ne possède aucun nœud fils. Ces nœuds correspondent alors systématiquement à des feuilles de l'arbre.

7.2.2 Le Semantic Search Engine SQL (SSE$_{SQL}$)

Lorsque j'ai implémenté le SSE$_{SQL}$, l'objectif sous-jacent était de l'utiliser dans le cadre du projet RAVEL [ANR–11–TECS–012] afin de répondre aux cas d'usages définis dans ce dernier. Le SSE$_{SQL}$ constitue donc la première mise en application concrète du langage de requête que j'ai modélisé dans le cadre de ma thèse.

D'un point de vu opérationnel, le SSE$_{SQL}$ a notamment été intégré comme moteur de recherche interne de l'application **R**echerche d'**I**nformation dans le **Do**ssier **P**atient **I**nformatisé (RIDoPI) dont le but est de fournir des fonctionnalités de visualisation et de navigation au sein du DPI. Aujourd'hui, le D2IM travaille au développement de l'interface ASIS (cf. section 8.1). Cette dernière vise à fournir des fonctionnalités similaires dans le cadre plus général de l'EDSS et ne repose non pas sur le SSE$_{SQL}$ mais sur le SSE$_{NoSQL}$.

Les données mises à disposition par RIDoPI sont celles du corpus 2 000 que j'ai précédemment décrit. Bien que réduites du point de vue de la volumétrie, ces données ont néanmoins l'avantage d'avoir été sélectionnées manuellement par un professionnel de santé et de présenter une « densité d'information » propice à la construction d'une preuve de concept. Comme décrit dans le chapitre 5 p. 111, la tentative de passage à l'échelle du SSE$_{SQL}$ avec le corpus 60 000 a cependant conduit à l'abandonner en raison des performances insuffisantes que ce dernier offre.

Au sein du SSE$_{SQL}$, les **Arbres** Binaires de $\mathcal{L}_{\text{arbre}}$–requêtes ($\mathcal{L}_{\text{arbre}}$–arbres), décrits dans la section précédente, constituent la structure informatique de base utilisée comme support d'exécution des $\mathcal{L}_{\text{arbre}}$–requêtes.

Dans une approche « boîte noire », le SSE$_{SQL}$ est un simple système permettant de transformer un $\mathcal{L}_{\text{arbre}}$–arbre d'une $\mathcal{L}_{\text{arbre}}$–requête en une requête SQL. Le SSE$_{SQL}$ est de plus conçu pour fonctionner sur un SGBDR muni du modèle de données générique du D2IM donné dans la section 5.1.

De manière synthétique, ce modèle de données permet de définir de façon générique des objets, leurs attributs ainsi que des relations entre ces derniers. Ces trois éléments de modélisation correspondent intrinsèquement aux trois éléments de syntaxe de base du langage $\mathcal{L}_{\text{arbre}}$. Une $\mathcal{L}_{\text{arbre}}$–requête peut ainsi être exécutée pas à pas en parcourant l'$\mathcal{L}_{\text{arbre}}$–arbre lui correspondant.

La Table 7.4 ci-dessous donne les tables du modèle de données du D2IM qui permettent l'exécution de chaque élément de syntaxe d'une $\mathcal{L}_{\text{arbre}}$–requête :

Langage $\mathcal{L}_{\text{arbre}}$ élément de syntaxe	⇒	**Modèle de donnée du D2IM élément de modélisation**	**table(s) à cibler**
Clause entité	⇒	Object	`TB_OBJECT`
Contrainte d'attribut	⇒	Attribut	`TB_DATATYPE_PROPERTY`
Contrainte sémantique	⇒	Relation	`TB_OBJECT_PROPERTY` ou `TB_INDEXING` ou `TB_HIERARCHY`

TABLE 7.4 – Correspondance entre les éléments de syntaxe des $\mathcal{L}_{\text{arbre}}$–requêtes et les éléments du modèle de données du D2IM permettant leur exécution.

L'exécution d'un $\mathcal{L}_{\text{arbre}}$–requête est ainsi effectuée en parcourant l'$\mathcal{L}_{\text{arbre}}$–arbre correspondant à cet requête depuis sa racine vers les feuilles. Afin d'illustrer cette dernière, un exemple est donné ci-dessous :

Exemple 33 :

Les étapes ci-dessous permettent d'illustrer la construction progressive de la requête SQL résultant de la $\mathcal{L}_{\text{share}}$–requête suivante :

```
sejours(
  diag(
    cim10{IND_PMSI}(
      id="ICD_SC_J450"))
  ET analyse_biologique[FILTERS="IND_EDS_ANA_EDS_TYPE_ANA#EDS_BIO_GAZSATU"](
    valeur<=90)
  ET patient(
    sexe="1"))
```

Cette dernière permet la récupération des séjours rattachés à un diagnostic d'« asthme à prédominance allergique » [J450 (CIM–10)] des patients de sexe masculin ayant une saturation en oxygène inférieure à 90% dans ce séjour.

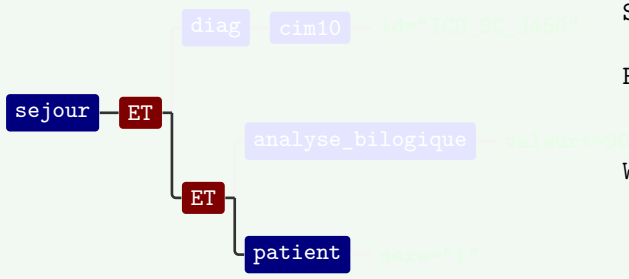
Les étapes suivantes mettent en évidence l'exécution de cette requête à mesure que l'$\mathcal{L}_{\text{share}}$–arbre de celle-ci est parcourue depuis sa racine. Pour plus de concision dans l'écriture des requêtes SQL, les raccourcis suivants ont été utilisés pour désigner certains noms de colonne :

RDF_RESOURCE	$\Leftrightarrow$	RR
RDF_RESOURCE_SOURCE	$\Leftrightarrow$	RR_S
RDF_RESOURCE_TARGET	$\Leftrightarrow$	RR_T
RDF_RESOURCE_INDEXED	$\Leftrightarrow$	RR_D
RDF_RESOURCE_INDEX	$\Leftrightarrow$	RR_I
TYPE_ID	$\Leftrightarrow$	T_{ID}

étape n° 1 :

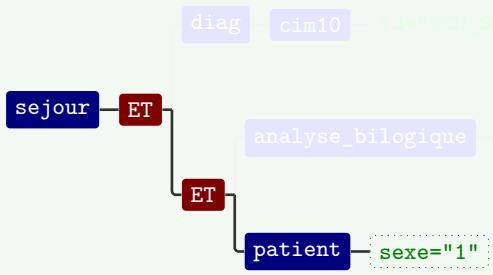

```
SELECT
  T1.RR
FROM
  TB_OBJECT T1
WHERE
  T1.T_ID= 'SEJOURS'
```

étape n° 2 :



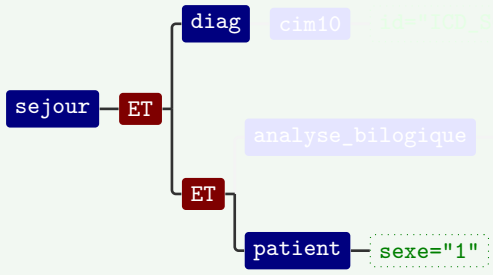

```
SELECT
  T1.RR
FROM
  TB_OBJECT T1,
  TB_OBJECT_PROPERTY T2
WHERE
  T1.T_ID= 'SEJOURS'
  AND T2.RR_S= T1.RR
  AND T2.T_ID= 'REL_SEJOURS_PATIENTS'
```

étape n° 3 :



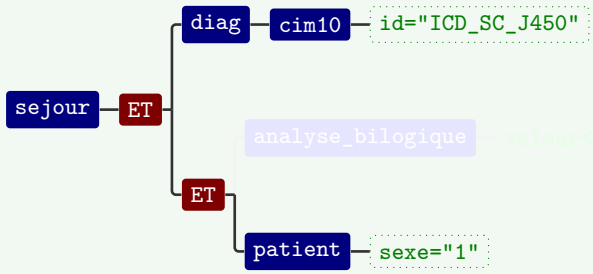

```
SELECT
  T1.RR
FROM
  TB_OBJECT T1,
  TB_OBJECT_PROPERTY T2,
  TB_DATATYPE_PROPERTY T3
WHERE
  T1.T_ID= 'SEJOURS'
  AND T2.RR_S= T1.RR
  AND T2.T_ID= 'REL_SEJOURS_PATIENTS'
  AND T3.RR= T2.RR_T
  AND T3.T_ID= 'PATIENTS.SEXE'
  AND T3.VAL = '1'
```

étape n° 4 :



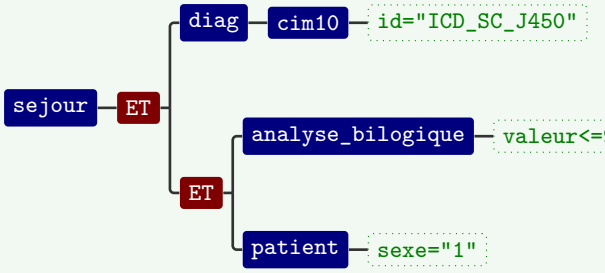

```
SELECT
  T1.RR
FROM
  TB_OBJECT T1,
  TB_OBJECT_PROPERTY T2,
  TB_DATATYPE_PROPERTY T3,
  TB_OBJECT_PROPERTY T4
WHERE
  T1.T_ID= 'SEJOURS'
  AND T2.RR_S= T1.RR
  AND T2.T_ID= 'REL_SEJOURS_PATIENTS'
  AND T3.RR= T2.RR_T
  AND T3.T_ID= 'PATIENTS.SEXE'
  AND T3.VAL = '1'
  AND T4.RR_T= T1.RR
  AND T4.T_ID= 'REL_DIAGNOSTICS_SEJOURS'
```

étape n° 5 :




```
SELECT
  T1.RR
FROM
  TB_OBJECT T1,
  TB_OBJECT_PROPERTY T2,
  TB_DATATYPE_PROPERTY T3,
  TB_OBJECT_PROPERTY T4,
  TB_INDEXING T5
WHERE
  T1.T_ID= 'SEJOURS'
  AND T2.RR_S= T1.RR
  AND T2.T_ID= 'REL_SEJOURS_PATIENTS'
  AND T3.RR= T2.RR_T
  AND T3.T_ID= 'PATIENTS.SEXE'
  AND T3.VAL = '1'
  AND T4.RR_T= T1.RR
  AND T4.T_ID= 'REL_DIAGNOSTICS_SEJOURS'
  AND T5.RR_D= T4.RR_S
  AND T5.T_ID= 'IND_PMSI'
  AND T5.RR_I= 'ICD_SC_J450'
```

étape n° 6,7,8 :




```
SELECT
  T1.RR
FROM
  TB_OBJECT T1,
  TB_OBJECT_PROPERTY T2,
  TB_DATATYPE_PROPERTY T3,
  TB_OBJECT_PROPERTY T4,
  TB_INDEXING T5,
  TB_OBJECT_PROPERTY T6,
  TB_INDEXING T7,
  TB_DATATYPE_PROPERTY T8
WHERE
  T1.T_ID= 'SEJOURS'
  AND T2.RR_S= T1.RR
  AND T2.T_ID= 'REL_SEJOURS_PATIENTS'
  AND T3.RR= T2.RR_T
  AND T3.T_ID= 'PATIENTS.SEXE'
  AND T3.VAL = '1'
  AND T4.RR_T= T1.RR
  AND T4.T_ID= 'REL_DIAGNOSTICS_SEJOURS'
  AND T5.RR_D= T4.RR_S
  AND T5.T_ID= 'IND_PMSI'
  AND T5.RR_I= 'ICD_SC_J450'
  AND T6.RR_T= T1.RR
  AND T6.T_ID= 'REL_EDS_ANA_SEJOURS'
  AND T7.RR_D= T6.RR_S
  AND T7.T_ID= 'IND_EDS_ANA_EDS_TYPE_ANA'
  AND T7.RR_I= 'EDS_BIO_GAZSATU'
  AND T8.RR= T6.RR_S
  AND T8.T_ID= 'EDS_ANA.RES'
  AND TO_NUMBER(T8.VAL, '999D99') <= 90
```

Bien que le SSE_{SQL} ne soit aujourd'hui plus maintenu compte tenu de ses faibles performances, il a non seulement permis d'implémenter la totalité de l'expressivité du langage $\mathcal{L}_{\text{share}}$ décrite dans la section 6.1 p. 130, mais également un certain nombre de fonctionnalités utiles qui n'étaient initialement pas prévues. Certaines d'entre elles sont présentées ci-dessous. Ces dernières correspondent toutes à des fonctionnalités qui ne sont pas fournies par le SSE_{NoSQL} :

Fonctions `MIN` et `MAX` : Les fonctions analytiques proposées par le langage SQL ont permis de mettre à disposition, au sein du langage $\mathcal{L}_{\text{share}}$, des syntaxes permettant la sélection de valeurs minimales et maximales de telle sorte que le SSE_{SQL} puisse interpréter et exécuter des requêtes telle que :

```
analyse_biologique(
   code_exe(label="Calcium")
   AND sejour(
      date_entree="MAX"
      AND patient(id="DM_PAT_47"))
   AND valeur="MIN")
```

Cette requête permet de sélectionner l'ensemble des *analyses biologiques de calcium ayant* ***la plus faible valeur*** *et ayant été effectués lors du* ***dernier séjour*** *du patient* `DM_PAT_47`.

Comparaison d'attributs d'une même entité : L'un des atouts majeurs du SSE_{SQL} par rapport au SSE_{NoSQL} est qu'il permet de comparer les valeurs d'attributs d'une même entité. Les mécanismes de jointure mis à disposition par le SQL permettent en effet de

constituer sans peine de telles comparaisons. À titre d'exemple, le SSE$_{SQL}$ permet notamment l'utilisation de requête du type :

```
patient(
   analyse_biologique(
      code_exe(label="Sodium")
      AND valeur>borneSup))
```

Celle-ci permet de rechercher les *patients possédant une analyse de Sodium* ***supérieure à la normale***. Elle permet donc de récupérer les *patients présentant une* ***Hypernatrémie***.

Comparaison d'attributs d'entités différentes : La comparaison de deux attributs d'entités différentes est également possible avec le SSE$_{SQL}$. Syntaxiquement, le langage $\mathcal{L}_{\text{share}}$ permet d'employer l'opérateur de « référence arrière » : « `../` ». Celui-ci permet de remonter à l'entité mère d'une clause entité. Un exemple de requête exploitant cette syntaxe est donné ci-dessous :

```
Niveau 1 :     acte_medical(
               date="MIN"
Niveau 2 :     AND patient(
                  id="DM_PAT_475"
Niveau 3 :        AND acte_medical(
                     date<../../date
                     AND code_CCAM(id="CCA_AM_HMFC004")))
```

Cette requête permet de rechercher *l'acte médical du patient 475 ayant survenu* ***juste après*** *un acte codé avec le code « Cholécystectomie, par coelioscopie »* [$HMFC004$ ($CCAM$)]. Dans cette requête, `../../date` désigne l'attribut `date` de l'entité `acte_medical` se trouvant au niveau 1 de la requête. Les deux attributs `date` des entités `acte_medical` de niveau 1 et 3 sont donc comparés pour construire une requête chronologique.

Maîtrise de l'expansion hiérarchique : SQL offre la possibilité d'exprimer des requêtes hiérarchiques. Dans le cadre du SSE$_{SQL}$, l'expansion hiérarchique est donc réalisée à la volée. Ainsi, cette expansion peut être paramétrée, notamment pour indiquer si elle doit être effectuée vers les concepts plus larges ou plus précis et avec quelle profondeur elle doit être réalisée. Dans l'exemple ci-dessous, la requête permet de rechercher l'ensemble des comptes-rendus indexés avec le concept « *maladie cardiaque (maladie)* » [56265001 (SNOMED–CT®)] ou bien l'un de ses fils (`EXPL="DOWN"`) à condition qu'ils n'en soient pas éloignés de plus de deux niveaux dans la hiérarchie de la SNOMED–CT® :

```
record(
   snomedct
      [EXPL="DOWN", EXPL_DEPTH = 2]
      (label="Heart disease (disorder)"))
```

7.2.3 Le Semantic Search Engine NoSQL (SSE$_{NoSQL}$)

Comme précisé dans la section 5.2, le SSE$_{NoSQL}$ repose sur une base de données NoSQL. Les fonctionnalités permettant la maintenance et l'accès « atomique » aux données de cette couche sont fournit par le SGBD NINJAC. Ce dernier met à disposition une API de base basée sur trois ensembles de structures de données :

— les **maps d'entités** qui sont décrites dans la Table 5.3.

— les **maps de jointures** qui sont décrites dans la Table 5.5.

— les **index inversés** Lucene qui sont maintenus pour chaque type objet utile à la RI au sein du SI.

Ces trois dernières constituent les uniques outils exploités par le SSE_{NoSQL} dans le processus d'exécution d'une $\mathcal{L}_{\text{<}}$–requête.

Tout comme le SSE_{SQL}, le SSE_{NoSQL} exécute les $\mathcal{L}_{\text{<}}$–requêtes à partir de l'$\mathcal{L}_{\text{<}}$–arbre obtenu en sortie de l'analyseur. Cependant, sa stratégie d'exécution diffère de celle de SSE_{SQL}.

Le SSE_{SQL} repose sur un SGBD relationnel muni du langage de requête SQL. Ce dernier est utilisé pour représenter l'intégralité d'un $\mathcal{L}_{\text{<}}$–arbre à l'aide d'une seule et même requête SQL **globale**. Pour le SSE_{SQL}, le $\mathcal{L}_{\text{<}}$–arbre n'est qu'une structure permettant une « traduction » la $\mathcal{L}_{\text{<}}$–requête qu'il représente en une requête SQL. C'est au SGBDR que revient la charge d'analyser la structure de la requête SQL et d'exécuter les différentes contraintes de celle-ci en organisant les accès basiques et atomiques aux données de la base de données.

En revanche, dans le cas du SSE_{NoSQL}, le langage $\mathcal{L}_{\text{<}}$ ne joue plus un simple rôle de langage intermédiaire entre l'humain et le SGBD. Il constitue pleinement **le langage de bas niveau** sur lequel repose la logique d'exécution du SSE_{NoSQL}. Il n'existe par ailleurs actuellement aucun langage de requête aussi puissant que le SQL pour des bases de données NoSQL de type IMDG. Par analogie, c'est donc au moteur SSE_{NoSQL} qu'incombe la charge d'analyser les $\mathcal{L}_{\text{<}}$–arbres et d'organiser les accès atomiques aux données nécessaire à l'exécution logique des nœuds de cet arbre.

En conséquence, la stratégie d'exécution du SSE_{NoSQL} repose sur une exécution **pas-à-pas** du $\mathcal{L}_{\text{<}}$–arbre et non sur une exécution globale de ce dernier. Chaque nœud du $\mathcal{L}_{\text{<}}$–arbre est **exécuté indépendamment**. De plus **chaque résultat intermédiaire est stocké au sein des nœuds des arbres** afin d'en permettre leurs **consommations** lors de l'exécution des nœuds parents.

L'exécution d'un $\mathcal{L}_{\text{<}}$–arbre s'effectue donc depuis **les feuilles du $\mathcal{L}_{\text{<}}$–arbre vers sa racine**.

Chacun des trois ensembles de structure de données est utilisé à des fins différentes et permettent la prise en charge de l'exécution de nœuds de types différents (i.e. entités, jointures, index). On donne ici le rôle de chacun d'entre eux. La Figure 7.8 reprend l'$\mathcal{L}_{\text{<}}$–arbre de l'exemple 32 en indiquant le rôle de ces structures de données en fonction des types de contraintes rencontrées dans cet arbre :

Les maps de base : Ces maps sont très peu exploitées dans le processus d'exécution du SSE_{NoSQL}. En réalité, seul le **cache d'objet** permettant de récupérer un objet (i.e. un DBObject) à partir de son identifiant (i.e. son `RDF_RESOURCE`) et réellement utilisé. Il permet d'exécuter les contraintes d'attribut basées sur un identifiant d'objet s'écrivant syntaxiquement sous la forme « `id="..."` ». Par exemple, dans la Figure 7.8, ce cache d'objet est utilisé pour récupérer les différents DBObjects correspondant aux divers concepts apparaissant dans la requête.

Les index inversés Lucene **:** Ils permettent de retrouver des objets à partir de la valeur de leurs attributs (textuel, numérique ou de type date). Ils permettent donc d'exécuter les contraintes d'attribut autres que celles gérées par le cache d'objet. Dans la Figure 7.8 par exemple, ces index permettent notamment de retrouver les objets de type `analyse_biologique` dont la valeur est supérieure à 4.

Les maps de Jointure : Ces caches, permettent quant à eux, de retrouver l'ensemble des entités reliées à une ou plusieurs autres par une relation. Du point de vue du langage $\mathcal{L}_{\text{<}}$, ils permettent donc l'exécution des contraintes sémantiques. Du point de vue du $\mathcal{L}_{\text{<}}$–arbre, cela s'assimile à la capacité à exécuter un nœud symbolisant une entité $\texttt{E}_1$ dont l'unique fils est également un nœud symbolisant une entité $\texttt{E}_2$. Plus précisément ces caches permettent de retrouver les objets de type $\texttt{E}_1$ reliés sémantiquement aux objets de type $\texttt{E}_2$ constituant le résultat partiel du nœud fils.

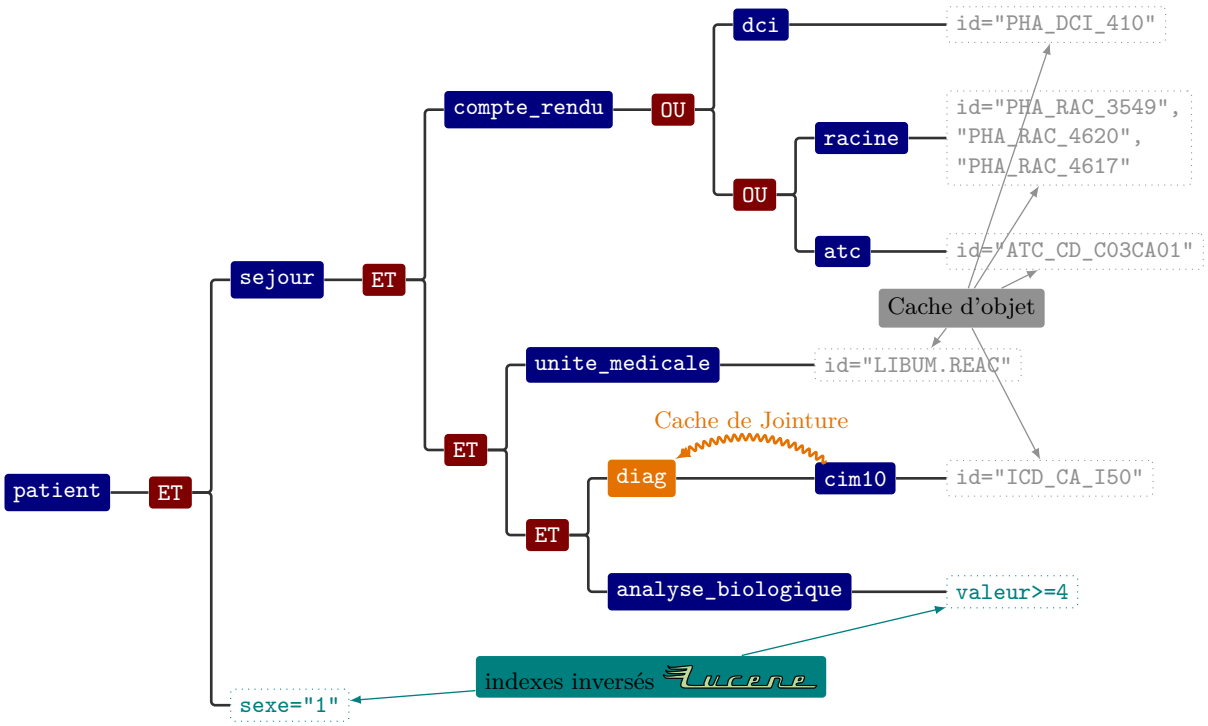
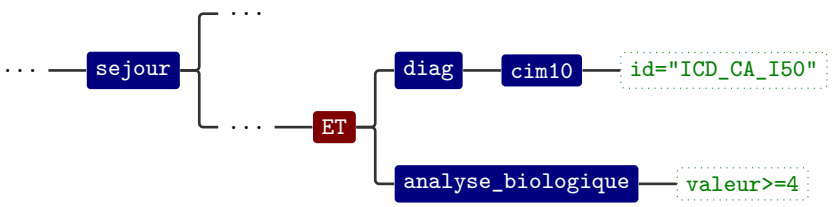


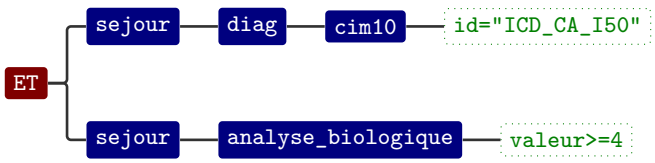
FIGURE 7.8 – Illustration de l'utilisation et du rôle des trois outils mis à disposition par le SI NoSQL du D2IM dans le processus de l'exécution d'un $\mathcal{L}_{\curvearrowright}$–arbre. Le $\mathcal{L}_{\curvearrowright}$–arbre en question est repris de l'exemple 32. Le cache d'objet permet l'exécution des contraintes d'attribut basées sur l'identifiant d'une entité, les caches de jointure permettent l'exécution des contraintes sémantiques et les index inversés lucene l'exécution des contraintes d'attribut dans leurs ensembles.

L'exécution des contraintes d'attribut ainsi que des contraintes sémantiques est réalisé à l'aide des maps de base, des maps de jointure et des index inversés lucene. En revanche, l'exécution des nœuds symbolisant des opérateurs Booléens consiste à effectuer des intersections et unions ensemblistes entre les résultats partiels de deux nœuds qu'ils relient. Ces opérations n'ont cependant réellement de sens que lorsque les objets sur lesquels s'appliquent cette opération ensembliste sont de mêmes types (i.e. correspondent à une même entité).

Cette problématique peut notamment être mise en évidence dans l'$\mathcal{L}_{\curvearrowright}$–arbre de la Figure 7.8. Le nœud symbolisant l'opérateur Booléen `ET` reliant les nœuds `diag` et `analyse_biologique` ne peut, dans l'état, être exécuté en réalisant une intersection entre les DBObjects constituant les résultats des nœuds `diag` et `analyse_biologique`. Ces DBObjects étant de types différents, leur intersection aboutirait en effet à un résultat vide. Afin de rendre « cohérente » cette intersection, l'entité mère commune à ces deux entités doit être prise en compte. Dans le cadre de cet exemple, cette entité commune est l'entité `sejour` et sa prise en compte reviendrait alors à exécuter le sous-$\mathcal{L}_{\curvearrowright}$–arbre :



L'exécution de ce sous-arbre devient alors possible en considérant l'$\mathcal{L}_{\curvearrowright}$–arbre équivalent :



Une intersection entre les résultats des nœuds fils du noeud `ET` est en effet possible et cohérente compte tenu que ces deux derniers correspondent tous deux à des ensembles de DBObjects de type `sejour`.

L'exécution de sous-arbres équivalent est cependant réalisée dans le respect des propriétés algébriques du langage $\mathcal{L}_{\text{arbre}}$. En généralisant ce procédé il est alors possible d'aboutir à un processus d'exécution cohérent et ordonné de tout $\mathcal{L}_{\text{arbre}}$–arbre.

Dans ce chapitre, l'intégration du SSE_{SQL} et du SSE_{NoSQL} au sein du SI du D2IM ainsi que les méthodes employées pour les implémenter ont été présentées de manière générale. Dans le chapitre suivant, je m'intéresse plus spécifiquement au contexte des données cliniques. Le SSE_{NoSQL} constitue aujourd'hui l'un des outils sur lequel repose l'EDSS du CHU de Rouen. L'interface ASIS permet notamment d'exploiter ce moteur en générant des requêtes portant sur les données cliniques de cet entrepôt. Je débute donc le chapitre suivant par une description de cette interface. Dans un second temps, je présente la méthodologie et le contexte d'une étude que j'ai mené afin d'évaluer la capacité du SSE_{NoSQL} à répondre à des critères d'inclusion et d'exclusion de patients dans des études cliniques.

Chapitre 8

Résultats

Sommaire

8.1 Accès Sémantique à l'Information de Santé (ASIS) 200
8.1.1 Première étape : la définition des contraintes 200
8.1.2 Deuxième étape : la constitution d'une requête Booléenne 202
8.1.3 Troisième étape : le choix du type de donnée de sortie 203
8.2 Évaluation 205
8.2.1 La recherche d'information au sein des textes Cliniques 205
8.2.2 Méthodologie 208
8.2.3 Résultats de l'évaluation 212

8.1 Accès Sémantique à l'Information de Santé (ASIS)

Accès **S**émantique à l'**I**nformation de **S**anté (**ASIS**) est le nom d'une application Web développée par le D2IM et permettant d'effectuer de la RI au sein des données de santé de patients du CHU de Rouen. Elle constitue une des applications possibles d'accès à l'EDSS. Tout comme les applications DocCISMeF et LiSSa, celle-ci repose sur le moteur de recherche SSE_{NoSQL}. Elle exploite en revanche une instance propre du SGBD NINJAC, intégrant des données de Santé de 1,8 millions de patients du CHU de Rouen.

L'application **ASIS** n'est pas une application originale dans le sens où elle est issue d'une « adaptation » de l'application DocCISMeF qui a, elle, été développée par divers ingénieurs de l'équipe et notamment Badisse DAHAMNA. Bien qu'ayant collaboré à cette « adaptation », le développement informatique de cette application ne peut pas être intégralement attribué aux travaux de thèse décrits dans ce mémoire. D'un point de vue fonctionnel, **ASIS** fournit une **interface graphique** imaginée par le professeur Stéfan J. DARMONI et composée de **formulaires** faciles à utiliser et à partir desquels une $\mathcal{L}_{\text{share}}$–requête peut être générée et transmise au SSE_{NoSQL}.

Dans cette section, les différents aspects de l'interface **ASIS** mettant en valeur le caractère sémantique de l'EDSS seront présentés. L'interface d'**ASIS** se décompose notamment en quatre étapes distinctes et clairement mises en évidence graphiquement. Chacune de ces quatre étapes sera présentée.

8.1.1 Première étape : la définition des contraintes

La première étape du processus de requêtage consiste à **définir un ensemble de contraintes** que devront vérifier les données renvoyées en sortie de l'outil. La philosophie de l'interface repose sur le caractère « multi-entités » du langage $\mathcal{L}_{\text{share}}$. Elle permet de définir des contraintes portant sur les différentes entités utilisées dans la modélisation de l'information de santé de l'EDSS. Une copie d'écran du formulaire permettant de constituer ces contraintes est donnée dans la Figure 8.1.

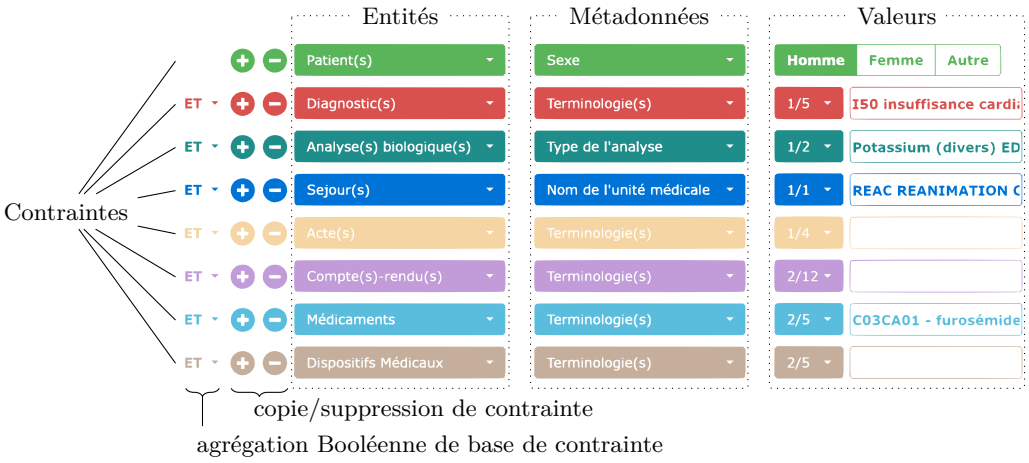


FIGURE 8.1 – Étape n° 1 de l'interface proposée par **ASIS** permettant de définir des contraintes relative à différentes entités et différentes métadonnées.

Ce formulaire fait apparaître trois zones majeures : « Entités », « Métadonnées » et « Valeurs ». Elles correspondent aux trois informations qu'il est nécessaire de renseigner pour constituer une contrainte à savoir :

1. l'entité concernée par la contrainte (e.g. *patient*) ;
2. la métadonnée de cette entité qui est visée (e.g. *date de naissance* ou encore *sexe* si l'entité choisie est *patient*) ;
3. la valeur requise pour cette métadonnée (e.g. *Homme* ou *Femme* pour la métadonnée *sexe* de l'entité *patient*).

Le sens intrinsèque de ces contraintes est intuitif. Par exemple, la première contrainte définie dans la Figure 8.1 impose que les données renvoyées par le système soit celles de « *patients de sexe masculin* » tandis que la deuxième impose qu'un « *Diagnostic d'insuffisance cardiaque* » ait été posé.

Du point de vue du langage $\mathcal{L}_{\text{share}}$, ces contraintes correspondent à des clauses entité munies d'une contrainte d'attribut ou sémantique. Par exemple, la première contrainte correspond en interne à la $\mathcal{L}_{\text{share}}$–requête `patient(sexe="M")`.

Huit types d'objets (i.e. d'entités) sont proposés par l'interface en revanche seulement six d'entre eux ont une réelle existence au sein de l'EDSS : Patients, Diagnostics, Analyses biologiques, Séjours, Actes et Comptes-rendus. Les entités Médicaments et Dispositifs Médicaux sont en effet également proposées par l'interface d'**ASIS** mais sont en réalité « fictives ». Les contraintes qu'elles permettent de construire correspondent, en interne, à des contraintes portant sur les comptes-rendus. La présence de ces entités dans l'interface est cependant justifiée par le fait que les champs proposés pour constituer ces contraintes présentent des spécificités qui facilitent respectivement la recherche des médicaments et des dispositifs médicaux au sein des comptes-rendus.

L'EDSS repose sur une **description sémantique** de l'information de santé. Cette dernière est réalisée à l'aide de concepts terminologiques et/ou ontologiques issus du portail **HeTOP**. Dans de nombreux cas, le renseignement des valeurs des métadonnées consiste à sélectionner un ou plusieurs de ces concepts. L'interface propose des composants graphiques effectuant une « auto-complétion » en temps réel des libellés saisis par l'utilisateur. Les concepts proposés par ces composants sont alors issues de TOs pertinentes vis-à-vis de l'entité et de la métadonnée choisies. De même, les résultats sont organisés et regroupés afin de permettre une sélection plus simple. La capture d'écran donnée dans la Figure 8.2 met en évidence la saisie de la valeur dans le cas de la contrainte de type *Médicament* de la Figure 8.1.

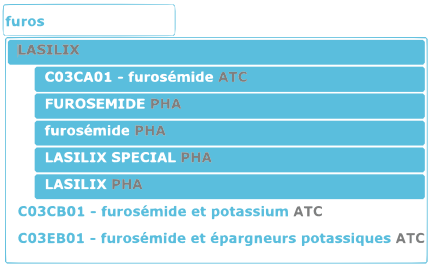


FIGURE 8.2 – Saisie avec auto-complétion de la valeur d'une contrainte de type *Médicaments*.

Dans le cadre de la Figure 8.1, ce procédé se retrouve également pour la contrainte de type *Diagnostic* qui permet la saisie du concept « *Insuffisance Cardiaque* » [*I*50 (CIM–10)] ou encore celle du concept d'analyse pour la contrainte de type *Analyse Biologique*. L'utilisation de concepts permet de plus, comme c'est le cas dans le cadre de la RI documentaire classique, d'exploiter la hiérarchie des concepts afin d'élargir les recherches. Par exemple, une fois le concept d'insuffisance cardiaque sélectionné pour la contrainte *Diagnostic*, l'interface propose d'élargir la recherche en effectuant une explosion hiérarchique du concept sélectionné (Figure 8.3).

FIGURE 8.3 – Options d'affinage d'une contrainte de type *Diagnostic* donnant la possibilité d'inclure les descendants du concept sélectionné, de rechercher ce concept au sein des comptes-rendus en plus des diagnostics et de préciser le type de diagnostic (principal ou relié).

Dans le cas de la contrainte *Analyse Biologique*, des options d'affinage sont également et automatiquement proposées à l'utilisateur une fois le type de cette dernière choisi. Ces dernières permettent notamment de contraindre la valeur de celle-ci (Figure 8.4).

FIGURE 8.4 – Options d'affinage d'une contrainte de type *Analyse Biologique* donnant la possibilité de contraindre la valeur de celle-ci et de considérer également les concepts fils du concept d'analyse sélectionnée.

8.1.2 Deuxième étape : la constitution d'une requête Booléenne

L'étape n° 1 décrite dans la section précédente permet de constituer un ensemble de contraintes portant sur diverses entités. Ces dernières sont cependant, a priori, définies de manière indépendante. L'étape n° 2, décrite dans cette section, permet de lier logiquement ces contraintes. Plus précisément, elle permet d'agréger ces contraintes unitaires à l'aide d'opérateurs Booléens afin d'en **former une contrainte Booléenne globale**.

Pour atteindre cet objectif, une **zone éditable** permet de manipuler les contraintes unitaires définies précédemment sous forme de boutons. Une copie d'écran de ce composant graphique est donnée Figure 8.5.

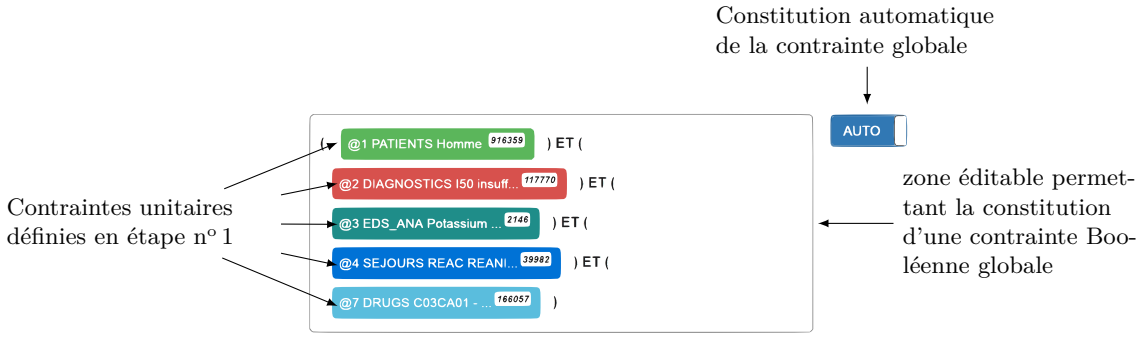


FIGURE 8.5 – Composant graphique éditable permettant la composition d'une contrainte Booléenne globale à partir des contraintes unitaires définies à l'aide du formulaire de l'étape n° 1.

Bien que l'étape n° 2 soit spécifiquement dédiée à la constitution d'une contrainte Booléenne globale, l'étape n° 1 permet de pré-définir un squelette de base de celle-ci. La zone éditable est en réalité pré-remplie « à la volée » à mesure que les contraintes unitaires de l'étape n° 1 sont constituées. Cette option peut néanmoins être activée/désactivée à l'aide d'un bouton de type « va-et-vient » (Figure 8.5 p. 202).

Au sein du formulaire de définition des contraintes unitaires (étape n° 1), la maîtrise de ce squelette de base s'effectue à l'aide :

— des boutons « + » et « - » qui permettent respectivement d'ajouter une contrainte par copie et d'en supprimer une ;

— des listes déroulantes permettant de spécifier un opérateur *ET* ou *OU*.

Un exemple est donné dans la Figure 8.6. Lorsque qu'aucune action visant à maîtriser ce squelette n'est réalisée, la contrainte Booléenne globale par défaut est constituée de l'ensemble des contraintes unitaires définies dans l'étape n° 1 liées entre elles par l'opérateur Booléen de conjonction. Ainsi, une contrainte Booléenne globale est systématiquement proposée à l'utilisateur. En l'occurrence, la Figure 8.5 correspond à la contrainte Booléenne globale proposé par défaut par l'application pour le formulaire de la Figure 8.1.

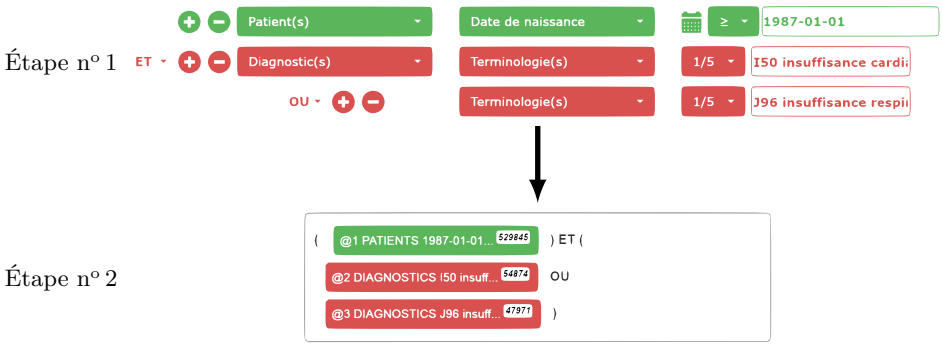


FIGURE 8.6 – Gestion d'un squelette de contrainte Booléenne globale depuis le formulaire de l'étape nº 1.

La zone éditable est munie d'un mécanisme d'auto-complétion. Lors de la définition manuelle d'une contrainte globale, il est alors possible de saisir le numéro d'une contrainte unitaire ou son type d'entité afin de la sélectionner parmi une liste déroulante. Chaque contrainte unitaire est représentée par un bouton cliquable indiquant le nombre de résultat correspondant à la sous-$\mathcal{L}_{\bullet}$–requête générée pour cette contrainte. Un clic sur ce bouton permet de visualiser les résultats de cette sous-requête.

8.1.3 Troisième étape : le choix du type de donnée de sortie

L'étape trois consiste à choisir un ou plusieurs types d'entité désirée(s) en sortie de l'outil. Ce choix est effectué à l'aide du formulaire dont une copie d'écran est donnée Figure 8.7.

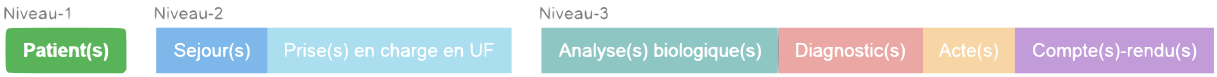


FIGURE 8.7 – Formulaire permettant de choisir le ou les différents types d'entité désiré(s) en sortie de l'application.

Les entités sont réparties en trois groupes qui correspondent aux trois catégories d'information clinique déja évoqués précédement dans le chapitre 2 mais aussi plus spécifiquement dans la Figure 5.3 du chapitre 5.

La sélection d'une entité génère un bouton affichant le nombre de résultats. Ce dernier est similaire à ceux des contraintes unitaires générées dans la zone éditable de l'étape nº 2 et permet, lorsqu'il est cliqué, d'afficher la liste des résultats (Figure 8.8).

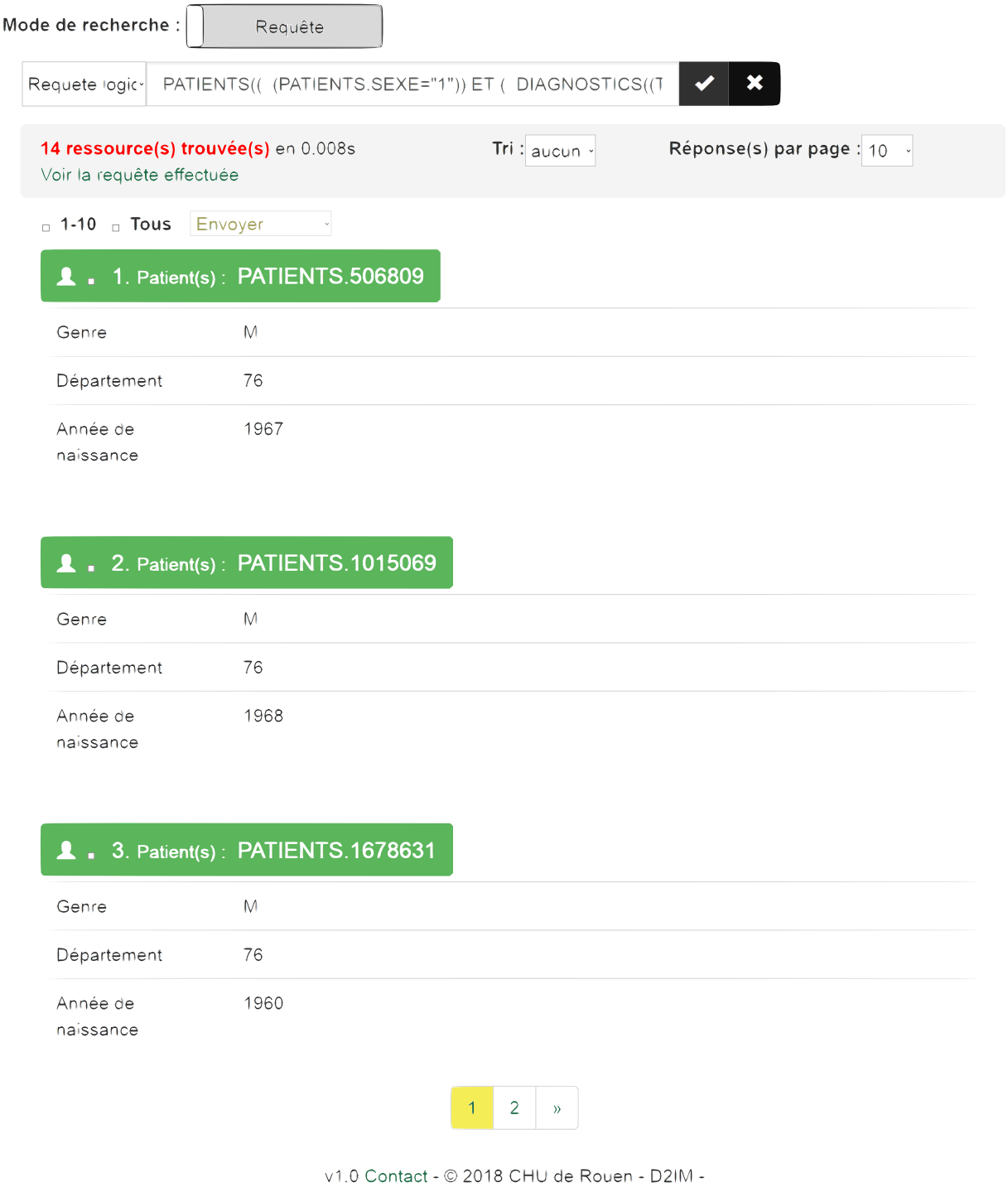


FIGURE 8.8 – Copie d'écran des résultats affichés par **ASIS**.

8.2 Évaluation

L'étude présentée dans cette section vise à apprécier les forces et les faiblesses de l'EDSS et des divers outils qui l'entourent dans le cadre d'un **cas d'utilisation concret**. Il s'agit plus précisément d'évaluer la capacité du système à répondre à des **critères d'inclusion et d'exclusion de plusieurs études cliniques**.

Bien que le SSE_{NoSQL} et l'application Web ASIS soient pleinement concernés par cette étude, les apports et les lacunes que celle-ci permet de mettre en évidence dépassent le cadre des fonctionnalités de ces deux outils. L'aspect sémantique de la RI au sein de l'EDSS, qui constitue la caractéristique première de cet entrepôt, ne résulte en effet pas seulement des outils assurant l'accès à l'information de santé mais également de la description sémantique de cette dernière.

8.2.1 La recherche d'information au sein des textes Cliniques

Le SI du CHU de Rouen compte plusieurs millions de **textes cliniques**. Il existe essentiellement deux approches permettant de rechercher l'information contenue dans ces données non structurées :

La recherche plein texte [1, 2] **:** Cette méthode est largement exploitée et/ou privilégiée par les outils d'accès aux EDSs présents dans la littérature. À titre d'exemple on peut notamment citer l'EMERSE (Michigan, **É**tats **U**nis d'**A**mérique (USA)) et plus localement DrWarehouse qui sont tous deux des outils spécialement conçus pour fournir ce type de recherche sur les textes cliniques.
La recherche plein texte consiste en un appariement des mots saisis par les utilisateurs avec ceux contenus dans les textes cliniques. Sa mise en place requiert donc la création d'index textuels permettant de rechercher les documents (i.e. les textes) à partir des mots qu'ils contiennent. Dans une grande majorité des cas, ces textes sont stockés au sein d'un SGBDR. Les index sont alors construits à l'aide des fonctionnalités offertes par celui-ci. Par exemple, dans le cas d'ORACLE, il s'agit des index dits de « contexte ». La couche NoSQL qui sert d'interface d'accès aux données du SI du D2IM impose cependant l'utilisation d'autres technologies.

L'annotation sémantique : Elle constitue également une approche classique de la RI sur les données non structurées. Elle vise en réalité à « restructurer » a posteriori les textes cliniques à l'aide de SOCs. Plus précisément, l'annotation sémantique vise à rattacher à ces textes des concepts terminologiques et/ou ontologiques susceptibles de décrire leurs contenus. La recherche d'information peut alors être effectuée en recherchant ces concepts tout en bénéficiant de leurs structures (i.e. leurs hiérarchies, leurs termes alternatifs, leurs alignements, etc.).

Dans le cadre de l'EDSS, la RI au sein des textes cliniques du CHU de Rouen peut être effectuée à l'aide de ces deux approches. L'implémentation de chacune de ces méthodes est décrite dans les sections suivantes, avec au préalable la description de l'extraction des textes cliniques du SIH du CHU de Rouen.

8.2.1.1 L'extraction

Depuis la fin des années 1990, le CHU de Rouen produit et archive divers textes cliniques et de santé sous forme électronique. Dans le cadre de cette étude, les documents créés entre la fin des années 1990 et le mois de novembre 2018 ont été récupérés. La base de données CDP (cf. section 1.6) fait état de plus de 11 928 168 textes sur cette période. Dans le cadre du CHU de Rouen, ces textes se présentent sous forme de fichiers générés avec la suite de logiciels de bureautique Microsoft (Microsoft). On retrouve ainsi ces documents sous différentes format (viz. `.rtf` pour les plus anciens d'entre eux, `.doc` pour une grand majorité et enfin `.docx` pour les

1. : « full-text search »
2. aussi appelée « recherche en texte intégral » ou « recherche de texte libre »

plus récents).

Dans CDP, ces documents sont compressés au sein d'archives `ZIP` puis stockés sous forme binaire au sein de la base de données ORACLE à l'aide du type de données `LONG RAW`. Afin de fournir les fonctionnalités de recherche plein texte sur ces textes, des scripts Java™ ont été développés afin :

1. D'extraire les archives d'extensions `.zip`.
2. De décompresser ces archives afin d'en extraire les documents Microsoft (i.e. `.rtf` ou `.doc` ou `.docx`).
3. De convertir ces fichiers en simple fichiers textes (i.e. en `.txt`).

C'est au sein de ces fichiers textes que l'EDSS effectue la RI.

8.2.1.2 La recherche d'information en texte intégral

L'EDSS ne repose pas directement sur un SGBDR mais s'intègre dans un environnement technique NoSQL auquel il est possible d'accéder par l'intermédiaire du SGBD NINJAC. Il n'est, de ce fait, pas possible de bénéficier des structures d'index nativement proposée par les SGBDRs.

Afin de mettre en place cette recherche plein texte, des index inversés Lucene ont été générés pour chacun des fichiers textes obtenus suite à leurs extractions de la base CDP.

Ces index sont similaires aux index utilisés par le SSE_{NoSQL} pour exécuter les contraintes d'attribut des $\mathcal{L}_{\text{arc}}$–requêtes (index permettant de retrouver un objet par l'intermédiaire de la valeur d'un de ses attributs).

Les comptes-rendus sont alors rendus disponibles au langage $\mathcal{L}_{\text{arc}}$ par le biais d'une modélisation de ceux-ci comme de simples entités dont les textes constituent un de leurs attributs (au même titre que `sexe` constitue un attribut de l'entité `patient` par exemple).

Au sein de la logique interne du SSE_{NoSQL}, la recherche s'effectue donc à l'aide du langage de requête spécifique à Lucene au même titre que n'importe quelle contrainte d'attribut.

8.2.1.3 La recherche d'information sémantique

L'exploitation de techniques de « RI sémantique » n'est pas une pratique nouvelle pour le D2IM. De nombreux travaux ont, en effet, été effectués ces dernières années, autour des TOs de Santé. Le SI de l'équipe repose aujourd'hui sur un socle terminologique et/ou ontologique important exploité par la plupart des outils développés au sein de l'équipe et dont l'accès est assuré par le portail HeTOP.

Dans le cadre spécifique de l'**extraction d'information clinique**, les **méthodes de TAL restent prédominantes**, et ce en dépit de l'intérêt croissant qu'il est porté aux méthodes statistiques et d'apprentissage automatique [158, 159]. Ce constat s'explique notamment par le manque à la fois d'interprétabilité et d'interopérabilité de ces dernières. A contrario, les bases de connaissance que constituent les terminologies et ontologies offrent ces possibilités et jouent, de surcroît, un rôle crucial dans toutes les tâches de TAL [160].

Une approche classique de TAL sur les textes cliniques consiste à utiliser un annotateur sémantique. Ce dernier permet de rattacher des concepts terminologiques aux mots et expressions présentes dans un texte. Dans le cadre du D2IM, l'annotateur sémantique ECMT permet une annotation de textes avec les différentes TOs du portail HeTOP.

Une multitude d'annotateurs sémantiques a été proposée ces dernières années dans la littérature. Une grande partie d'entre eux sont cependant destinés à la langue anglaise. L'annotation sémantique de la langue française, et plus généralement des langues autres que l'anglais, reste

encore aujourd'hui un challenge [161] [3] . Nombre d'annotateurs sémantiques reposent, en effet, sur des TOs en anglais (e.g. **C**linical **T**ext **A**nalysis and **K**nowledge **E**xtraction **S**ystem™ (cTAKES) (SNOMED–CT®, RxNorm) [162], NCBO Annotator [163]) ou bien sur l'UMLS au sein duquel le français est peu représenté (e.g. MetaMap [164]).

L'ECMT repose, lui, sur le portail HeTOP qui fournit un accès aux concepts et termes en français beaucoup plus étendu que l'UMLS. De plus, HeTOP intègre à la fois des SOCs génériques et spécialisés (e.g. maladies rares (orphanet), dispositifs médicaux (**CLA**ssification des **DI**spositifs **MÉD**icaux (CLADIMED)), etc.). Cette diversité permet d'annoter des textes cliniques issus d'un large panel de spécialités médicales.

Dans le cadre de l'EDSS, les 11,9 millions de comptes-rendus, d'ordonnances et de courriers du CHU de Rouen ont été annotés à l'aide de l'ECMT. La Figure 8.9 synthétise sous forme de schéma ce processus.

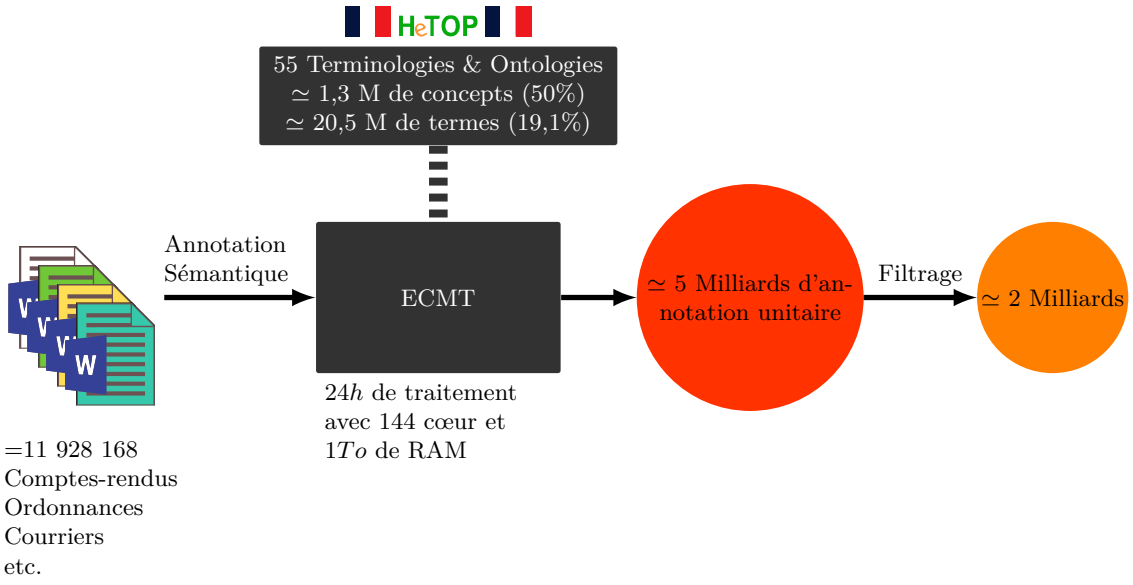


FIGURE 8.9 – Annotation sémantique des documents médicaux du CHU de Rouen et post-filtrage manuel des annotations.

Seules les 55 TOs totalement ou partiellement disponibles en français du HeTOP ont été utilisées. La partie francophone d'HeTOP représente 1,3 million de concepts et 20,5 millions de termes soit respectivement environ 50% et 19,1% des concepts et termes disponibles au total.

Afin de rendre cette annotation techniquement possible, des optimisations de l'ECMT ont été entreprises. Ces dernières se sont matérialisées principalement par la mise en cache des indexations récurrentes.

De plus, l'utilisation d'une machine puissante munie de 144 cœurs et de 1*To* de RAM s'est avérée nécessaire. Les optimisations effectuées couplées à l'emploi de cette machine ont ainsi permis de réaliser l'annotation des 11,9 millions de textes cliniques en un peu moins de 24 heures.[4]

Un total de 5 043 731 628 annotations unitaires [5] a ainsi pu être généré. Une observation

3. Dans cet article, une revue de la littérature des méthodes de TAL portant sur des données cliniques en langue autre que l'anglais est menée. Dans cette étude, le français arrive en tête des langages investigués suivi de l'allemand et du chinois.

4. Bien que les applications du SI accèdent et stockent les données à travers une couche NoSQL, un SGBDR est néanmoins utilisé pour persister ces dernières. Ce temps d'exécution de 24 heures n'inclut pas les temps d'insertion des annotations générées par l'ECMT dans le SGBDR (celles-ci ayant été réalisées a posteriori).

5. L'annotation de deux textes différents avec le même concept étant comptabilisé comme deux annotations distinctes de même que l'annotation de deux portions d'un même texte avec le même concept.

des annotations obtenues a permis de mettre en avant le caractère **peu informatif** de certaines d'entre elles et la **nécessité de les filtrer**. La Table 8.1 donne à titre d'exemple les 10 annotations les plus fréquemment obtenues.

Libellés de concepts	**Nombre d'annotations**
« *Centre Hospitalier Universitaire* »	29 422 878
« *Compte-rendu* »	15 560 399
« *Docteur* »	11 279 905
« *Traitement* »	3 107 854
« *Gauche* »	2 769 142
« *Médecin traitant* »	2 493 460
« *Prise en charge* »	2 160 465
« *Motif d'hospitalisation* »	2 115 910
« *Indication* »	1 985 944
« *Maladie intercurrente* »	1 980 116

TABLE 8.1 – Top 10 des annotations les plus fréquente avant filtrage

Ces dernières ne relèvent, en effet, pas nécessairement d'une information sur la santé des patients. Elles « bruitent » l'annotation globale des textes cliniques et, d'un point de vue plus opérationnel, consomment inutilement de l'espace mémoire.

Bien qu'effectivement fréquent, une grande partie de ces termes apparaît davantage au sein des entêtes et titres de sections ou de paragraphes des textes cliniques que dans le corps de ces derniers (e.g. « *Centre Hospitalier Universitaire* », « *Compte-rendu* », « *Docteur* », « *Motif d'hospitalisation* » ou encore « *Maladie intercurrente* »).

De même, certaines de ces annotations sont trop génériques ou peu informatives prises isolément (e.g. « *Traitement* », « *Gauche* », « *Médecin traitant* »).

Enfin, l'apport informatif potentiel de certaines de ces annotations est à remettre en question compte tenu de l'existence de cette même information par ailleurs en tant que donnée structurée (e.g. annoter un compte-rendu avec « *Compte-rendu* » est inutile compte tenu qu'un type de documents plus granulaire lui est rattaché de manière structurée par ailleurs).

Un **filtrage manuel** a été entrepris par un certain nombre de membre du **D2IM**. Celui ci s'est effectué sur la base de l'observation des 5 000 concepts les plus identifiés et a permis de ne conserver que 2 087 784 055 annotations unitaires (soit un pourcentage d'annotations filtrées d'environ 58,6% et un pourcentage d'annotations conservées d'environ 41,4%).

Une étude de la couverture terminologique observée a été menée dans [59]. Les principaux résultats de cette dernière sont donnés dans l'Annexe C.

8.2.2 Méthodologie

Dans cette section, la méthodologie qui a été employée pour l'évaluation, et dont traite plus généralement la section 8.2 dans son ensemble, est détaillée. Cette dernière vise à évaluer la capacité, de l'EDSS et de « ses » outils, à automatiser et/ou à favoriser la recherche des patients répondant à des critères d'inclusion et d'exclusion d'études cliniques.

Cinq études cliniques du CHU de Rouen ont été sélectionnées aléatoirement comptabilisant, au total, 36 critères d'inclusion et 59 critères d'exclusion (soit 95 critères en tout). La liste complète de ces critères par étude clinique est donnée dans l'Annexe B.

L'évaluation a consisté à utiliser l'interface **ASIS** et/ou des $\mathcal{L}_{\text{share}}$–requêtes pour tenter de répondre à chacun des 95 critères. Ces derniers ont été traités et interprétés indépendamment les uns des autres, mais aussi indépendamment du contexte et de l'objectif général de l'étude clinique à laquelle ils se rapportaient.

Pour chaque critère, une **stratégie** de recherche du critère a été définie. Chaque stratégie a été construite à l'aide d'une ou de plusieurs **directives** de recherche agrégées en $\mathcal{L}_{\text{<}}$–requête.

Une stratégie de recherche correspond à une $\mathcal{L}_{\text{<}}$–requête globale tandis que les directives correspondent aux diverses contraintes dont elle est composée. Elles se présentent également sous forme d'une $\mathcal{L}_{\text{<}}$–requête mais se focalisent sur une unique source d'information (i.e. une entité telle que Diagnostic, Séjour, Analyse Biologique, etc.).

Une directive est donc équivalente à une contrainte de l'interface **ASIS** définissable dans l'étape n° 1 de cette celle-ci et une stratégie correspond à la requête finale formée à son étape n° 2.

La définition de la stratégie et de ses directives pour chaque critère a nécessité la collaboration d'un médecin (essentiellement Mehdi TAALBA) et d'un ingénieur en informatique (moi).

Les rôles de ces derniers étaient alors les suivants :

Le Médecin : interpréter cliniquement le critère et définir les différentes sources d'information à cibler afin d'y répondre.

L'ingénieur en Informatique : maîtriser les outils de requêtage (i.e. **ASIS** et le langage $\mathcal{L}_{\text{<}}$) afin de créer une requête à partir de l'analyse clinique du critère effectuée par le médecin.

La Figure 8.10 synthètise ce processus de contruction des stratégies de recherche pour chaque critère d'inclusion ou d'exclusion de patient. Chaque critère a alors été **classifié** de manière

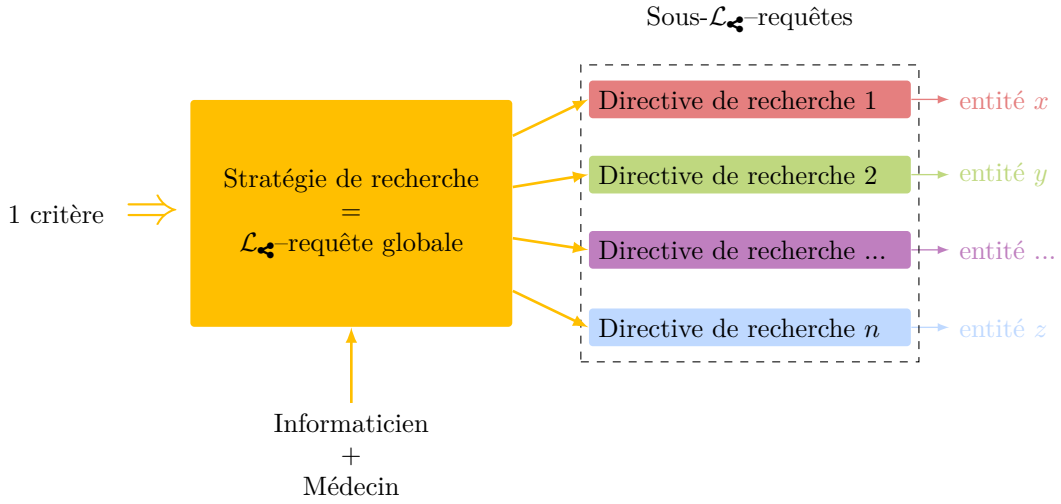


FIGURE 8.10 – Représentation globale du processus de construction d'une $\mathcal{L}_{\text{<}}$–requête pour un critère d'étude clinique quelconque.

« **empirique** » selon **six niveaux de prise en charge** par le système. Ces six niveaux et leurs conditions d'attribution aux critères sont énumérés et décrits ci-dessous. Des exemples permettant de « saisir » et d'illustrer les conditions d'attribution de chacun de ces six niveaux de prise en charge aux critère sont également donnés.

Définition 31 (Niveaux de prise en charge) :
Les six niveaux de prise en charge des critères d'études cliniques par l'EDSS et les outils qui l'exploitent ont été définis comme suit :

***Total* :** *Critères pleinement pris en charge et pouvant être automatisés de manière sûre par le système en renvoyant intégralement et uniquement les ressources satisfaisant exactement les exigences du critère (e.g. « patients présentant une leuconeutropénie (taux de polynucléaires neutrophiles inférieur à 1700/mm3 »)*

***Précis* :** *Critères reposant sur la recherche des données renseignées et requêtables de manière fiable au sein du système. La recherche de ce type de critère peut néanmoins mener à quelques résultats non-pertinents en fonction des choix effectués dans la définition de la stratégie de recherche et notamment en ce qui concerne le choix des concepts terminologiques ou ontologiques les plus appropriés aux exigences du critère (e.g. « patient atteint d'hépatite B ou d'hépatite C active ou non » ou encore « Insuffisance rénale aiguë »).*

***Partiel* :** *Critères pour lesquels la recherche aboutit à un manque de précision avéré (i.e. résultats pertinents non-renvoyés ou résultats non-pertinents renvoyés). Ils correspondent aux critères ne pouvant être recherchés que partiellement et pour lesquels un élargissement ou une spécification des contraintes de la stratégie de recherche s'avère inévitable. Ils impliquent la nécessité d'un post-filtrage et/ou d'une supervision des résultats par un professionnel de santé pour évaluer leurs adéquations avec le critère original (e.g. « Patient présentant une pathologie organique digestive et/ou inflammatoire évolutive » ou encore « troubles du rythme cardiaque mal équilibrés »).*

***Imprécis* :** *Critères qui ne peuvent être recherchés de manière suffisamment précise (techniquement et/ou en terme de données) afin de satisfaire aux exigences de base de ces derniers ou pour lesquels aucune stratégie de recherche ne permet de fournir de manière constante des résultats cohérents (e.g. « femme enceinte ou allaitante » ou encore « Patient âgé de plus de 65 ans hospitalisé pour une hémorragie digestive haute ou basse d'évolution favorable en cours d'hospitalisation sans recours à la chirurgie »).*

***Inopérant* :** *Critères pour lesquels le système échoue systématique à sélectionner des résultats pertinents et pour lesquels une stratégie de recherche est difficilement définissable (e.g. « patients consommant régulièrement de la réglisse ou ses dérivés » ou encore « des douleurs abdominales survenant au moins 1 jour par semaine durant les 3 derniers mois associé à au moins 2 des critères suivants : en relation avec la défécation, survenue associée à une modification de la fréquence ou de la consistance des selles (à type de diarrhée) »).*

***Non-Applicable (N/A)* :** *Critères ne portant pas sur des critères médicaux ou davantage assimilable à des instructions qu'à des contraintes (e.g. « Patient participant à un autre essai clinique interventionnel ayant le même objectif principal » ou encore « pour les femmes en âge de procréer, une contraception efficace [...] sera exigée pendant le traitement et pendant l'année suivant l'arrêt de celui-ci »).*

Afin de donner davantage de sens à cette classification et surtout de **donner des éléments de réponse pouvant expliquer les résultats obtenus**, **deux paramètres supplémentaires ont été observés** pour chaque **directive de recherche** :

— la source d'information ciblée pour chacune d'entre elles ;

— les différents types d'obstacles et barrières auxquelles ces dernières sont confrontées.

Chaque source d'information possède, en effet, ses propres caractéristiques (e.g. structurées, non-structurées, données numériques, données textuelles, données sémantiques, etc.). Ces spécificités en font des données plus ou moins utiles et/ou aisées à exploiter. Elles jouent, de surcroît, potentiellement, toutes un rôle informatif différent au sein du SIH.

L'observation de ces dernières présente donc un intérêt dans l'analyse de l'efficacité globale du système. Dans le cadre de cette étude, ces dernières ont été réparties en six catégories.

Définition 32 (Sources d'information potentielles d'une directive de recherche) :
Les six catégories de source d'information potentiellement ciblées pour une directive de recherche ont été définies comme suit :

Les données patients () **:** *correspondent aux données structurées relatives aux patients (e.g. son âge, son sexe, etc.).*

Les données médicales () **:** *se matérialisent essentiellement par des données de codage. Cette catégorie comprend notamment les données relatives au PMSI avec le codage CIM–10 des diagnostics posés aux patients, mais aussi celles des codages CCAM relatant les divers actes médicaux et chirurgicaux effectués sur ces derniers. Ces données constituent par nature des données structurées et sémantiques.*

Les données de séjours () **:** *regroupent évidemment les données structurées relatives à la notion de séjour (e.g. dates d'entrée et de sortie) mais également les informations relatives aux mouvements du patient au sein de l'établissement telles que les unités médicales dans lesquelles ont été effectués ces séjours.*

Les données biologiques () **:** *correspondent essentiellement aux résultats des examens biologiques (données structurées).*

Les textes cliniques () **:** *correspondent aux comptes-rendus, ordonnances, courriers, etc.. Ces données sont par nature non structurées.*

Les information externes () **:** *correspondent à des informations n'existant pas dans le SIH du CHU de Rouen et ne peuvent provenir que de sources externes.*

De même, les obstacles et barrières potentiels pouvant entraver l'efficacité de chacune des directives de recherche ont été observés. Six catégories d'obstacles ont ainsi été identifiées. À la différence des sources d'information, plusieurs limites peuvent être attribuées à une directive de recherche.

Ces limites sont définies ci-dessous :

Définition 33 (Limites potentielles d'une directive de recherche) :
Les six limites pouvant entraver l'efficacité d'une directive de recherche ont été définies comme suit :

Aucun () **:** *Pour les directives pour lesquelles aucune entrave n'a été constatée.*

Données inconsistante () **:** *Lorsque les données fournies par l'EDSS sont pertinentes mais ne sont pas suffisamment précises ou sont renseignées de manière trop aléatoire.*

Recherche imprécise () **:** *Lorsque l'information à rechercher s'avère complexe et difficile à trouver de manière précise. Typiquement, ce type d'obstacle se rencontre dans les recherches au sein des textes cliniques et/ou lors de la recherche d'un concept terminologique adapté à la pathologie recherchée.*

Limitation technique () **:** *Lorsque l'information recherchée existe pleinement au sein de l'EDSS mais que les fonctionnalités techniques fournies par le SSE_{NoSQL} et/ou* **ASIS** *ne permettent pas de requêter cette information.*

Critère subjectif () **:** *Lorsque la recherche porte sur un aspect subjectif et/ou trop générique du critère nécessitant un jugement de valeur de la part d'un professionnel de Santé.*

Communication () **:** *Lorsque l'information à rechercher ne peut être obtenue que par l'intermédiaire d'une communication avec le patient.*

Dans cette étude, le niveau de prise en charge des 95 critères d'inclusion et d'exclusion ont d'abord été colligés et analysés de manière globale. Un test statistique bilatéral de Wilcoxon (i.e. Test des rangs signés de Wilcoxon) visant à évaluer la différence de prise en charge entre critères d'inclusion et critères d'exclusion a également été réalisé.

Par la suite, les trois caractéristiques observées (viz. prises en charge, sources d'information et limites) ont été mises en correspondance afin d'identifier les différentes aptitudes et limitations de l'EDSS et des outils qui permettent d'y accéder.

8.2.3 Résultats de l'évaluation

8.2.3.1 Prise en charge globale des critères

Le nombre et le pourcentage de critères d'inclusion et d'exclusion appartenant à chaque niveau de prise en charge sont donnés dans la Table 8.2.

Niveaux	**Critères d'inclusion**			**Critères d'exclusion**			**Total**		
	n	$p(\%)$	$I_c(\%)$	n	$p(\%)$	$I_c(\%)$	n	$p(\%)$	$I_c(\%)$
Total	6	16,67	[4,5 ; 28,8]	5	8,47	[1,4 ; 15,6]	11	11,58	[5,1 ; 18,0]
Précis	3	8,33	[0 ; 17,4]	15	25,42	[14,3 ; 36,5]	18	18,95	[11,1 ; 26,8]
Partiel	6	16,67	[4,5 ; 28,8]	19	32,20	[20,3 ; 44,1]	25	26,32	[17,5 ; 35,2]
Imprécis	4	11,11	[0,8 ; 21,4]	6	10,17	[2,5 ; 17,9]	10	10,53	[4,4 ; 16,7]
Inopérant	3	8,33	[0 ; 17,4]	7	11,86	[3,6 ; 20,1]	10	10,53	[4,4 ; 16,7]
N/A	14	38,89	[23 ; 54,8]	7	11,86	[3,6 ; 20,1]	21	22,10	[13,8 ; 30,4]
Total	36	100		59	100		95	100	

TABLE 8.2 – Nombre n et pourcentage p de critères d'inclusion et d'exclusion en fonction de leurs niveaux de prise en charge par le système. Pour chaque pourcentage p, il est également donné l'intervalle de confiance I_c à 95% de ce dernier.

Compte tenu de la méthodologie employée, seulement trois des niveaux peuvent être considérés comme « contribuant » à la sélection automatique de patients : *Total*, *Précis* et *Partiel*. Trois « classes d'efficacité » du système peuvent être distinguées suivant les niveaux de prise en charge des critères :

Classe n° 2 ({*Total*, *Précis*}) : Pour les critères de niveau *Total* ou *Précis*, le système peut être pleinement considéré comme un **outil d'aide à la constitution de cohorte**.

Classe n° 1 ({*Partiel*}) : Pour les critères de niveau *Partiel*, le système peut efficacement être exploité comme un **outil de pre-screening**.[6]

Classe n° 0 ({*Imprécis*, *Inopérant*, *N/A*}) : Dans tous les autres cas, le système s'avère peu efficace et peu utile pour une sélection de patients sur des critères précis. Il s'avère alors être plus un **outil d'exploration d'information** qu'un outil de requêtage avancé.

Les résultats obtenus (Table 8.2) montrent que le système a été en capacité d'automatiser totalement ou partiellement (i.e. Classe n° 1 + Classe n° 2) la recherche de 41,67% des critères d'inclusion et 66,9% des critères d'exclusion.

Les études cliniques reposent généralement sur davantage de critères d'inclusion que d'exclusion. À titre d'exemple, sur l'ensemble des cinq études cliniques sélectionnées dans cette étude, le nombre de critères d'inclusion dépassait celui des critères d'exclusion d'en moyenne 20,14%. En d'autres termes, les conditions d'acceptation d'un patient au sein d'une étude clinique reposent sur un nombre moindre de critères. Unitairement, un critère d'inclusion joue donc un rôle plus critique vis-à-vis d'une étude clinique qu'un critère d'inclusion.

Au regard de cette considération, le pourcentage de 41,67% des critères d'inclusion pour lesquels le système s'avère profitable, semble relativement faible. Ce dernier est cependant « faussé »

6. 🇫🇷 : « Outil de présélection ou pré-filtrage »

dans le sens où les pourcentages calculés dans la Table 8.2 incluent les critères de type Non-Applicable (N/A). Comme évoqué précédemment (définition 31), ces critères ne relèvent pas du domaine d'application d'un EDS. Ils représentent 22,1% de l'ensemble des critères des cinq études et 66% d'entre eux sont des critères d'inclusion.

En les excluant, les pourcentages de critères d'inclusion et d'exclusion pour lesquels le système peut être utile (i.e. Classe n° 1 + Classe n° 2) augmentent respectivement à 68,18% et 75% pour un pourcentage total de critères de 72,97%.

De plus, en ne considérant que les niveaux *Total* et *Précis* (i.e. Classe n° 1) et en écartant le niveau *Partiel* qui requiert un post-filtrage manuel de la part d'un professionnel de santé, on constate que 40,9% des critères d'inclusion peuvent être automatisés contre 38,46% des critères d'exclusion pour un total de 39,18% de critères automatisés.

Ces résultats sont synthétisés dans la Figure 8.11.

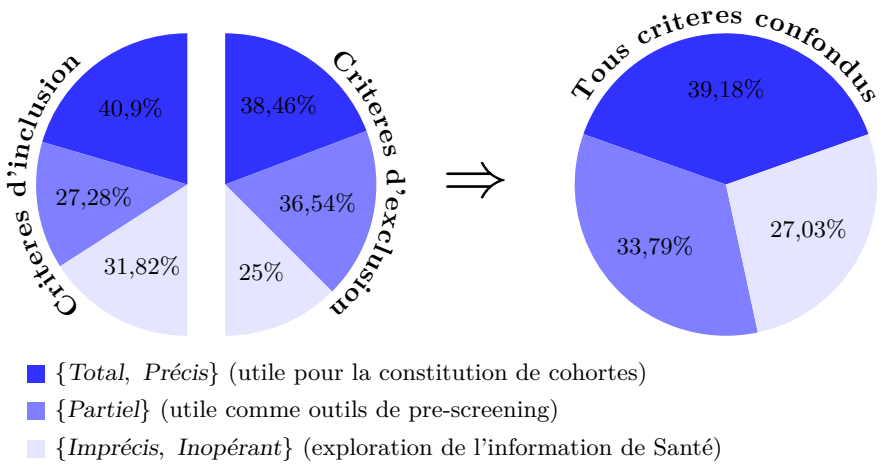


FIGURE 8.11 – Pourcentages des critères (critères Non-Applicables exclus) par classe d'utilité de l'outil : outil de constitution de cohorte, outil de pre-screening et outil d'exploration d'information.

8.2.3.2 Répartition des sources d'information

On observe, dans cette section, la distribution des sources d'information ciblées par les 95 critères des cinq études cliniques sélectionnées. Certains critères ne ciblent qu'une seule source d'information tandis que d'autres s'appuient sur diverses informations de santé et ciblent par conséquent une combinaison de ces dernières.

La Table 8.3 donne le nombre de critères appartenant à chaque niveau de prise en charge en fonction de la combinaison de sources d'information ciblées.

Niveaux																**Total**
Total	4	0	7	0	0	0	0	0	0	0	0	0	0	0	0	11
Précis	1	0	0	8	0	0	0	1	3	0	5	0	0	0	0	18
Partiel	0	2	2	6	7	0	1	0	0	1	4	0	0	2	0	25
Imprécis	0	0	0	0	5	0	0	0	0	1	1	0	1	1	1	10
Inopérant	0	0	0	0	2	3	0	0	0	0	3	1	0	1	0	10
N/A	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	21
Total	5	2	9	14	14	24	1	1	3	2	13	1	1	4	1	95

TABLE 8.3 – Nombre de critères pour chaque niveau de prise en charge en fonction de la combinaison de source d'information nécessaire pour établir sa stratégie de recherche.

La Figure 8.12 illustre, quant à elle, graphiquement la distribution des sources d'information sur les 95 critères.

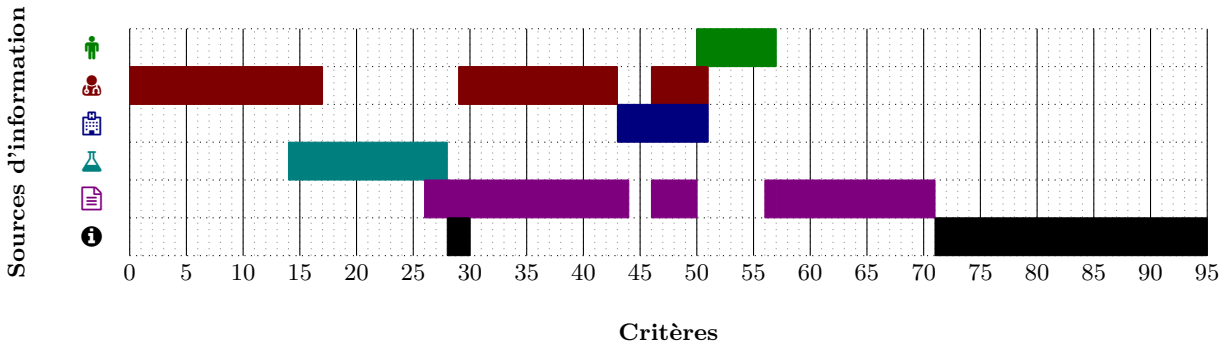


FIGURE 8.12 – Illustration graphique de la répartition des sources d'information parmi les critères d'inclusion et d'exclusion.

Globalement, on peut remarquer une distribution non uniforme des sources d'information ciblées par les critères. Les textes cliniques ( ) et les données de codage ( ) sont deux sources d'information prédominantes. Ces deux dernières sont, en effet, impliquées dans respectivement 38,95% et 37,89% des critères. La table Table 8.4 donne, en ordre décroissant, le taux d'implication de chacune des sources d'information.

Sources d'information	t	p
	37/95	(38,95%)
	36/95	(37,89%)
	26/95	(27,36%)
	14/95	(14,73%)
	8/95	(8,42%)
	7/95	(3,37%)

TABLE 8.4 – Taux t et pourcentage p d'implication de chaque source d'information dans les stratégies de recherche des 95 critères de l'étude.

Aucun des critères impliquant une combinaison de source d'information ne peut être pris en charge totalement. Afin d'établir une distinction des niveaux de prise en charge des critères en fonction des sources d'information on effectue la distinction entre les critères ciblant une, deux ou trois sources d'information. Si l'on écarte les 21 critères *N/A*, la Table 8.3 permet de plus d'établir que :

— 47/74 critères (63,51%) ne ciblent qu'une seule source d'information ;
— 20/74 critères (27,03%) ciblent deux sources d'information ;
— 7/74 critères (9,46%) ciblent trois sources d'information ;
— 36/74 critères (36,49%) ciblent une combinaison de deux ou trois sources d'information.

Afin de pouvoir calculer un niveau de prise en charge moyen des critères de chacun de ces groupes, on attribue un score de 1 à 5 à chacun des niveaux de *Inopérant* à *Total* :

$$\textit{Inopérant} \to 1,\ \textit{Imprécis} \to 2,\ \textit{Partiel} \to 3,\ \textit{Précis} \to 4,\ \textit{Total} \to 5$$

On obtient alors les niveaux de prise en charge moyen donnés dans la Figure 8.13.

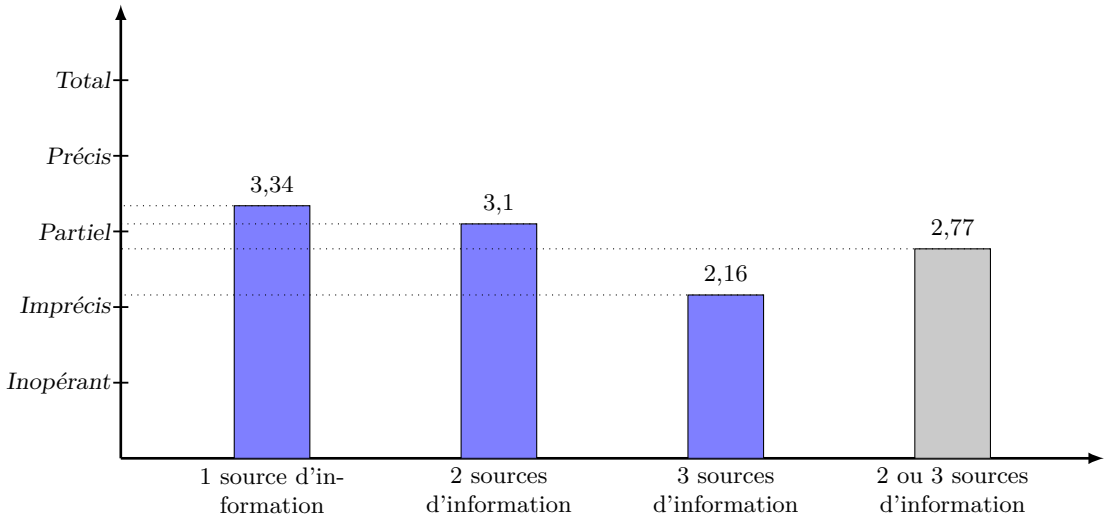


FIGURE 8.13 – Niveaux de prise en charge moyens des critères ciblant une seule source d'information, deux sources d'information, trois sources d'information ou une combinaison de sources d'information de manière générale.

Le niveau de prise en charge diminue faiblement lorsque l'on passe d'une source unique d'information à deux sources d'information. Les niveaux de prise en charge de ces deux groupes de critères se trouvent, en effet, tous deux entre *Partiel* et *Précis*. En revanche, les combinaisons de trois sources d'information font chuter le niveau de prise en charge des critères de presque un niveau.

8.2.3.3 Analyse

Les intervalles de confiance à 95% fournis dans la Table 8.2 font état d'une étendue moyenne d'environ 14,18 % (inclusion et exclusion confondus, critères de type *N/A* exclus). En d'autres termes, le faible nombre de critères considérés dans cette étude rend les pourcentages obtenus dans les deux sections précédentes peu fiables.

Ces derniers doivent par conséquent être considérés avec prudence. Il est par conséquent important d'apporter à ces valeurs quantitatives un regard davantage qualitatif.

Dans cette section, on « met en correspondance » :

— les niveaux de prise en charge des critères ;

— les sources d'information diverses qu'ils ciblent ;

— les limitations observées pour chaque directive de recherche de ces derniers.

Cette mise en correspondance vise à identifier de manière plus concrète les forces et faiblesses d'**ASIS**, du SSE_{NoSQL} et, plus généralement, de l'ensemble de l'EDSS.

La Figure 8.14 illustre cette mise en correspondance en fournissant pour chaque niveau de prise en charge :

— un diagramme en barre donnant les pourcentages des directives de recherche qui ciblent chacune des sources d'information ;

— un diagramme en barre donnant les pourcentages de chacun des types d'obstacle observés sur l'ensemble des directives de recherche.

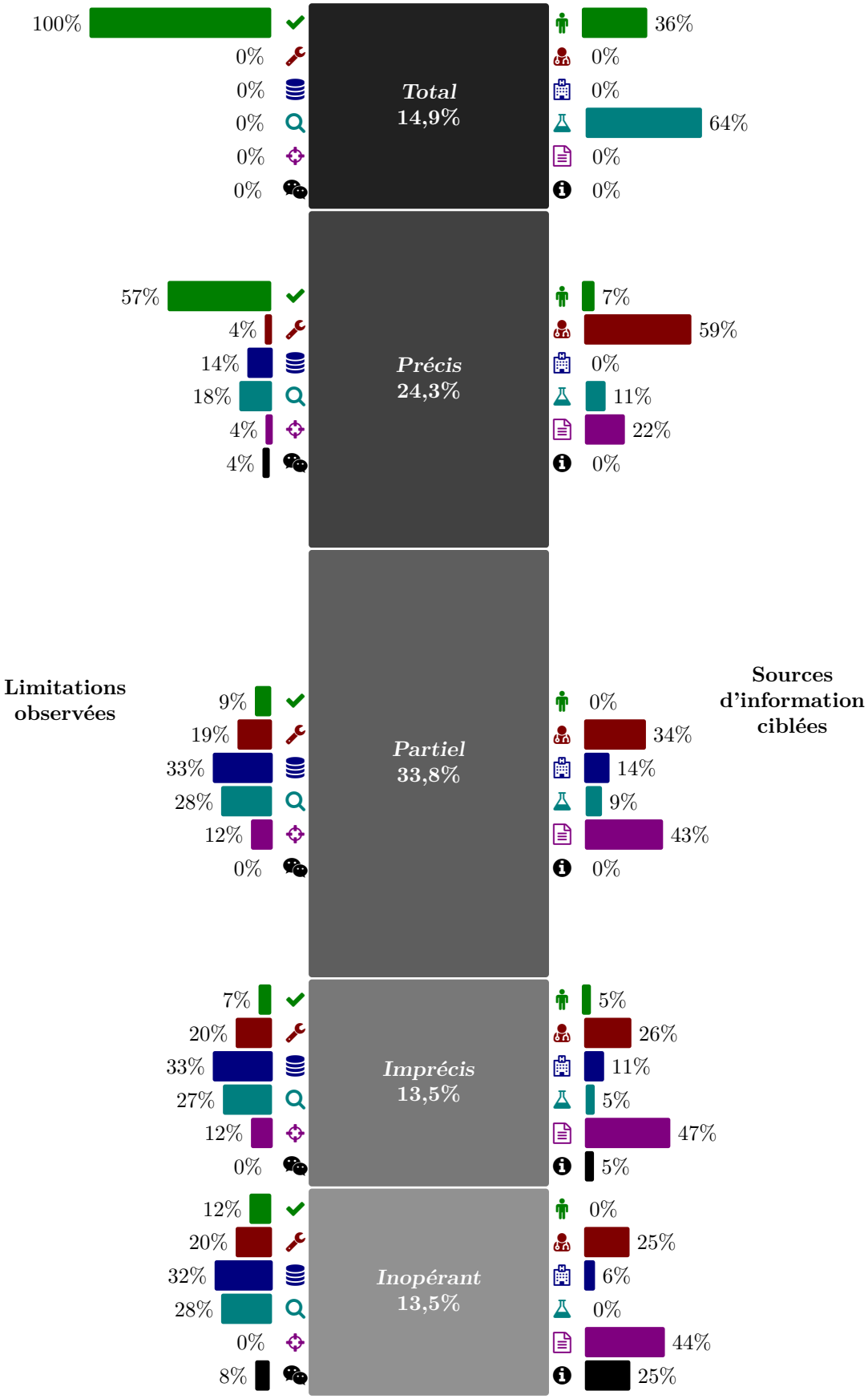


FIGURE 8.14 – Mise en correspondance des sources d'information ciblées et des limitations observées pour chaque niveau de prise en charge.

Les données patients (👤) et biologiques (🧪) sont les deux seules sources d'information ciblées par les critères de niveau *Total*.

Ces données sont des données structurées et renseignées de manière constante et rigoureuse au sein du SIH. Elles sont également basées sur une modélisation simple et fixe ne requérant pas d'interprétation.

Dans la pratique, les critères de niveau *Total* sont formulés autour de conditions précises et non ambiguës reposant essentiellement sur des contraintes portant sur des données numériques ou symboliques tels que « *Patients = 18 ans et <75 ans* » ou encore « *avec une hémoglobine glyquée entre 6,5% et 8% maximum* ».

Les critères de niveau *Précis* reposent, quant à eux, majoritairement sur la sélection de patients par l'intermédiaire d'informations contenues dans des données de codage (🏷) de diagnostics et/ou d'actes.

Ces critères ciblent le plus souvent une unique source d'information et imposent une unique caractéristique de santé aisément sélectionnable à l'aide d'un concept adéquate (e.g. « *sujet diabétique de type 1* », « *patient séropositif pour le VIH* »).

Ils peuvent néanmoins, tout de même, cibler des combinaisons de sources d'information (e.g. « *sujet âgé de 18 à 70 ans, homme ou femme ménopausée* » ⇒ données patients (👤) + textes cliniques (📄)).

L'aspect sémantique de l'EDSS joue un rôle particulièrement important dans le cadre des critères de niveau *Précis*. Il permet notamment de bénéficier des termes alternatifs des concepts et de l'annotation sémantique des textes cliniques (par exemple pour la recherche de la « *ménopause* ») mais aussi de leurs hiérarchies notamment pour les critères reposant sur des conditions de santé « génériques » (e.g. « ***hépatopathie connue active*** *[...] ou antécédent* ***d'hépatopathie*** *récente [...]* ***quelle que soit sa nature*** *[...]* »).

Que cela concerne l'annotation sémantique des textes cliniques (📄) ou les données de codage (🏷), il peut néanmoins arriver que l'information ne soit pas systématiquement renseignée dans le SIH où que les concepts disponibles ne répondent pas exactement aux exigences formulées par les critères. C'est la raison pour laquelle des obstacles de recherche imprécise (🔍) et d'inconsistance des données (🗄) peuvent être observés sur les critères de ce niveau de prise en charge.

Une observation globale de la Figure 8.14 permet de mettre en évidence une baisse du niveau de prise en charge des critères à mesure que l'exploitation des textes cliniques prend le pas sur celle des données structurées et notamment des données de codage (🏷). Cette observation corrobore notamment l'assertion communément admise dans la littérature [23, 67, 81] selon laquelle une importante partie de l'information de santé réside au sein des textes cliniques.

Cependant, dans le cadre de cette étude, elle montre également que la recherche au sein des textes cliniques reste encore un challenge technique majeur pour l'EDSS. Cette difficulté du système à réaliser une recherche efficace au sein des textes cliniques se confirme d'autant plus que 84,4% des obstacles de type « recherche imprécise » (🔍) sont attribuables à des directives de recherche ciblant des textes cliniques (🏷).

Comme mentionné précédemment, même lorsqu'un critère ne peut être géré que de manière partiel, le système peut efficacement être utilisé comme un outil de pre-screening. Le critère « *insuffisance cardiaque sévère (classe III ou IV, NYHA)* » est un exemple de critère partiellement géré.

Dans le cadre de cet exemple, l'insuffisance cardiaque peut être recherchée précisément à l'aide des données de codage (🏷) par exemple en recherchant les diagnostics codés avec « *insuffisance cardiaque* » [*I*50 (CIM–10)].

En revanche « *NYHA classe III* » et « *NYHA classe IV* » [7] ne peuvent, eux, être recherchés que dans les comptes-rendus de manière moins précise. La possibilité qu'offre cependant le langage $\mathcal{L}_{\text{graph}}$ de considérer les informations de santé comme un graphe d'entités multiples permet de

7. Le terme « *NYHA* » fait référence à une classification des stades de gravité des insuffisances cardiaques proposée par la **N**ew **Y**ork **H**eart **A**ssociation

définir de manière indépendante chacune des requêtes portant sur chacune de ces entités.

Plus concrètement, cela permet notamment à l'interface **ASIS** d'offrir la possibilité de visualiser les résultats intermédiaires des différentes directives avant d'accéder aux résultats finaux.

La Figure 8.15 illustre ce principe avec le critère précédent.

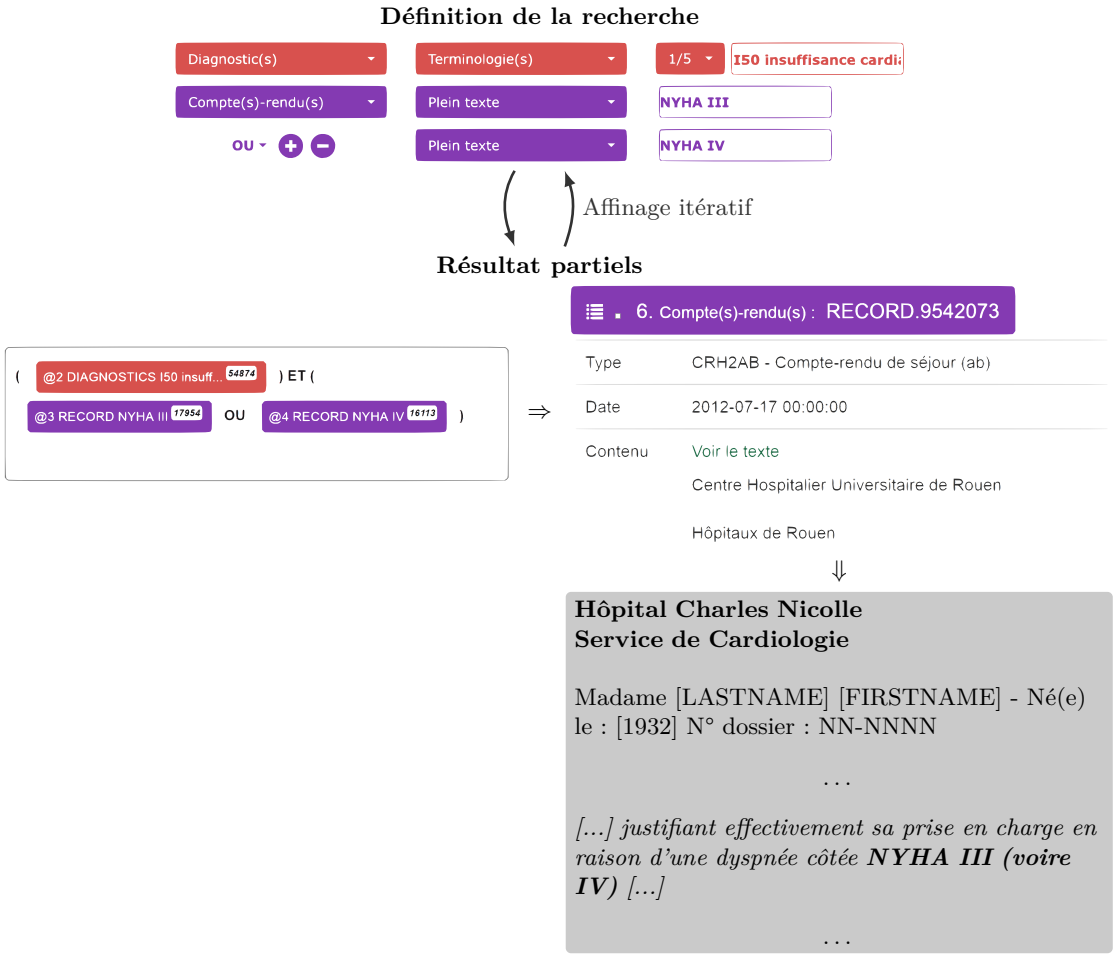


FIGURE 8.15 – Affinage itératif des stratégies de recherches rendu possible par **ASIS** et la SSE_{NoSQL}

Cet accès aux résultats partiels favorise l'utilisation de l'outil comme un outil de pre-screening. Il permet de paramétrer l'étendue et la précision des recherches et offre la possibilité d'affiner ces dernières là où cela est nécessaire.

En tout état de cause, la philosophie orientée entité sur laquelle repose la modélisation des données, le SSE_{NoSQL}, **ASIS** et finalement l'EDSS, permet une maîtrise des recherches et une pré-sélection efficace.

Même si un post-filtrage peut s'avérer tout de même nécessaire, il fournit néanmoins un moyen de réduire de manière conséquente le champ de recherche.

Cette capacité de pré-sélection est mise en évidence dans la Figure 8.16. Dans celle-ci, les recherches des diagnostics d'insuffisance cardiaque et des comptes-rendus mentionnant une insuffisance cardiaque de stade NYHA III ou IV aboutissent isolément et respectivement à 54 874 diagnostics et 27 822 comptes-rendus. En liant ces deux contraintes, l'outil permet de pré-sélectionner 1 518 séjours correspondant à 1 357 patients.

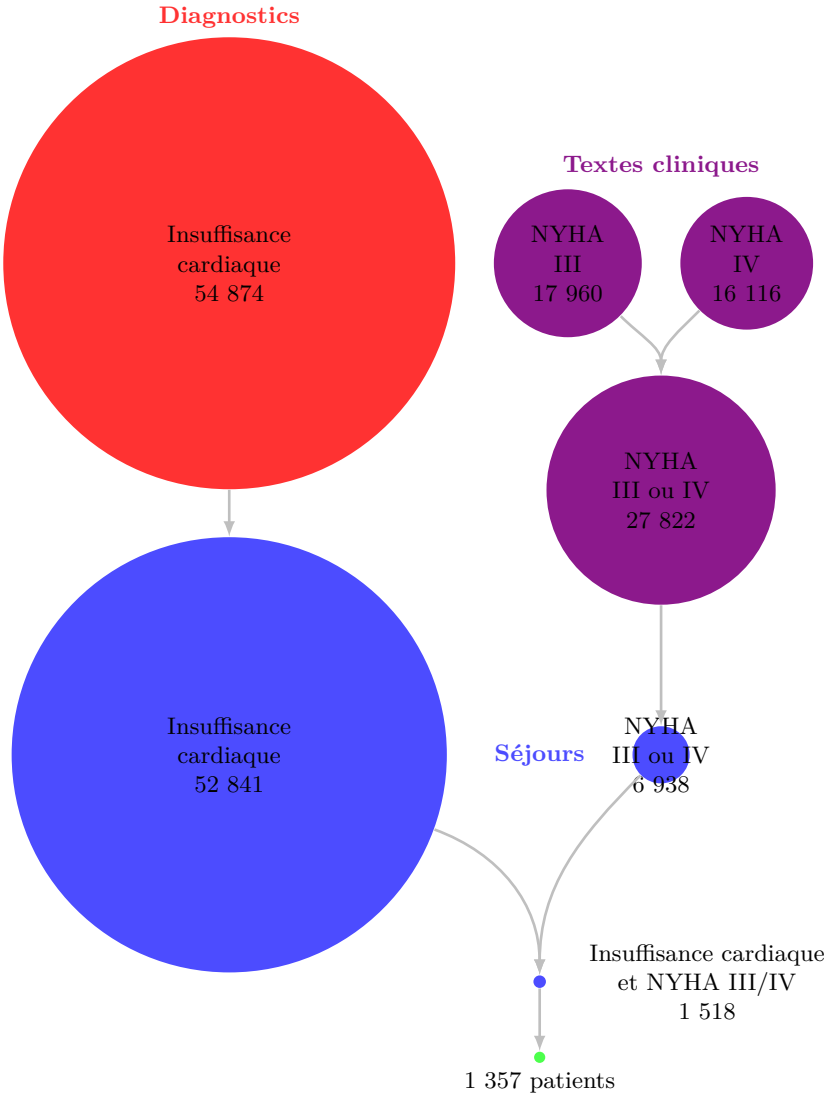


FIGURE 8.16 – Mise en évidence des capacités de pré-filtrage du système dans le cadre de la recherche du critère « *insuffisance cardiaque sévère (classe III ou IV, NYHA)* »

Conclusions et perspectives

Depuis sa création en 1995, le D2IM a toujours mené des travaux de recherches relatifs à l'information de santé. Ces derniers ont cependant majoritairement porté sur la RI documentaire et bibliographique ainsi que les terminologies et ontologies de santé. Depuis sa participation aux projets RAVEL [ANR–11–TECS–012] et SIFaDo [ANR–11–TECS–0014], le D2IM s'intéresse néanmoins d'avantage au domaine de la visualisation et de l'accès à l'information clinique. Cette évolution a notamment donné lieu à la création de l'outil RIDoPI au début des années 2010 dont l'objectif était la visualisation du DPI. Cet axe de recherche se poursuit aujourd'hui avec le projet de création de l'EDSS du CHU de Rouen. Dans ce contexte, mon travail de thèse s'attache à proposer des méthodes de RI au sein de données cliniques. Ces données hétérogènes et massives imposent du fait de leur complexité l'emploi de nouvelles méthodes et j'ai, dans cet objectif, développé deux preuves de concept de moteurs de recherche internes (i.e. d'arrière plan [8]) reposant sur l'architecture du SI du D2IM :

— Le SSE_{SQL} reposant sur une SGBDR ORACLE.

— Le SSE_{NoSQL} reposant sur NINJAC, un SGBD spécifique développé au sein du D2IM et basé sur une technologie NoSQL.

Chacun de ces deux moteurs repose sur une technologie de stockage de données différente. Néanmoins, ils exploitent tous deux un même modèle de représentation générique de l'information développé au sein du D2IM depuis plusieurs années. Ce dernier permet de représenter les informations sous la forme d'un graphe de données qui s'accommode de la complexité de l'information clinique. Afin d'exploiter pleinement cette généricité, j'ai dans le cadre de cette thèse, modélisé et implémenté, au sein des ces deux moteurs, un langage de requête logique et spécifique permettant d'accéder aux informations à l'aide d'un formalisme orienté objet.

Le SSE_{SQL} a été le point de départ de mes travaux. Bien qu'il fournisse des fonctionnalités exhaustives, ce dernier n'est aujourd'hui plus maintenu au sein du D2IM compte tenu des faibles performances qu'il offre. Il a néanmoins permis de mettre en évidence les limites des SGBDRs dans le cadre de la manipulation de données massives telles que les données cliniques d'un EDS. En outre, ce dernier a permis de répondre à certains cas d'usages notamment dans le cadre du projet RAVEL [ANR–11–TECS–012].

Le SSE_{NoSQL}, quant à lui, bien que les technologies sur lesquelles il repose aient imposé des concessions en terme de fonctionnalités de RI par rapport au SSE_{SQL}, est aujourd'hui utilisé opérationnellement comme moteur de recherche sous-jacent à divers outils de recherche d'information en Santé du système d'information du D2IM tels que :

— la RI documentaire en Santé avec l'outil DocCiSMeF ;

— la RI bibliographique avec l'outil LiSSa.

Le SSE_{NoSQL} fait également partie intégrante de l'EDSS qui repose entre autres, également, sur l'annotateur sémantique ECMT, le portail Termino-Ontologique HeTOP, l'interface graphique ASIS ou encore le SGBD NoSQL NINJAC du D2IM.

D'un point de vue technique, l'utilisation du SSE_{NoSQL} et, plus généralement, de l'ensemble de ces outils au sein de l'EDSS, a permis d'objectiver leur viabilité technologique dans le cadre d'une problématique de données massives. Le SSE_{NoSQL} accède en effet aux données par l'intermédiaire du SGBD NINJAC reposant, quant à lui, sur la solution NoSQL Infinispan orienté Clé-Valeur. Bien que performante, cette solution implique une consommation de mémoire vive

8. : « back-end »

(RAM) conséquente et, donc, l'utilisation de machines puissantes. Le D2IM est actuellement en train d'opérer une « refonte » de NINJAC afin, d'une part, de limiter sa consommation mémoire et, d'autre part, d'optimiser les temps de traitement. La nouvelle version de ce SGBD devrait remettre en cause l'utilisation d'Infinispan en raison de certaines « instabilités » constatées, notamment en ce qui concerne la persistance des données. Bien que l'exploitation d'une base de données orienté graphe ne soit pas envisagée à court terme, ces dernières présentent une alternative intéressante vis à vis du SSE_{NoSQL} compte tenu des langages de requête que ces solutions mettent à disposition et qui offrent des perspectives de recherche d'information plus étendues.

J'ai également mené une étude visant à évaluer la capacité de l'EDSS dans sa globalité à répondre à des critères d'études cliniques. Cette dernière a permis de rendre compte des apports et des limites en terme de recherche d'information de l'EDSS et, a fortiori, du SSE_{NoSQL}.

L'un des aspects fondamentaux du SSE_{NoSQL}, qui constitue en outre sa force majeure, réside dans son orientation entité. Celle-ci apporte une flexibilité à l'égard de la complexité et de l'hétérogénéité des données cliniques et rend possible une recherche générique de ces informations. En comparaison, la plupart des EDSs existants, à commencer par i2b2, STRIDE ou encore ehop, proposent des outils essentiellement orientés patient. Ceux-ci permettent généralement de visualiser des listes de patients répondants à des critères par l'intermédiaire de formulaires spécifiques générant des requêtes pré-définies. A contrario, le SSE_{NoSQL} et le langage $\mathcal{L}_{\text{<}}$ que j'ai proposé fournissent un accès à l'information plus générique en permettant d'accéder indépendamment aux diverses entités qui composent conceptuellement l'information clinique et de santé en général. En d'autres termes, la philosophie du SSE_{NoSQL} repose sur une vision des données sous formes de graphes. L'étendue des possibilités de recherche d'information du SSE_{NoSQL} est néanmoins dépendante du choix de la modélisation conceptuelle des données puisque le formalisme des $\mathcal{L}_{\text{<}}$–requêtes est basé sur cette modélisation. Dans le cadre de l'EDSS, le D2IM travaille actuellement à la conception d'un modèle conceptuel stable, apte à intégrer de manière cohérente l'ensemble des informations cliniques issues du CHU de Rouen. Ce dernier devrait constituer une base solide et concrète, susceptible de guider les futures évolutions de l'outil. En tout état de cause, la description sous forme d'un graphe de données donne toute liberté à une description sémantique cohérente et précise des informations à l'aide de concepts terminologiques et/ou Ontologiques. Celle-ci s'avère, en outre, indispensable dans un contexte de santé compte-tenu de la diversité de l'information issue de ce domaine. Le SSE_{NoSQL}, par l'intermédiaire des fonctionnalités de recherche au sein du graphe de données qu'il fournit, permet l'exploitation de cette description sémantique et participe à l'établissement d'une recherche d'information sémantique.

C'est en partie, également, cet aspect « multi-entités » qui confère à l'interface ASIS son aptitude à partitionner les recherches globales en contraintes basiques ciblant des entités d'information différentes (e.g. patient, diagnostic, séjour, analyse biologique, etc.) et à visualiser de manière indépendante les résultats de ces requêtes intermédiaires. En conséquence, la généricité avec laquelle le SSE_{NoSQL} donne un accès aux données participe à la mise en place d'une RI qui peut être affinée itérativement et à la faculté de l'EDSS de servir d'environnement de pre-screening.

L'un des atouts notables de cette approche multi-entités est également qu'elle permet de combiner de manière pertinente et précise des recherches portant à la fois sur des données structurées et non structurées. Bien qu'une partie conséquente de l'information clinique réside dans les textes cliniques (données non structurées), ces derniers ne permettent pas, à eux seuls, de répondre efficacement à la totalité des besoins d'informations en santé [23, 24].

L'étude que j'ai menée dans le cadre de ma thèse a également permis de mettre en évidence certaines limitations du SSE_{NoSQL}. Une claire diminution du nombre de critères d'inclusion et d'exclusion auquel le système a été en mesure de répondre a pu être observée en ce qui concerne les critères ciblant majoritairement des textes cliniques. D'une manière générale, la recherche au sein de données non structurées, qu'elle soit plein texte ou sémantique, reste un defi pour le SSE_{NoSQL}. De nombreuses fonctionnalités devront à l'avenir vernir compléter la recherche tex-

tuelle. D'autres travaux d'évaluation seront menés au D2IM dans les mois prochains concernant la mesure de la plus value éventuelle de l'approche sémantique vs. l'approche texte intégrale (e.g. ehop, DrWarehouse, EMERSE, etc.) qui sera réalisée par des internes de santé publique et de médecine générale. Une exploitation plus fine de la description sémantique que permet l'EDSS pourrait néanmoins améliorer les performances de RI du SSE_{NoSQL} et plus particulièrement en ce qui concerne les textes cliniques. De nombreuses mesures de similarité et de distance entre concepts ont été proposées dans la littérature par le passé [165–167]. Ces dernières permettent, entre autre, une désambiguïsation des mots du langage naturel. De plus, des travaux ont été menés afin d'exploiter ces mesures dans le cadre de données représentées sous la forme de graphes ces dernières années [168, 169]. Ensan and Du [170] proposent par exemple une méthode sémantique de RI basée sur un système de liaisons d'entités sémantiques pour former une représentation sous la forme d'un graphe de documents et de requêtes. Dans celle-ci les nœuds représentent des concepts extraits de documents et les arrêtes représentent la relation sémantique entre ces concepts. Compte tenu de la capacité du SSE_{NoSQL} à recherche des informations modélisées sous la forme d'un graphe, l'investigation de telles méthodes pourrait apporter des perspectives d'évolution inintéressantes à la problématique des textes cliniques.

Les textes cliniques regorgent également de valeurs numériques telles que la tension artérielle du patient ou encore son poids. Ces données sont d'une grande valeur informative et l'extraction ainsi que le requêtage de ces dernières permettraient de répondre à un plus large spectre de questions cliniques.

De même, le SSE_{NoSQL} ne permet pas d'effectuer de calcul algébrique à la volée. Cette fonctionnalité peut néanmoins être utile à certains cas d'usages par exemple lorsque ces derniers portent sur des critères impliquant des scores divers (e.g. Indice de Masse Corporelle, clairance de la créatinine, etc.) qui requièrent un calcul basé sur de multiples données (poids, taille, age, etc.).

Plus généralement, une annotation plus fine et plus granulaire des textes cliniques est envisagée dans un avenir proche au sein du D2IM et à laquelle le SSE_{NoSQL} devra fournir un accès. Celle-ci passera notamment par une analyse de la structure des textes cliniques qui permettra, par exemple, de séparer les différentes sections des comptes-rendus (e.g. motif d'hospitalisation, antécédents, anamnèse, examen clinique à l'entrée, etc.). De plus, dans le cadre de la thèse de Émeric DYNOMANT (co-encadré par le Pr. CANU de l'Institut National des Sciences Appliquées (INSA) de Rouen), une méthode hybride d'annotation sémantique des textes cliniques alliant du traitement automatique du langage naturel et de l'apprentissage automatique (apprentissage profond) sera développée. Une seconde thèse en co-tutelle vient également de démarrer avec l'Université de Tunis concernant un caractère bilingue « français-arabe » de l'annotateur sémantique. De plus, Mikaël DUSENNE, interne de santé publique ayant passé 18 mois à Harvard Medical School, débutera, en mai prochain, sa thèse de science en co-encadrement avec le Pr. AVILLACH (Harvard) sur une annotation sémantique bilingue « français-anglais » d'une part, et la poursuite des travaux sur l'apprentissage profond d'Emeric DYNOMANT d'autre part (en particulier sur la fonction « Patient2Vec » permettant de retrouver des patients « proches »).

Un large panel des besoins en information formulé par les professionnels de Santé implique des considérations temporelles et/ou chronologiques. Ces dernières visent sommairement à rechercher des informations relatives à des événements ayant eu lieu, avant, après ou pendant un autre ou bien à des moments définis du temps. La construction de telles contraintes au sein du SSE_{NoSQL} n'est cependant que partiellement gérée. Des contraintes temporelles portant sur des dates concrètes peuvent être formulées mais la sélection d'événement occurrent à des instants définis relativement à d'autres n'est actuellement pas possible. Par ailleurs, le SSE_{SQL} offrait, lui, davantage de possibilités mais ne permettait néanmoins pas de répondre à la totalité des fonctionnalités requises. Des pistes d'évolution du SSE_{NoSQL} ont déjà été suggérées au sein de l'équipe technique du D2IM impliquant l'ajout d'opérateur spécifique au langage $\mathcal{L}_{\text{share}}$. Certaines des relations possibles entre des intervalles de temps proposé dans l'algèbre des intervalles d'Allen devrait être, ainsi, implémenté. La viabilité théorique et technique de ces lourdes modifications devra néanmoins être évaluée.

Il faut néanmoins noter que le langage de requête $\mathcal{L}_{\text{share}}$, de par son orientation entité, permet

une sélection des événements cooccurrents basée sur une même entité organisationnelle (e.g. même séjour, même hospitalisation, même service, etc.). L'interface ASIS ne permet cependant pas d'exploiter la totalité de l'expressivité du $\mathcal{L}_{\text{  }}$. Ainsi, certaines recherches, et notamment dans le cadre de ce type de requêtes ciblant des événements cooccurrents, ne peuvent être réalisées à l'aide des formulaires proposés par ASIS. Là encore, cet outil devra faire l'objet d'évolutions afin de maximiser l'étendue de ses possibilités.

Enfin, le langage naturel reste le moyen de communication privilégié des professionnels de santé. L'apprentissage d'un langage de requêtes logique et Booléen tel que le langage $\mathcal{L}_{\text{  }}$ n'est pas envisageable pour ces derniers qui n'ont pas, dans le cadre de leur activité, les ressources en temps nécessaires. De plus, la formulation de $\mathcal{L}_{\text{  }}$–requêtes requiert une bonne connaissance du modèle conceptuel des données cliniques. Même si ce dernier dérive du processus de prise en charge des patients, il n'en reste pas moins un composant technique. Ainsi, une méthode d'accès au SSE_{NoSQL} en langage naturel apporterait une utilisabilité plus grande à cet outil. Peu d'efforts sont, a priori, nécessaires à la réadaptation de la méthode décrite dans la sous-section 7.1.4. Néanmoins, le travail manuel indispensable en amont pour assurer une couverture suffisante de cas d'usage de requêtes en langage naturel reste un problème majeur. Plusieurs perspectives d'évolution de l'outil peuvent ainsi être envisagées pour pallier à ce problème. Des méthodes d'apprentissage automatique pourraient par exemple permettre une reconnaissance plus flexible des patrons. De même, l'investigation de méthodes de TAL pourrait également être utilisée pour définir de manière plus souple des patrons de requêtes en langage naturel.

Durant cette thèse, les deux outils HeTOP et ECMT, qui constituent deux composants majeurs de l'EDSS, ont été valorisés par la Petite ou Moyenne Entreprise (PME) Alicante des Hauts de France. Les travaux réalisés dans cette thèse (i.e. le SSE_{SQL} et le SSE_{NoSQL}) seront valorisés par cette même société au cours de l'année 2019 pour commercialiser l'EDSS dans son ensemble.

En conclusion, de multiples facteurs interviennent dans l'efficacité de l'accès à l'information de Santé et, plus spécifiquement, dans l'accès sémantique à cette dernière. La complexité, la volumétrie, et l'hétérogénéité des données cliniques imposent à cette problématique globale de nombreuses contraintes en terme de modélisation de l'information, de temps de traitements et d'ergonomie des interfaces d'accès etc.. Le travail réalisé dans le cadre de cette thèse à permis de contribuer à celle-ci en fournissant des fonctionnalités de recherche d'information générique.

Bibliographie

[1] Henry J Lowe, Todd A Ferris, Penni M Hernandez, and Susan C Weber. Stride–an integrated standards-based translational research informatics platform. In **AMIA Annual Symposium Proceedings**, volume 2009, page 391. American Medical Informatics Association, 2009.

[2] Jennifer Rowley. The wisdom hierarchy : representations of the DIKW hierarchy. **Journal of Information Science**, 33(2) :163–180, feb 2007. doi: 10.1177/0165551506070706. URL https://doi.org/10.1177/0165551506070706.

[3] Chapter three - rdf and the semantic web stack. In Olivier Curé and Guillaume Blin, editors, **RDF Database Systems**, pages 41 – 80. Morgan Kaufmann, Boston, 2015. ISBN 978-0-12-799957-9. doi: https://doi.org/10.1016/B978-0-12-799957-9.00003-1. URL http://www.sciencedirect.com/science/article/pii/B9780127999579000031.

[4] Marc Cuggia. Exploitation des données massives en santé pour la recherche médicale : méthodes, outils et cas d'utilisation. Lausanne, Suisse, 04 2016. Colloque de l'Institut universitaire de médecine sociale et préventive. URL https://www.iumsp.ch/fr/node/6768.

[5] R. S. Ledley and L. B. Lusted. Reasoning foundations of medical diagnosis : Symbolic logic, probability, and value theory aid our understanding of how physicians reason. **Science**, 130(3366) :9–21, jul 1959. doi: 10.1126/science.130.3366.9. URL https://doi.org/10.1126/science.130.3366.9.

[6] Toni Hebda, Patricia Czar, and Cynthia Mascara. **Handbook of informatics for nurses and health care professionals**. Prentice Hall, 5 edition, 2012.

[7] Jeanne P. Sewell and Linda Q. Thede. **Informatics and Nursing**. Wolters Kluwer Health/Lippincott Williams & Wilkins, 3 edition, 12 2018. ISBN 9781609136956.

[8] Linda Q. Thede. **Informatics and Nursing**. 12 2018. ISBN 9780781740203.

[9] P. J. O'Connor, J. M. Sperl-Hillen, W. A. Rush, P. E. Johnson, G. H. Amundson, S. E. Asche, H. L. Ekstrom, and T. P. Gilmer. Impact of electronic health record clinical decision support on diabetes care : A randomized trial. **The Annals of Family Medicine**, 9 (1) :12–21, jan 2011. doi: 10.1370/afm.1196. URL https://doi.org/10.1370/afm.1196. DOI : 10.1370/afm.1196.

[10] Chaitanya Shivade, Preethi Raghavan, Eric Fosler-Lussier, Peter J Embi, Noemie Elhadad, Stephen B Johnson, and Albert M Lai. A review of approaches to identifying patient phenotype cohorts using electronic health records. **Journal of the American Medical Informatics Association**, 21(2) :221–230, 2013.

[11] Matthew D. Krasowski, Andy Schriever, Gagan Mathur, John L. Blau, Stephanie L. Stauffer, and Bradley A. Ford. Use of a data warehouse at an academic medical center for clinical pathology quality improvement, education, and research. **Journal of Pathology Informatics**, 6(1) :45, 2015. doi: 10.4103/2153-3539.161615. URL https://doi.org/10.4103/2153-3539.161615. DOI : 10.4103/2153-3539.161615.

[12] Kali VanLangen and Greg Wellman. Trends in electronic health record usage among US colleges of pharmacy. **Currents in Pharmacy Teaching and Learning**, 10(5) : 566–570, may 2018. doi: 10.1016/j.cptl.2018.01.010. URL https://doi.org/10.1016/j.cptl.2018.01.010. DOI : 10.1016/j.cptl.2018.01.010.

[13] DA Lindberg. Internet access to the national library of medicine. **Effective clinical practice : ECP**, 3(5) :256, 2000.

[14] William Hersh. **Information retrieval : a health and biomedical perspective**. Springer Science & Business Media, 2008.

[15] William R Hersh. Information retrieval for healthcare., 2015.

[16] Armen Yuri Gasparyan, Marlen Yessirkepov, Alexander A. Voronov, Vladimir I. Trukhachev, Elena I. Kostyukova, Alexey N. Gerasimov, and George D. Kitas. Specialist bibliographic databases. **Journal of Korean Medical Science**, 31(5) :660, 2016. doi: 10.3346/jkms.2016.31.5.660. URL https://doi.org/10.3346/jkms.2016.31.5.660.

[17] Shannon Kugley, Anne Wade, James Thomas, Quenby Mahood, Anne-Marie Klint Jørgensen, Karianne Hammerstrøm, and Nila Sathe. Searching for studies : A guide to information retrieval for campbell. **Campbell Systematic Reviews**, 2016.

[18] Gail M. Hodge. **Systems of Knowledge Organization for Digital Libr-Aries : Beyond Traditional Authority Files**. Commission on Preservation, jul 2000. ISBN 1887334769. URL https://www.xarg.org/ref/a/1887334769/.

[19] Ceri Binding and Douglas Tudhope. Kos at your service : programmatic access to knowledge organisation systems. **Journal of Digital Information**, 4(4), 2004.

[20] Klaar Vanopstal, Robert Vander Stichele, Godelieve Laureys, and Joost Buysschaert. Vocabularies and retrieval tools in biomedicine : Disentangling the terminological knot. **Journal of Medical Systems**, 35(4) :527–543, nov 2009. doi: 10.1007/s10916-009-9389-z. URL https://doi.org/10.1007/s10916-009-9389-z.

[21] S. J. Darmoni, B. Thirion, J. P. Leroy, M. Douyère, B. Lacoste, C. Godard, I. Rigolle, M. Brisou, S. Videau, E. Goupy, J. Piot, M. Quéré, S. Ouazir, and H. Abdulrab. Doc'cismef : a search tool based on "encapsulated" mesh thesaurus. **Medinfo**, 10(Pt 1) :314–318, 2001.

[22] Nicolas Griffon, Matthieu Schuers, and Stéfan J. Darmoni. Littérature scientifique en santé (LiSSa) : une alternative à l'anglais ? **La Presse Médicale**, 45(11) :955–956, nov 2016. doi: 10.1016/j.lpm.2016.11.001. URL https://doi.org/10.1016/j.lpm.2016.11.001.

[23] Preethi Raghavan, James L. Chen, Eric Fosler-Lussier, and Albert M. Lai. How essential are unstructured clinical narratives and information fusion to clinical trial recruitment ? **AMIA Summits on Translational Science Proceedings**, 2014 :218, 2014.

[24] Sijia Liu, Yanshan Wang, Andrew Wen, Liwei Wang, Na Hong, Feichen Shen, Steven Bedrick, William Hersh, and Hongfang Liu. Create : Cohort retrieval enhanced by analysis of text from electronic health records using omop common data model. **arXiv preprint arXiv :1901.07601**, 2019.

[25] Tao Huang, Liang Lan, Xuexian Fang, Peng An, Junxia Min, and Fudi Wang. Promises and challenges of big data computing in health sciences. **Big Data Research**, 2(1) :2–11, mar 2015. doi: 10.1016/j.bdr.2015.02.002. URL https://doi.org/10.1016/j.bdr.2015.02.002.

[26] Chuanren Liu, Fei Wang, Jianying Hu, and Hui Xiong. Temporal phenotyping from longitudinal electronic health records. In **Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD**

'15. ACM Press, 2015. doi: 10.1145/2783258.2783352. URL https://doi.org/10.1145/2783258.2783352.

[27] Xiaoshu Wang, Robert Gorlitsky, and Jonas S Almeida. From XML to RDF : how semantic web technologies will change the design of 'omic' standards. **Nature Biotechnology**, 23(9) :1099–1103, sep 2005. doi: 10.1038/nbt1139. URL https://doi.org/10.1038/nbt1139.

[28] Ahmed-Diouf Dirieh Dibad. **Recherche d'information multi-terminologique - Application au Dossier patient Informatisé**. PhD thesis, University of Rouen, France and University of Djibouti, 2012. URL http://www.chu-rouen.fr/cismef/wp/wp-content/uploads/2017/01/theseDIRIEHDIBADAumedDiouf_LITISCISMeF_VFF.pdf.

[29] Chloé Cabot. **Recherche d'information clinomique au sein du Dossier Patient Informatisé : modélisation, implantation et évaluation**. PhD thesis, Université de Rouen, 2017.

[30] Julien Grosjean. **Modélisation, réalisation et évaluation d'un portail multi-terminologique multi-discipline, multi-lingue (3M) dans le cadre de la Plateforme d'Indexation Régionale (PlaIR)**. PhD thesis, Université de Rouen, 2014.

[31] B. L. Humphreys, A. T. McCray, and D. A. B. Lindberg. The unified medical language system. **Yearbook of Medical Informatics**, 02(01) :41–51, aug 1993. doi: 10.1055/s-0038-1637976. URL https://doi.org/10.1055/s-0038-1637976. DOI : 10.1055/s-0038-1637976.

[32] Comité Technique 46 Sous-Comité 9 (ISO/TC 46/SC 9) Organisation Internationale de Normalisation. ISO 25964-1 Information and documentation – Thesauri and interoperability with other vocabularies – Part 1 : Thesauri for information retrieval. Technical reports, International Organization for Standardization, 2011. URL https://www.iso.org/standard/53657.html.

[33] Cui Tao, Jyotishman Pathak, Harold R. Solbrig, Wei-Qi Wei, and Christopher G. Chute. Terminology representation guidelines for biomedical ontologies in the semantic web notations. **Journal of Biomedical Informatics**, 46(1) :128–138, feb 2013. doi: 10.1016/j.jbi.2012.09.003. URL https://doi.org/10.1016/j.jbi.2012.09.003.

[34] Frantz Thiessard, Fleur Mougin, Gayo Diallo, Vianney Jouhet, Sébastien Cossin, Nicolas Garcelon, Boris Campillo-Gimenez, Wassim Jouini, Julien Grosjean, Philippe Massari, et al. Ravel : Retrieval and visualization in electronic health records. In **Medical Informatics Europe (MIE) 2012**, pages 194–198, 2012.

[35] M. Dupuch, Frédérique Segond, André Bittar, Luca Dini, Lina F. Soualmia, Stéfan J. Darmoni, Quentin Gicquel, and Marie-Hélène Metzger. Separate the grain from the chaff : make the best use of language and knowledge technologies to model textual medical data extracted from electronic health records. In **proceedings of the 6th Language & Technology Conference**, 2013.

[36] Chloé Cabot, Lina F. Soualmia, Badisse Dahamna, and Stéfan J. Darmoni. Sibm at clef ehealth evaluation lab 2016 : Extracting concepts in french medical texts with ecmt and cimind. In **2016 Conference and Labs of the Evaluation Forum, CLEF**, pages 47–60, 2016. URL http://ceur-ws.org/Vol-1609/16090047.pdf.

[37] P. Massari, I. Smuraga, L. Froment, S. Boudehent, P. Czernichow, J. Streiff, M. Baldenweck, and P. Hecketsweiler. Application de gestion des dossiers patients du c.h.u. de rouen (diamant). mise en place et évaluation de l'utilisation. In **Cinquièmes Journées Francophones d'Informatique Médicale, Dossier Patient, Codage et Langages Médicaux**, Genève, Suisse, Juin 1994.

[38] Kristiina Häyrinen, Kaija Saranto, and Pirkko Nykänen. Definition, structure, content, use and impacts of electronic health records : A review of the research literature. **International Journal of Medical Informatics**, 77(5) :291 – 304, 2008. ISSN 1386-5056. doi: http://dx.doi.org/10.1016/j.ijmedinf.2007.09.001. URL `http://www.sciencedirect.com/science/article/pii/S1386505607001682`.

[39] Comité Technique 251 (ISO/TC 215) Organisation Internationale de Normalisation. ISO/TR 20514 Health Informatics – Electronic Health Record – Definition, Scope, and Context. Technical reports, International Organization for Standardization, 2005. URL `https://www.iso.org/obp/ui/#iso:std:iso:tr:20514:ed-1:v1:en`.

[40] A Hoerbst and E Ammenwerth. Electronic health records : A systematic review on quality requirements. **Methods of Information in Medicine**, 49(4) :320–336, 2010.

[41] Marion J Ball, NCMN Carla Smith, and Richard S Bakalar. Personal health records : empowering consumers. **J Healthc Inf Manag**, 21(1) :77, 2007.

[42] Dominique Pon and Annelore Coury. Stratégie de transformation du système de santé, rapport final, accélérer le virage numérique, 09 2018. URL `https://solidarites-sante.gouv.fr/IMG/pdf/masante2022_rapport_virage_numerique.pdf`.

[43] Nir Menachemi and Taleah H Collum. Benefits and drawbacks of electronic health record systems. **Risk management and healthcare policy**, 4 :47, 2011.

[44] N. Griffon, M. Schuers, M. Joulakian, M. Bubenheim, J.-P. Leroy, and S.J. Darmoni. Physician satisfaction with transition from CPOE to paper-based prescription. **International Journal of Medical Informatics**, 103 :42–48, jul 2017. doi: 10.1016/j.ijmedinf.2017.04.007. URL `https://doi.org/10.1016/j.ijmedinf.2017.04.007`.

[45] Richard Hillestad, James Bigelow, Anthony Bower, Federico Girosi, Robin Meili, Richard Scoville, and Roger Taylor. Can electronic medical record systems transform health care ? potential health benefits, savings, and costs. **Health Affairs**, 24(5) :1103–1117, sep 2005. doi: 10.1377/hlthaff.24.5.1103. URL `https://doi.org/10.1377/hlthaff.24.5.1103`.

[46] Dipak Kalra. Electronic health record standards. **IMIA Yearbook 2006 : Assessing Information - Technologies for Health**, 2006.

[47] C Peter Waegemann. Ehr vs. cpr vs. emr. **Healthcare Informatics Online**, 1 :1–4, 2003.

[48] R. Smith. What clinical information do doctors need ? **BMJ**, 313(7064) :1062–1068, oct 1996. doi: 10.1136/bmj.313.7064.1062. URL `https://doi.org/10.1136/bmj.313.7064.1062`.

[49] CP Waegemann. Medical record institute's survey of electronic health record trends and usage. In **Toward an Electronic Health Record Europe**, volume 99, pages 147–158, 1999.

[50] Assurance Maladie. Classification commune des actes médicaux. **Disponible sur https ://www.ameli.fr/accueil-de-la-ccam/index.php**, 2015.

[51] WHO World Health Organization. **International Statistical Classification of Diseases and Related Health Problems : 10th Revision (ICD-10)**. World Health Organization, dec 2015. ISBN 9241549165. URL `https://www.xarg.org/ref/a/9241549165/`.

[52] Kevin Donnelly. Snomed-ct : The advanced terminology and coding system for ehealth. **Studies in health technology and informatics**, 121 :279–90, 02 2006.

[53] Comité Technique 251 (ISO/TC 215) Organisation Internationale de Normalisation and Comité Technique 251 (CEN/TC 251) Comité Européen de Normalisation. ISO 13606-5 : Health Informatics – Electronic Record Communication. International standard, International Organization for Standardization, 2010.

[54] Mirjana Ivanović and Zoran Budimac. An overview of ontologies and data resources in medical domains. **Expert Systems with Applications**, 41(11) :5158–5166, sep 2014. doi: 10.1016/j.eswa.2014.02.045. URL `https://doi.org/10.1016/j.eswa.2014.02.045`.

[55] Comité Technique 251 (ISO/TC 215) Organisation Internationale de Normalisation. ISO/TS 29585 Health informatics – Deployment of a clinical data warehouse. Technical specifications, International Organization for Standardization, 2010. URL `https://www.iso.org/obp/ui/fr/#iso:std:iso:ts:29585:ed-1:v1:en`.

[56] Mike Cottle, Waco Hoover, Shadaab Kanwal, Marty Kohn, Trevor Strome, and N Treister. Transforming health care through big data strategies for leveraging big data in the health care industry. **Institute for Health Technology Transformation, http ://ihealthtran. com/big-data-in-healthcare**, 2013.

[57] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare : promise and potential. **Health information science and systems**, 2(1) :3, 2014.

[58] John Gantz and 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(2012) :1–16, 2012.

[59] Marie Ndangang, Julien Grosjean, Romain Lelong, Badisse Dahamna, Ivan Kergourlay, Nicolas Griffon, and Stéfan J. Darmoni. Terminology coverage from semantic annotated health documents. **Studies in Health Technology and Informatics**, 255 (Decision Support Systems and Education) :20–24, 2018. ISSN 0926-9630. doi: 10.3233/978-1-61499-921-8-20. URL `http://doi.org/10.3233/978-1-61499-921-8-20`.

[60] Sullivan Frost. Drowning in big data ? reducing information technology complexities and costs for healthcare organizations, 2015.

[61] Lama El Sarraj, Sophie Rodier, and Bernard Espinasse. Entrepôt de données autour du pmsi pour le pilotage d'établissements hospitaliers. **Techniques Hospitalières : la revue des techniciens de la santé**, (729) :49–52, 2011.

[62] N. Malafaye, D. Demoulin, P. Mailhe, M. Morell, D. Pellecuer, and C. Dunoyer. Mise en place et exploitation d'un entrepôt de données au département d'information médicale du {CHU} de montpellier, france. **Revue d'Épidémiologie et de Santé Publique**, 66, Supplement 1 :S26 –, 2018. ISSN 0398-7620. doi: https://doi.org/10.1016/j.respe.2018.01.055. URL `https://www.sciencedirect.com/science/article/pii/S0398762018300592`. Colloque Adelf-Emois - Montpellier, 29 et 30 mars 2018.

[63] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient : an unsupervised representation to predict the future of patients from the electronic health records. **Scientific reports**, 6 :26094, 2016.

[64] George Hripcsak, Jon D Duke, Nigam Shah, Christian G Reich, Vojtech Huser, Martijn J Schuemie, Marc A Suchard, Rae Woong Park, Ian Chi Kei Wong, Peter Rijnbeek, Johan Lei, Nicole Pratt, Niklas Norén, Yu-Chuan Li, Paul E Stang, David Madigan, and Patrick B Ryan. Observational health data sciences and informatics (ohdsi) : Opportunities for observational researchers. **Studies in health technology and informatics**, 216 :574–8, 08 2015.

[65] Pierre Heudel, Alain Livartowski, Patrick Arveux, Eddy Willm, and Christophe Jamain. ConSoRe : un outil permettant de rentrer dans le monde du big data en santé. **Bulletin**

du Cancer, 103(11) :949–950, nov 2016. doi: 10.1016/j.bulcan.2016.10.001. URL https://doi.org/10.1016/j.bulcan.2016.10.001.

[66] Denis Delamarre, Guillaume Bouzille, Kevin Dalleau, Denis Courtel, and Marc Cuggia. Semantic integration of medication data into the ehop clinical data warehouse. **Studies in health technology and informatics**, 210 :702–706, 2015.

[67] Nicolas Garcelon, Antoine Neuraz, Rémi Salomon, Hassan Faour, Vincent Benoit, Arthur Delapalme, Arnold Munnich, Anita Burgun, and Bastien Rance. A clinician friendly data warehouse oriented toward narrative reports : Dr. warehouse. **Journal of Biomedical Informatics**, 80 :52–63, apr 2018. doi: 10.1016/j.jbi.2018.02.019. URL https://doi.org/10.1016/j.jbi.2018.02.019.

[68] Shawn N Murphy, Michael E Mendis, David A Berkowitz, Isaac Kohane, and Henry C Chueh. Integration of clinical and genetic data in the i2b2 architecture. In **AMIA Annual Symposium Proceedings**, volume 2006, page 1040. American Medical Informatics Association, 2006.

[69] Shawn N Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C Chueh, Susanne Churchill, and Isaac Kohane. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). **Journal of the American Medical Informatics Association**, 17(2) :124–130, 2010.

[70] Vikrant G Deshmukh, Stéphane M Meystre, and Joyce A Mitchell. Evaluating the informatics for integrating biology and the bedside system for clinical research. **BMC medical research methodology**, 9(1) :70, 2009.

[71] Zapletal Eric, Rodon Nicolas, Grabar Natalia, and Degoulet Patrice. Methodology of integration of a clinical data warehouse with a clinical information system : the hegp case. **Studies in Health Technology and Informatics**, 160(MEDINFO 2010) :193–197, 2010. ISSN 0926-9630. doi: 10.3233/978-1-60750-588-4-193. URL http://doi.org/10.3233/978-1-60750-588-4-193.

[72] et al Wack, Maxime. **Installation d'un entrepôt De données Cliniques Pour La Recherche Au CHRU De Nancy : déploiement Technique, intégration Et Gouvernance Des données.** PhD thesis, Université de Lorraine, 2017.

[73] Informatics for Integrating Biology and the Bedside. **i2b2 Cell Messaging Data Repository (CRC) Cell**. Partners HealthCare, 1.7.08–004 edition. https ://www.i2b2.org.

[74] Ralph Kimball. The data warehouse toolkit : practical techniques for building dimensional data warehouse. **NY : John Willey & Sons**, 248(4), 1996.

[75] Prakash M Nadkarni. Qav : querying entity-attribute-value metadata in a biomedical database. **Computer methods and programs in biomedicine**, 53(2) :93–103, 1997.

[76] Riccardo Bellazzi, Marco Masseroli, Shawn Murphy, Amnon Shabo, and Paolo Romano. Clinical bioinformatics : challenges and opportunities, 2012.

[77] S. Liu, Wei Ma, R. Moore, V. Ganesan, and S. Nelson. RxNorm : prescription for electronic drug information exchange. **IT Professional**, 7(5) :17–23, sep 2005. doi: 10.1109/mitp.2005.122. URL https://doi.org/10.1109/mitp.2005.122.

[78] C. G. Chute, S. A. Beck, T. B. Fisk, and D. N. Mohr. The enterprise data trust at mayo clinic : a semantically integrated warehouse of biomedical data. **Journal of the American Medical Informatics Association**, 17(2) :131–135, feb 2010. doi: 10.1136/jamia.2009.002691. URL https://doi.org/10.1136/jamia.2009.002691. DOI : 10.1136/jamia.2009.002691.

[79] Hai Hu, Mick Correll, Leonid Kvecher, Michelle Osmond, Jim Clark, Anthony Bekhash, Gwendolyn Schwab, De Gao, Jun Gao, Vladimir Kubatin, Craig D. Shriver, Jeffrey A. Hooke, Larry G. Maxwell, Albert J. Kovatich, Jonathan G. Sheldon, Michael N. Liebman, and Richard J. Mural. DW4TR : A data warehouse for translational research. **Journal of Biomedical Informatics**, 44(6) :1004–1019, dec 2011. doi: 10.1016/j.jbi.2011.08.003. URL https://doi.org/10.1016/j.jbi.2011.08.003. DOI : 10.1016/j.jbi.2011.08.003.

[80] Karsten U Kortüm, Michael Müller, Christoph Kern, Alexander Babenko, Wolfgang J Mayer, Anselm Kampik, Thomas C Kreutzer, Siegfried Priglinger, and Christoph Hirneiss. Using electronic health records to build an ophthalmologic data warehouse and visualize patients' data. **American journal of ophthalmology**, 178 :84–93, 2017.

[81] Marc Cuggia, Nicolas Garcelon, Boris Campillo-Gimenez, Thomas Bernicot, Jean-François Laurent, Etienne Garin, André Happe, and Régis Duvauferrier. Roogle : an information retrieval engine for clinical data warehouse. In **MIE**, pages 584–588, 2011.

[82] C. Percy, A Fritz, A. Jack, S. Shanmugarathan, L. Sobin, D.M. Parkin, and S. Whelan. **Classification internationale des maladies pour l'oncologie (CIM-O-3) (French Edition)**. World Health Organization, 2009. ISBN 9242545341. URL https://www.amazon.com/Classification-internationale-maladies-loncologie-French/dp/9242545341?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=9242545341.

[83] Nicolas Garcelon, Antoine Neuraz, Vincent Benoit, Rémi Salomon, and Anita Burgun. Improving a full-text search engine : the importance of negation detection and family history context to identify cases in a biomedical data warehouse. **Journal of the American Medical Informatics Association**, page ocw144, oct 2016. doi: 10.1093/jamia/ocw144. URL https://doi.org/10.1093/jamia/ocw144.

[84] Marcin Mazurek. Applying NoSQL databases for operationalizing clinical data mining models. In **Communications in Computer and Information Science**, pages 527–536. Springer International Publishing, 2014. doi: 10.1007/978-3-319-06932-6_51. URL https://doi.org/10.1007/978-3-319-06932-6_51.

[85] Ken Ka-Yin Lee, Wai-Choi Tang, and Kup-Sze Choi. Alternatives to relational database : Comparison of NoSQL and XML approaches for clinical data storage. **Computer Methods and Programs in Biomedicine**, 110(1) :99–109, apr 2013. doi: 10.1016/j.cmpb.2012.10.018. URL https://doi.org/10.1016/j.cmpb.2012.10.018.

[86] Siri Krishan Wasan, Vasudha Bhatnagar, and Harleen Kaur. The impact of data mining techniques on medical diagnostics. **Data Science Journal**, 5 :119–126, 2006. doi: 10.2481/dsj.5.119. URL https://doi.org/10.2481/dsj.5.119.

[87] Kent Allen, Madeline M Berry, Fred U Luehrs Jr, and James W Perry. Machine literature searching viii. operational criteria for designing information retrieval systems. **American Documentation (pre-1986)**, 6(2) :93, 1955.

[88] Robert Mayo Hayes. **Information Storage and Retrieval : Tools, Elements, Theories**. New York : Wiley, 1963.

[89] Chaim Zins. Conceptual approaches for defining data, information, and knowledge. **Journal of the American Society for Information Science and Technology**, 58(4) : 479–493, 2007. doi: 10.1002/asi.20508. URL https://doi.org/10.1002/asi.20508.

[90] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. **Modern Information Retrieval : The Concepts and Technology behind Search (2nd Edition) (ACM Press Books)**. Addison-Wesley Professional, 2011. ISBN 0321416910. URL https://www.amazon.com/Modern-Information-Retrieval-Concepts-Technology/dp/0321416910?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0321416910.

[91] William Bruce Frakes and Ricardo Baeza-Yates. **Information retrieval : Data structures & algorithms**, volume 331. Prentice Hall Englewood Cliffs, NJ, 1992.

[92] KAREN SPARCK JONES. A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. **Journal of Documentation**, 28 (1) :11–21, jan 1972. doi: 10.1108/eb026526. URL https://doi.org/10.1108/eb026526.

[93] G. SALTON and C.S. YANG. ON THE SPECIFICATION OF TERM VALUES IN AUTOMATIC INDEXING. **Journal of Documentation**, 29(4) :351–372, apr 1973. doi: 10.1108/eb026562. URL https://doi.org/10.1108/eb026562.

[94] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. **Communications of the ACM**, 18(11) :613–620, nov 1975. doi: 10.1145/361219.361220. URL https://doi.org/10.1145/361219.361220.

[95] G. Salton. **The SMART Retrieval System—Experiments in Automatic Document Processing**. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[96] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. **IBM Journal of Research and Development**, 1(4) :309–317, oct 1957. doi: 10.1147/rd.14.0309. URL https://doi.org/10.1147/rd.14.0309.

[97] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. **Information Processing & Management**, 24(5) :513–523, jan 1988. doi: 10.1016/0306-4573(88)90021-0. URL https://doi.org/10.1016/0306-4573(88)90021-0.

[98] Ian H Witten, Ian H Witten, Alistair Moffat, Timothy C Bell, Timothy C Bell, and Timothy C Bell. **Managing gigabytes : compressing and indexing documents and images**. Morgan Kaufmann, 1999.

[99] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. **Journal of the American Society for Information Science**, 27(3) :129–146, may 1976. doi: 10.1002/asi.4630270302. URL https://doi.org/10.1002/asi.4630270302.

[100] W.B. CROFT and D.J. HARPER. USING PROBABILISTIC MODELS OF DOCUMENT RETRIEVAL WITHOUT RELEVANCE INFORMATION. **Journal of Documentation**, 35(4) :285–295, apr 1979. doi: 10.1108/eb026683. URL https://doi.org/10.1108/eb026683.

[101] Ronald A Jydstrup and Malvern J Gross. Cost of information handling in hospitals. **Health services research**, 1(3) :235, 1966.

[102] Joseph J Mamlin and Duke H Baker. Combined time-motion and work sampling study in a general medicine clinic. **Medical care**, pages 449–456, 1973.

[103] D. L Sackett, W. M C Rosenberg, J A M. Gray, R B. Haynes, and W S. Richardson. Evidence based medicine : what it is and what it isn't. **BMJ**, 312(7023) :71–72, jan 1996. doi: 10.1136/bmj.312.7023.71. URL https://doi.org/10.1136/bmj.312.7023.71.

[104] Sharon E. MD Straus, Paul MRCGP FRACGP PhD Glasziou, W. Scott MD Richardson, and R. Brian MD Haynes. **Evidence-Based Medicine : How to Practice and Teach EBM**. Elsevier, 2018. ISBN 0702062960. URL https://www.amazon.com/Evidence-Based-Medicine-How-Practice-Teach/dp/0702062960?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0702062960.

[105] Jennifer Frankovich, Christopher A Longhurst, and Scott M Sutherland. Evidence-based medicine in the emr era. **N Engl J Med**, 365(19) :1758–1759, 2011.

[106] M. H. Coletti and H. L. Bleich. Medical subject headings used to search the biomedical literature. **Journal of the American Medical Informatics Association**, 8(4) :317–323, jul 2001. doi: 10.1136/jamia.2001.0080317. URL https://doi.org/10.1136/jamia.2001.0080317.

[107] Juan J. Manríquez. A highly sensitive search strategy for clinical trials in literatura latino americana e do caribe em ciências da saúde (LILACS) was developed. **Journal of Clinical Epidemiology**, 61(4) :407–411, apr 2008. doi: 10.1016/j.jclinepi.2007.06.009. URL https://doi.org/10.1016/j.jclinepi.2007.06.009.

[108] J Wyatt. Medical informatics, artefacts or science ? **Methods of information in medicine**, 35(03) :197–200, 1996.

[109] J C Wyatt. Basic concepts in medical informatics. **Journal of Epidemiology & Community Health**, 56(11) :808–812, nov 2002. doi: 10.1136/jech.56.11.808. URL https://doi.org/10.1136/jech.56.11.808.

[110] Edward H. Shortliffe and James J. Cimino, editors. **Biomedical Informatics**. Springer London, 2014. doi: 10.1007/978-1-4471-4474-8. URL https://doi.org/10.1007/978-1-4471-4474-8.

[111] J. W Ely. A taxonomy of generic clinical questions : classification study. **BMJ**, 321 (7258) :429–432, aug 2000. doi: 10.1136/bmj.321.7258.429. URL https://doi.org/10.1136/bmj.321.7258.429.

[112] W. R. Hersh. Factors associated with success in searching MEDLINE and applying evidence to answer clinical questions. **Journal of the American Medical Informatics Association**, 9(3) :283–293, may 2002. doi: 10.1197/jamia.m0996. URL https://doi.org/10.1197/jamia.m0996.

[113] Minsuk Lee, James Cimino, Hai Ran Zhu, Carl Sable, Vijay Shanker, John Ely, and Hong Yu. Beyond information retrieval—medical question answering. **AMIA Annu Symp Proc**, pages 469–473, 02 2006.

[114] Sadaf Aslam and Patricia Emmanuel. Formulating a researchable question : A critical step for facilitating good clinical research. **Indian journal of sexually transmitted diseases**, 31(1) :47, 2010.

[115] S. T. Rosenbloom, J. C. Denny, H. Xu, N. Lorenzi, W. W. Stead, and K. B. Johnson. Data from clinical notes : a perspective on the tension between structure and flexible documentation. **Journal of the American Medical Informatics Association**, 18 (2) :181–186, mar 2011. doi: 10.1136/jamia.2010.007237. URL https://doi.org/10.1136/jamia.2010.007237.

[116] Ivo D. Dinov. Methodological challenges and analytic opportunities for modeling and interpreting big healthcare data. **GigaScience**, 5(1), feb 2016. doi: 10.1186/s13742-016-0117-6. URL https://doi.org/10.1186/s13742-016-0117-6.

[117] Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. Learning a health knowledge graph from electronic medical records. **Scientific Reports**, 7(1), jul 2017. doi: 10.1038/s41598-017-05778-z. URL https://doi.org/10.1038/s41598-017-05778-z.

[118] Koki Tsuyuzaki and Itoshi Nikaido. Biological systems as heterogeneous information networks : a mini-review and perspectives. **arXiv preprint arXiv :1712.08865**, 2017.

[119] John F. Sowa. **Conceptual Structures : Information Processing in Mind and Machine (SYSTEMS PROGRAMMING SERIES)**. Addison-Wesley, 1983. ISBN 0201144727. URL https://www.amazon.com/

Conceptual-Structures-Information-Processing-PROGRAMMING/dp/0201144727?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0201144727.

[120] Michel Chein and Marie-Laure Mugnier. **Graph-based Knowledge Representation : Computational Foundations of Conceptual Graphs (Advanced Information and Knowledge Processing)**. Springer, 2008. URL https://www.amazon.com/Graph-based-Knowledge-Representation-Computational-Foundations-ebook/dp/B00ANF8U0O?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B00ANF8U0O.

[121] Ines Bannour, Haifa Zargayouna, and Adeline Nazarenko. Modèle unifié pour la recherche d'information sémantique. In **IC2016 : Ingénierie des Connaissances**, 2016.

[122] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. **Scientific american**, 284(5) :28–37, 2001.

[123] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing ? **International Journal of Human-Computer Studies**, 43(5-6) :907–928, nov 1995. doi: 10.1006/ijhc.1995.1081. URL https://doi.org/10.1006/ijhc.1995.1081.

[124] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM : A benchmark for OWL knowledge base systems. **Journal of Web Semantics**, 3(2-3) :158–182, oct 2005. doi: 10.1016/j.websem.2005.06.005. URL https://doi.org/10.1016/j.websem.2005.06.005.

[125] Christian Bizer and Andreas Schultz. The berlin SPARQL benchmark. **International Journal on Semantic Web and Information Systems**, 5(2) :1–24, apr 2009. doi: 10.4018/jswis.2009040101. URL https://doi.org/10.4018/jswis.2009040101.

[126] Michael Schmidt, Thomas Hornung, Georg Lausen, and Christoph Pinkel. SP^2bench : A SPARQL performance benchmark. In **2009 IEEE 25th International Conference on Data Engineering**. IEEE, mar 2009. doi: 10.1109/icde.2009.28. URL https://doi.org/10.1109/icde.2009.28.

[127] Mohamed Morsey, Jens Lehmann, Sören Auer, and Axel-Cyrille Ngonga Ngomo. DBpedia SPARQL benchmark – performance assessment with real queries on real data. In **The Semantic Web – ISWC 2011**, pages 454–469. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-25073-6_29. URL https://doi.org/10.1007/978-3-642-25073-6_29.

[128] Mohamed Morsey, Jens Lehmann, Sören Auer, and Axel-Cyrille Ngonga Ngomo. Usage-centric benchmarking of rdf triple stores. In **Twenty-Sixth AAAI Conference on Artificial Intelligence**, 2012.

[129] Christophe Gueret, Spyros Kotoulas, and Paul Groth. TripleCloud : An infrastructure for exploratory querying over web-scale RDF data. In **2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology**. IEEE, aug 2011. doi: 10.1109/wi-iat.2011.166. URL https://doi.org/10.1109/wi-iat.2011.166.

[130] Jacopo Urbani, Frank Van Harmelen, Stefan Schlobach, and Henri Bal. Querypie : Backward reasoning for owl horst over very large knowledge bases. In **International Semantic Web Conference**, pages 730–745. Springer, 2011.

[131] Günter Ladwig and Andreas Harth. Cumulusrdf : linked data management on nested key-value stores. In **The 7th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2011)**, volume 30, 2011.

[132] Jianling Sun and Qiang Jin. Scalable RDF store based on HBase and MapReduce. In **2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)**. IEEE, aug 2010. doi: 10.1109/icacte.2010.5578937. URL https://doi.org/10.1109/icacte.2010.5578937.

[133] Nikolaos Papailiou, Ioannis Konstantinou, Dimitrios Tsoumakos, and Nectarios Koziris. H2rdf : adaptive query processing on rdf data in the cloud. In **Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion**. ACM Press, 2012. doi: 10.1145/2187980.2188058. URL https://doi.org/10.1145/2187980.2188058.

[134] Philippe Cudré-Mauroux, Iliya Enchev, Sever Fundatureanu, Paul Groth, Albert Haque, Andreas Harth, Felix Leif Keppmann, Daniel Miranker, Juan F. Sequeda, and Marcin Wylot. NoSQL databases for RDF : An empirical evaluation. In **Advanced Information Systems Engineering**, pages 310–325. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-41338-4_20. URL https://doi.org/10.1007/978-3-642-41338-4_20.

[135] Steve Harris, Nick Lamb, Nigel Shadbolt, et al. 4store : The design and implementation of a clustered rdf store. In **5th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2009)**, pages 94–109, 2009.

[136] Rick Cattell. Scalable SQL and NoSQL data stores. **ACM SIGMOD Record**, 39(4) : 12, may 2011. doi: 10.1145/1978915.1978919. URL https://doi.org/10.1145/1978915.1978919.

[137] Pramod J Sadalage and Martin Fowler. **NoSQL distilled : a brief guide to the emerging world of polyglot persistence**. Pearson Education, 2012.

[138] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable : A distributed storage system for structured data. **ACM Transactions on Computer Systems**, 26 (2) :1–26, jun 2008. doi: 10.1145/1365815.1365816. URL https://doi.org/10.1145/1365815.1365816.

[139] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo : amazon's highly available key-value store. In **Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles - SOSP '07**. ACM Press, 2007. doi: 10.1145/1294261.1294281. URL https://doi.org/10.1145/1294261.1294281.

[140] Carlo Strozzi. Nosql-a relational database management system. **Lainattu**, 5 :2014, 1998.

[141] M Aslett. Nosql, newsql and beyond (2011).

[142] Nadime Francis, Andrés Taylor, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, and Petra Selmer. Cypher. In **Proceedings of the 2018 International Conference on Management of Data - SIGMOD '18**. ACM Press, 2018. doi: 10.1145/3183713.3190657. URL https://doi.org/10.1145/3183713.3190657.

[143] Marko A. Rodriguez. The gremlin graph traversal machine and language (invited talk). In **Proceedings of the 15th Symposium on Database Programming Languages - DBPL 2015**. ACM Press, 2015. doi: 10.1145/2815072.2815073. URL https://doi.org/10.1145/2815072.2815073.

[144] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. **Acm Sigact News**, 33(2) :51–59, 2002.

[145] Ahmed-Diouf Dirieh Dibad, Lina F. Soualmia, Tayeb Merabti, Julien Grosjean, Saoussen Sakji, Philippe Massari, and Stéfan J. Darmoni. Un modèle de données adapté à la recherche d'information dans le dossier patient informatisé : étude, conception et évaluation. In **Systèmes d'information pour l'amélioration de la qualité en santé. Comptes rendus des quatorzièmes Journées francophones d'informatique médicale (JFIM).**, Informatique et Santé, pages 251–262, Tunis, 09 2012. Springer. doi: 10.1007/978-2-8178-0285-5_22. URL http://www.springerlink.com/content/rx6402278w6m7863/.

[146] Lina F. Soualmia. **Gestion des Connaissances pour l'Accès aux Informations en Santé**. Habilitation à Diriger des Recherches, Université de Rouen Normandie, 2015.

[147] Sebastian Rudolph. Foundations of description logics. In **Reasoning Web International Summer School**, pages 76–136. Springer, 2011.

[148] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. **An Introduction to Description Logic**. Cambridge University Press, 2017. doi: 10.1017/9781139025355. URL https://doi.org/10.1017/9781139025355.

[149] Red Hat. Jboss. jboss infinispan, 2011.

[150] Francesco Marchioni and Manik Surtani. **Infinispan data grid platform**. Packt Publishing Ltd, 2012.

[151] K Manoj Kumar, S Tejasree, and S Swarnalatha. Effective implementation of data segregation & extraction using big data in e-health insurance as a service. In **Advanced Computing and Communication Systems (ICACCS), 2016 3rd International Conference on**, volume 1, pages 1–5. IEEE, 2016.

[152] Pradeeban Kathiravelu and Ashish Sharma. Mediator : A data sharing synchronization platform for heterogeneous medical image archives. In **Workshop on Connected Health at Big Data Era (BigCHat'15), co-located with 21 st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2015). ACM**, 2015.

[153] Haytham Salhi, Feras Odeh, Rabee Nasser, and Adel Taweel. Open source in-memory data grid systems : Benchmarking hazelcast and infinispan. In **Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering**, pages 163–164. ACM, 2017.

[154] Michael McCandless, Erik Hatcher, and Otis Gospodnetic. **Lucene in Action**. Manning Publications, second edition : covers apache lucene 3.0 edition, 8 2010. ISBN 9781933988177. URL http://amazon.de/o/ASIN/1933988177/.

[155] V. Kodaganallur. Incorporating language processing into java applications : a JavaCC tutorial. **IEEE Software**, 21(4) :70–77, jul 2004. doi: 10.1109/ms.2004.16. URL https://doi.org/10.1109/ms.2004.16.

[156] Tom Copeland. **Generating parsers with JavaCC**. Centennial Books, 2007. ISBN 0976221438. URL https://generatingparserswithjavacc.com/.

[157] Lina F. Soualmia, Romain Lelong, Badisse Dahamna, and Stéfan J. Darmoni. Rewriting natural language queries using patterns. In **Lecture Notes in Computer Science**, volume 9059, pages 40–53. Springer International Publishing, 2015. doi: 10.1007/978-3-319-24471-6_4. URL https://doi.org/10.1007/978-3-319-24471-6_4.

[158] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, and Hongfang Liu. Clinical information extraction applications : A literature review. **Journal of Biomedical Informatics**, 77 :34–49, jan 2018. doi: 10.1016/j.jbi.2017.11.011. URL https://doi.org/10.1016/j.jbi.2017.11.011.

[159] Kory Kreimeyer, Matthew Foster, Abhishek Pandey, Nina Arya, Gwendolyn Halford, Sandra F. Jones, Richard Forshee, Mark Walderhaug, and Taxiarchis Botsis. Natural language processing systems for capturing and standardizing unstructured clinical information : A systematic review. **Journal of Biomedical Informatics**, 73 :14–29, sep 2017. doi: 10.1016/j.jbi.2017.07.012. URL https://doi.org/10.1016/j.jbi.2017.07.012.

[160] Son Doan, Mike Conway, Tu Minh Phuong, and Lucila Ohno-Machado. Natural language processing in biomedicine : A unified system architecture overview. In **Methods in Molecular Biology**, pages 275–294. Springer New York, 2014. doi: 10.1007/978-1-4939-0847-9_16. URL https://doi.org/10.1007/978-1-4939-0847-9_16.

[161] Aurélie Névéol, Hercules Dalianis, Sumithra Velupillai, Guergana Savova, and Pierre Zweigenbaum. Clinical natural language processing in languages other than english : opportunities and challenges. **Journal of Biomedical Semantics**, 9(1), mar 2018. doi: 10.1186/s13326-018-0179-8. URL https://doi.org/10.1186/s13326-018-0179-8.

[162] Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C. Kipper-Schuler, and Christopher G. Chute. Mayo clinical text analysis and knowledge extraction system (cTAKES) : architecture, component evaluation and applications. **Journal of the American Medical Informatics Association**, 17(5) :507–513, sep 2010. doi: 10.1136/jamia.2009.001560. URL https://doi.org/10.1136/jamia.2009.001560.

[163] N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, Jonquet C., D. L. Rubin, M.-A. Storey, C. G. Chute, and M. A. Musen. BioPortal : ontologies and integrated data resources at the click of a mouse. **Nucleic Acids Research**, 37(Web Server) : W170–W173, may 2009. doi: 10.1093/nar/gkp440. URL https://doi.org/10.1093/nar/gkp440.

[164] Alan R. Aronson and François-Michel Lang. An overview of MetaMap : historical perspective and recent advances. **Journal of the American Medical Informatics Association**, 17(3) :229–236, may 2010. doi: 10.1136/jamia.2009.002733. URL https://doi.org/10.1136/jamia.2009.002733.

[165] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In **Proceedings of the 32nd annual meeting on Association for Computational Linguistics**, pages 133–138. Association for Computational Linguistics, 1994.

[166] Haïfa Zargayouna. **Indexation sémantique de documents XML**. PhD thesis, Paris 11, 2005.

[167] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet : An experimental, application-oriented evaluation of five measures. In **Workshop on WordNet and other lexical resources**, volume 2, pages 2–2, 2001.

[168] Ignacio Traverso, Maria-Esther Vidal, Benedikt Kämpgen, and York Sure-Vetter. Gades : a graph-based semantic similarity measure. In **Proceedings of the 12th International Conference on Semantic Systems**, pages 101–104. ACM, 2016.

[169] Ganggao Zhu and Carlos A. Iglesias. Computing semantic similarity of concepts in knowledge graphs. **IEEE Transactions on Knowledge and Data Engineering**, 29(1) : 72–85, January 2017. doi: 10.1109/tkde.2016.2610428. URL https://doi.org/10.1109/tkde.2016.2610428.

[170] Faezeh Ensan and Weichang Du. Ad hoc retrieval via entity linking and semantic similarity. **Knowledge and Information Systems**, 58(3) :551–583, April 2018. doi: 10.1007/s10115-018-1190-1. URL https://doi.org/10.1007/s10115-018-1190-1.

[171] Romain Lelong, Lina F. Soualmia, Badisse Dahamna, Nicolas Griffon, and Stéfan J. Darmoni. Querying ehrs with a semantic and entity-oriented query language. **Studies in Health Technology and Informatics**, 235(Informatics for Health : Connected Citizen-Led Wellness and Population Health) :121–125, 2017. ISSN 0926-9630. doi: 10.3233/978-1-61499-753-5-121. URL http://doi.org/10.3233/978-1-61499-753-5-121.

[172] Romain Lelong, Chloé Cabot, Lina F. Soualmia, and Stéfan J. Darmoni. Semantic search engine to query into electronic health records with a multiple-layer query language. In **Proceedings of the 2nd Special Interest Group on Information Retrieval (SIGIR) workshop on Medical Information Retrieval (MedIR)**, Pisa, Italy, 07 2016. URL http://medir2016.imag.fr/data/MEDIR_2016_paper_8.pdf.

[173] Chloé Cabot, Romain Lelong, Julien Grosjean, Lina F. Soualmia, and Stéfan J. Darmoni. Retrieving clinical and omic data from electronic health records. **Studies in Health Technology and Informatics**, 221(Transforming Healthcare with the Internet of Things) :115–115, 2016. ISSN 0926-9630. doi: 10.3233/978-1-61499-633-0-115. URL http://doi.org/10.3233/978-1-61499-633-0-115.

[174] Romain Lelong, Lina F. Soualmia, Badisse Dahamna, Julien Grosjean, and Stéfan J. Darmoni. Rewriting natural language queries using patterns in an electronic health records system. In **European Conference on Information Retrieval ; Workshop on Multimodal Retrieval in the Medical Domain**, Vienna, Austria, 03 2015.

[175] Tayeb Merabti, Romain Lelong, and Stéfan J. Darmoni. Inforoute : the cismef context-specific search algorithm. **Studies in Health Technology and Informatics**, 216 (MEDical INFOrmatics conference (MedInfo) 2015 : eHealth-enabled Health) :544–548, 2015. ISSN 0926-9630. doi: 10.3233/978-1-61499-564-7-544. URL http://doi.org/10.3233/978-1-61499-564-7-544.

[176] Chloé Cabot, Lina F. Soualmia, Julien Grosjean, Romain Lelong, and Stéfan J. Darmoni. Integrating and retrieving clinical and omic data in electronic health records. In **7th International Workshop on Knowledge Representation for Health Care (KRH4C) and 8th International Workshop on Process-oriented Information Systems in Healthcare (ProHealth)**, pages 154–159, 2015. URL https://www.researchgate.net/profile/Lina_Soualmia/publication/280066101_Integrating_and_Retrieving_Clinical_and_Omic_Data_in_Electronic_Health_Records/links/55a7a47408aeceb8cad65695.pdf.

[177] Chloé Cabot, Julien Grosjean, Romain Lelong, Arnaud Lefebvre, Thierry Lecroq, Lina F. Soualmia, and Stéfan J. Darmoni. Omic data modelling for information retrieval. In **Proceedings of the 2nd International Work-Conference on Bioinformatics and Biomedical Engineering, International Work-conference on Bioinformatics and BIOmedical engineering (IWBBIO)**, 2014. URL http://iwbbio.ugr.es/2014/papers/IWBBIO_2014_paper_50.pdf.

[178] Emeric Dynomant, Romain Lelong, Badisse Dahamna, Clément Massonnaud, Gaëtan Kerdelhué, Julien Grosjean, Stéphane Canuc, and Stéfan J. Darmoni. Word embedding for french natural language in healthcare : a comparison study. (MEDical INFOrmatics conference (MedInfo) 2019), 2019.

[179] Romain Lelong, Lina Soualmia, Saoussen Sakji, Badisse Dahamna, and Stéfan J. Darmoni. Une technologie nosql au service de moteur de recherche en santé. In **4ème édition du Symposium sur l'Ingénierie de l'Information Médicale**, Toulouse, France, 11 2017.

[180] Romain Lelong, Chloé Cabot, Tayeb Merabti, Julien Grosjean, Nicolas Griffon, Badisse Dahamna, Philippe Massari, and Stéfan J. Darmoni. Information retrieval in electronic health records using a multiple layer query language. In **Journées Recherche en Imagerie et Technologies pour la Santé (RITS) 2015**, pages 128–129, 2015. projet RAVEL ANR TecSan.

[181] Romain Lelong, Tayeb Merabti, Julien Grosjean, Mher Joulakian, Nicolas Griffon, Badisse Dahamna, Marc Cuggia, Suzanne Pereira, Natalia Grabar, Franck Thiessard, Philippe Massari, and Stéfan J. Darmoni. Moteur de recherche sémantique au sein du dossier du patient informatisé : langage de requêtes spécifique. In **Journées Francophones d'Informatique Médicale (JFIM)**, pages 139–151, Fès, Maroc, 06 2014.

[182] Romain Lelong, Lina F. Soualmia, Saoussen Sakji, Badisse Dahamna, and Stéfan J. Darmoni. Nosql technology in order to support semantic health search engine. In **Medical Informatics Europe (MIE) 2018**, Gothenburg, Sweden, 04 2018.

[183] Chloé Cabot, Julien Grosjean, Romain Lelong, Arnaud Lefebvre, Thierry Lecroq, Lina F. Soualmia, and Stéfan J. Darmoni. Integrating omic and clinical data in electronic health records for visualisation and retrieval. In **3ème Journée Scientifique de l'Institute for Research and Innovation in Biomedicine (IRIB)**, page 64, Caugé, France, 2014.

[184] Arnaud Lefebvre, Alexandra Martins, Karim Labrèche, Vivien Deshaies, Alan Lahure, Pascaline Gaildrat, Hélène Dauchel, Chloé Cabot, Julien Grosjean, Romain Lelong, A. Lefebvre, Thierry Lecroq, Lina F. Soualmia, and Stéfan J. Darmoni. Hexosplice : a bioinformatics software based on overlapping hexamer scores for prediction and stratification of exonic variants altering splicing regulation of human genes. In **3ème Journée Scientifique de l'Institute for Research and Innovation in Biomedicine (IRIB)**, Caugé, France, 2014.

Listes des publications

Communications avec actes dans un congrès international

— Marie Ndangang, Julien Grosjean, Romain Lelong, Badisse Dahamna, Ivan Kergourlay, Nicolas Griffon, and Stéfan J. Darmoni. Terminology coverage from semantic annotated health documents. **Studies in Health Technology and Informatics**, 255(Decision Support Systems and Education):20–24, 2018. ISSN 0926-9630. doi: 10.3233/978-1-61499-921-8-20. URL `http://doi.org/10.3233/978-1-61499-921-8-20`

— Romain Lelong, Lina F. Soualmia, Badisse Dahamna, Nicolas Griffon, and Stéfan J. Darmoni. Querying ehrs with a semantic and entity-oriented query language. **Studies in Health Technology and Informatics**, 235(Informatics for Health: Connected Citizen-Led Wellness and Population Health):121–125, 2017. ISSN 0926-9630. doi: 10.3233/978-1-61499-753-5-121. URL `http://doi.org/10.3233/978-1-61499-753-5-121`

— Romain Lelong, Chloé Cabot, Lina F. Soualmia, and Stéfan J. Darmoni. Semantic search engine to query into electronic health records with a multiple-layer query language. In **Proceedings of the 2nd Special Interest Group on Information Retrieval (SIGIR) workshop on Medical Information Retrieval (MedIR)**, Pisa, Italy, 07 2016. URL `http://medir2016.imag.fr/data/MEDIR_2016_paper_8.pdf`

— Chloé Cabot, Romain Lelong, Julien Grosjean, Lina F. Soualmia, and Stéfan J. Darmoni. Retrieving clinical and omic data from electronic health records. **Studies in Health Technology and Informatics**, 221(Transforming Healthcare with the Internet of Things):115–115, 2016. ISSN 0926-9630. doi: 10.3233/978-1-61499-633-0-115. URL `http://doi.org/10.3233/978-1-61499-633-0-115`

— Lina F. Soualmia, Romain Lelong, Badisse Dahamna, and Stéfan J. Darmoni. Rewriting natural language queries using patterns. In **Lecture Notes in Computer Science**, volume 9059, pages 40–53. Springer International Publishing, 2015. doi: 10.1007/978-3-319-24471-6_4. URL `https://doi.org/10.1007/978-3-319-24471-6_4`

— Romain Lelong, Lina F. Soualmia, Badisse Dahamna, Julien Grosjean, and Stéfan J. Darmoni. Rewriting natural language queries using patterns in an electronic health records system. In **European Conference on Information Retrieval; Workshop on Multimodal Retrieval in the Medical Domain**, Vienna, Austria, 03 2015

— Tayeb Merabti, Romain Lelong, and Stéfan J. Darmoni. Inforoute: the cismef context-specific search algorithm. **Studies in Health Technology and Informatics**, 216 (MEDical INFOrmatics conference (MedInfo) 2015: eHealth-enabled Health):544–548, 2015. ISSN 0926-9630. doi: 10.3233/978-1-61499-564-7-544. URL `http://doi.org/10.3233/978-1-61499-564-7-544`

— Chloé Cabot, Lina F. Soualmia, Julien Grosjean, Romain Lelong, and Stéfan J. Darmoni. Integrating and retrieving clinical and omic data in electronic health records. In **7th International Workshop on Knowledge Representation for Health Care (KRH4C) and 8th International Workshop on Process-oriented Information Systems in Healthcare (ProHealth)**, pages 154–159, 2015. URL `https://www.researchgate.net/profile/Lina_Soualmia/publication/280066101_Integrating_and_Retrieving_Clinical_and_Omic_Data_in_Electronic_Health_Records/links/55a7a47408aeceb8cad65695.pdf`

— Chloé Cabot, Julien Grosjean, Romain Lelong, Arnaud Lefebvre, Thierry Lecroq, Lina F. Soualmia, and Stéfan J. Darmoni. Omic data modelling for information retrieval. In **Proceedings of the 2nd International Work-Conference on Bioinformatics and Biomedical Engineering, International Work-conference on Bioinformatics and BIOmedical engineering (IWBBIO)**, 2014. URL `http://iwbbio.ugr.es/2014/papers/IWBBIO_2014_paper_50.pdf`

Communications avec actes dans un congrès national

— Emeric Dynomant, Romain Lelong, Badisse Dahamna, Clément Massonnaud, Gaëtan Kerdelhué, Julien Grosjean, Stéphane Canuc, and Stéfan J. Darmoni. Word embedding for french natural language in healthcare: a comparison study. (MEDical INFOrmatics conference (MedInfo) 2019), 2019

— Romain Lelong, Lina Soualmia, Saoussen Sakji, Badisse Dahamna, and Stéfan J. Darmoni. Une technologie nosql au service de moteur de recherche en santé. In **4ème édition du Symposium sur l'Ingénierie de l'Information Médicale**, Toulouse, France, 11 2017

— Romain Lelong, Chloé Cabot, Tayeb Merabti, Julien Grosjean, Nicolas Griffon, Badisse Dahamna, Philippe Massari, and Stéfan J. Darmoni. Information retrieval in electronic health records using a multiple layer query language. In **Journées Recherche en Imagerie et Technologies pour la Santé (RITS) 2015**, pages 128–129, 2015. projet RAVEL ANR TecSan

— Romain Lelong, Tayeb Merabti, Julien Grosjean, Mher Joulakian, Nicolas Griffon, Badisse Dahamna, Marc Cuggia, Suzanne Pereira, Natalia Grabar, Franck Thiessard, Philippe Massari, and Stéfan J. Darmoni. Moteur de recherche sémantique au sein du dossier du patient informatisé : langage de requêtes spécifique. In **Journées Francophones d'Informatique Médicale (JFIM)**, pages 139–151, Fès, Maroc, 06 2014

Posters

— Romain Lelong, Lina F. Soualmia, Saoussen Sakji, Badisse Dahamna, and Stéfan J. Darmoni. Nosql technology in order to support semantic health search engine. In **Medical Informatics Europe (MIE) 2018**, Gothenburg, Sweden, 04 2018

— Chloé Cabot, Julien Grosjean, Romain Lelong, Arnaud Lefebvre, Thierry Lecroq, Lina F. Soualmia, and Stéfan J. Darmoni. Integrating omic and clinical data in electronic health records for visualisation and retrieval. In **3ème Journée Scientifique de l'Institute for Research and Innovation in Biomedicine (IRIB)**, page 64, Caugé, France, 2014

— Arnaud Lefebvre, Alexandra Martins, Karim Labrèche, Vivien Deshaies, Alan Lahure, Pascaline Gaildrat, Hélène Dauchel, Chloé Cabot, Julien Grosjean, Romain Lelong, A. Lefebvre, Thierry Lecroq, Lina F. Soualmia, and Stéfan J. Darmoni. Hexosplice: a bioinformatics software based on overlapping hexamer scores for prediction and stratification of exonic variants altering splicing regulation of human genes. In **3ème Journée Scientifique de l'Institute for Research and Innovation in Biomedicine (IRIB)**, Caugé, France, 2014

Annexe A

Le thésaurus Le thésaurus Medical Subject Headings (MeSH)

Les **concept terminologique** du MeSH sont de **trois types** :

Les « Descriptors » (Descripteurs) : Ces derniers constituent l'unité d'indexation de base du thésaurus MeSH. Ils permettent généralement d'indiquer le sujet des article scientifique qu'ils indexent (e.g. « *asthme* » [*D*001249 (MeSH)] ou encore « *VIH (Virus de l'Immunodéficience humaine)* » [*D*006678 (MeSH)]). Ils peuvent néanmoins également correspondre à des type de publication (e.g. « *article historique* » [*D*016456 (MeSH)], « *essai clinique* » [*D*016430 (MeSH)] etc.), des indications géographiques (e.g. « *France* » [*D*005602 (MeSH)], « *Paris* » [*D*010297 (MeSH)], etc.) ou enfin à des « Check-Tags » (e.g. « *Mâle* » [*D*008297 (MeSH)], « *Femelle* » [*D*005260 (MeSH)], etc.).

Les « Qualifiers » (Qualificatifs) : Ils sont au nombre de 81. Ils sont utilisé en conjonction avec un descripteur afin de préciser un aspect particulier du sujet spécifié par ce dernier. Par exemple, une indexation du type « *insuffisance hépatique* » [*D*048550 (MeSH)]/« *traitement médicamenteux* » [*Q*000188 (MeSH)] permet d'indiquer qu'un article scientifique traite spécifiquement du traitement médicamenteux de insuffisance hépatique.

Les « Supplementary Chemical Concept » (Concepts Chimiques supplémentaires) : Il permettent de représenter des produits chimiques (e.g. « *2-methyl-4-chlorophenoxy gamma-butyric acid* » [*C*008418 (MeSH)]), des protocoles de chimiothérapie (e.g. « *VAD regimen* » [*C*053329 (MeSH)]), des maladies rares (e.g. « *syndrome d'hyperlaxité articulaire marfanoïde* » [*C*531742 (MeSH)]) ou encore des organismes tels que des virus (e.g. « *H1N1 virus hemagglutinin* » [*C*543345 (MeSH)]).

Le MeSH organise sémantiquement ces types de concepts terminologique par l'intermédiaire, d'une part, d'une **hiérarchie de descripteur** et, d'autre part, d'une **hiérarchie de qualificatifs**. Les concepts chimiques supplémentaires ne sont, eux, **pas hiérarchisés** mais simplement rattachés (ou reliés) à un ou plusieurs descripteurs.

De plus, un niveau de structuration supplémentaire est ajouté à chacun des ces trois type de concepts terminologiques. Chaque descripteur, qualificatif ou concept chimique supplémentaire correspond en réalité à une liste de **concept MeSH** dont l'un est considéré comme le **concept MeSH préféré** du descripteur. Chacun des ces concepts sont à leurs tours rattachés à une liste de **termes MeSH** dont l'un, au moins, est considéré comme le **terme MeSH préféré** du concept MeSH.

Un exemple de cette structuration est donnée dans l'exemple 34 p. 244. Dans la pratique, les termes MeSH correspondent à des synonymes stricte du concept MeSH auquel ils appartiennent. En revanche, les concepts MeSH peuvent également correspondre à une notion plus spécifique ou plus générale que celle exprimée par le descripteur auquel ils sont rattachés.

Exemple 34 (Structuration des concepts terminologiques du MeSH) :
Dans la terminologie MeSH, le concept terminologique (et descripteur) MeSH « AIDS Dementia Complex » [D015526 (MeSH)] est structuré de la manière suivante :

« AIDS Dementia Complex » [D015526 (MeSH)]	*Descripteur*
« AIDS Dementia Complex » [M0023886 (MeSH)]	*Concept MeSH préféré*
« AIDS Dementia Complex »	*Terme MeSH préféré*
« Acquired-Immune Deficiency Syndrome Dementia Complex »	*Terme MeSH*
« AIDS-Related Dementia Complex »	*Terme MeSH*
« HIV Dementia »	*Terme MeSH*
« Dementia Complex, Acquired Immune Deficiency Syndrome »	*Terme MeSH*
« Dementia Complex, AIDS-Related »	*Terme MeSH*
« HIV Encephalopathy » [M0023888 (MeSH)]	*Concept MeSH spécifiant*
« HIV Encephalopathy »	*Terme MeSH préféré*
« AIDS Encephalopathy »	*Terme MeSH*
« Encephalopathy, HIV »	*Terme MeSH préféré*
« Encephalopathy, AIDS »	*Terme MeSH*
« HIV-1-Associated Cognitive Motor Complex » [M0023889 (MeSH)]	*Concept MeSH spécifiant*
« HIV-1-Associated Cognitive Motor Complex »	*Terme MeSH préféré*
« HIV-1 Cognitive and Motor Complex »	*Terme MeSH*

Cet exemple est reprit de la documentation en ligne du MeSH[a].

a. url : `https://www.nlm.nih.gov/mesh/concept_structure.html`

Annexe B

Étude cliniques

B.1 Étude clinique n° 1

Critères d'inclusion :

1. âge 18 ans, < 70 ans
2. patient ayant été informé et ayant donné son consentement
3. SUB pelade décalvante "totale, universelle touchant la totalité ou la quasi totalité de la surface du cuir chevelu" (et éventuellement les poils corporels, incluant les cils, les sourcils)
4. SUB pelade d'évolution chronique définie comme étant au stade de pelade décalvante évoluant sans repousse depuis au moins 6 mois et moins de 5 ans, malgré un ou plusieurs traitements préalables habituels (dont photothérapie (PUVA ou UVB), applications d'un dermocorticoïde puissant (type propionate de clobetasol : crème ou gel dermoval), applications de minoxidil 5%, ou bolus de corticoïde IV, à l'exclusion du méthotrexate (médicament testé dans l'essai). NB : les repousses minimes, esthétiquement incorrectes ou les repousses d'un simple duvet sont incluables
5. SUB altération importante de la qualité de vie, définie par un score 10 au questionnaire DLQI
6. pour les femmes en âge de procréer, une contraception efficace (stérilet, contraception oestroprogestative. . . .) sera exigée pendant le traitement et pendant l'année suivant l'arrêt de celui-ci
7. pour les hommes participant à l'étude, une contraception est nécessaire pendant la durée de l'essai et pendant 5 mois apres l'arrêt du traitement.
8. période de wash out de 2 mois entre la fin du dernier traitement systémique essayé et l'inclusion dans l'étude
9. statut vaccinal à jour.

Critères d'exclusion :

1. femme enceinte ou allaitante
2. hypersensibilité connue à l'un des produits (méthotrexate, corticoïdes)
3. patient séropositif pour le VIH
4. patient atteint d'hépatite B ou d'hépatite C active ou non (les patients ayant des anticorps contre le virus B du fait d'une vaccination sont cependant incluables)
5. patient ayant reçu un traitement immunosuppresseur (type ciclosporine, mycophénolate mofetil, cyclophosphamide, azathioprine, méthotrexate) ou tout autre traitement systémique pouvant potentiellement être actif sur la pelade pendant les 2 mois précédant l'inclusion dans l'essai.
6. troubles du rythme cardiaque mal équilibrés
7. insuffisance cardiaque sévère (classe III ou IV, NYHA) voir annexe n°05

8. angor instable ou cardiopathie ischémique évoluée (infarctus étendu récent inférieur à trois mois ou insuffisance cardiaque post infarctus)
9. hépatopathie connue active (en dehors d'une simple stéatose hépatique) transaminases et/ou phosphatases alcalines supérieures à deux fois la norme supérieure du laboratoire) ou antécédent d'hépatopathie récente (moins de 2 ans) quelle que soit sa nature pouvant faire craindre une mauvaise tolérance du méthotrexate
10. Prise de médicaments notoirement reconnus comme hépatotoxiques ou interférant avec le métabolisme ou la toxicité hématologique du méthotrexate
11. consommation régulière d'alcool supérieure à 60 g d'alcool par jour,(soit environ 0.5 litre de vin ou équivalent par jour)
12. insuffisance rénale significative définie par une clairance de la créatinine inférieure à 50 ml/mn selon la formule de Cockcroft
13. diabète déséquilibré (glycémie à jeun = 2,5g/L et/ou HbA1C = 8.5% avant traitement)

B.2 Étude clinique n° 2

Critères d'inclusion :

1. Patient hospitalisé en réanimation pour syndrome d'activation macrophagique secondaire à un sepsis/choc septique bactérien ayant un score de probabilité clinique fort (défini par un HScore> 80% cf annexe 4) (population expérimentale) OU Patient hospitalisé en réanimation pour sepsis ou choc septique (population contrôle)
2. Age = 18 ans
3. Personne affiliée à un régime de sécurité sociale
4. Personne de confiance ou patient informée et ayant signé le consentement de participation à la recherche. (Si le patient est en impossibilité de signer son consentement (situations d'urgence) le consentement sera signé par la personne de confiance, et un consentement de poursuite de l'étude sera demandé par la suite au patient)).
5. NB Contraception efficace chez les femmes en âge de procréer (test de grossesse négatif). Pour les femmes ménopausées, un diagnostic de confirmation devra être obtenu (aménorrhée depuis au moins 12 mois avant la visite d'inclusion).

Critères d'exclusion :

1. Femme enceinte ou allaitante
2. Personne privée de liberté par une décision administrative ou judiciaire
3. Sujet majeur protégé, sous tutelle ou curatelle
4. Patient participant à un autre essai clinique interventionnel ayant le même objectif principal

B.3 Étude clinique n° 3

Critères d'inclusion :

1. Patient âgé de plus de 65 ans hospitalisé pour une hémorragie digestive haute ou basse d'évolution favorable en cours d'hospitalisation sans recours à la chirurgie
2. Patient présentant une anémie persistante au décours de la prise en charge, définie par un taux d'hémoglobine =11g/dL.
3. Poids supérieur ou égal à 50 kg
4. Anémie bien tolérée selon les critères cliniques habituels
5. Signature d'un consentement éclairé
6. Affiliation à un régime de sécurité sociale

Critères d'exclusion :

1. Les hémorragies non contrôlées définies par toute nouvelle extériorisation et/ou une diminution des valeurs biologiques d'hémoglobine et d'hématocrite
2. Les hémorragies rattachées à une hypertension portale ou une néoplasie
3. Taux d'hémoglobine = 9g/dL ou > 11g/dL
4. Poids inférieur à 50 kg
5. Cancer évolutif
6. Patient sous tutelle, curatelle ou hors d'état d'exprimer son consentement
7. Signe de surcharge martiale ou troubles de l'utilisation du fer
8. Antécédents d'asthme, d'eczéma ou d'allergies atopiques.
9. Hypersensibilité à la substance active (Ferinject®) ou à l'un des excipients (hydroxyde de Sodium, Acide Chlorhydrique)
10. Hypersensibilité grave connue à tout autre fer administré par voie parentérale
11. Cirrhose hépatique décompensée
12. Infection en cours de traitement < 48h ou infections non contrôlés
13. Troubles immunitaires ou inflammatoires (par exemple polyarthrite rhumatoïde ou lupus érythémateux systémique)
14. Insuffisance rénale aiguë
15. Anémie microcytaire
16. Porphyrie cutanée tardive

B.4 Étude clinique n° 4

Critères d'inclusion :

1. sujet âgé de 18 à 70 ans, homme ou femme ménopausée
2. sujet diabétique de type 2,
3. sujet ayant un index pondéral (poids/taille2) supérieur à 27,
4. ayant une hypertension artérielle, dont le contrôle par inhibiteur de l'enzyme de conversion ou antagoniste des récepteurs de l'angiotensine 2, éventuellement associé à un inhibiteur calcique, un a-bloquant ou un anti-hypertenseur central, est insuffisant (pression artérielle > 140/85 mm Hg TAS>140 et/ou TAD>85 mm Hg),
5. avec une hémoglobine glyquée entre 6,5% et 8% maximum
6. ayant des sérologies HIV et hépatites B et C négatives,
7. n'ayant pas participé à un essai thérapeutique au cours des trois mois précédent l'étude et ne participant pas à un autre essai pendant toute la durée de l'étude.
8. inscrit ou ayant droit au régime général de la Sécurité Sociale,
9. ayant signé un consentement éclairé.

Critères d'exclusion :

1. sujets de sexe féminin, en âge de procréer,
2. sujet mineur et âgé de plus de 70 ans,
3. sujet diabétique de type 1,
4. sujet non diabétique et sujet diabétique de type 2 normo tendu (pression artérielle < 140/85 mm Hg), TAS<140 et/ou TAD<85 mm Hg),
5. sujet avec une hémoglobine glyquée inférieure à 6,5% ou supérieure à 8%,
6. patients présentant une leuconeutropénie (taux de polynucléaires neutrophiles inférieur ou 1700/mm3)

7. patients ayant des antécédents médicaux ou chirurgicaux sévères en particulier endocriniens,
8. patients traités par des médicaments métabolisés par les cytochromes CYP3A4 et CYP2C9 : corticoïdes, anti vitamine K, contraceptifs hormonaux, tolbutamide, benzodiazépines, dérivés de l'ergot de seigle, antiépileptiques, millepertuis, macrolides, antifongiques azolés, pimozide, terfénadine, astémizole, cisapride, rifampicine, inhibiteurs de protéase, ciclosporine, tacrolimus, sirolimus, évérolimus, alfentanil, fentanyl, quinidine, irinotécan, et médicaments chimiothérapeutiques (etoposide, vinorelbine, ifosfamide)
9. patients traités par des médicaments interférant avec le système rénine-angiotensine-aldostérone : bêta-bloquants, diurétiques, médicaments anti-aldostérone, inhibiteurs directs de la rénine, modamide, insuline,
10. patients diabétiques de type 2 avec une neuropathie végétative autonome,
11. patients consommant régulièrement de la réglisse ou ses dérivés
12. patients chez qui une masse surrénalienne a été diagnostiquée à l'imagerie,
13. insuffisance hépatique ou rénale (définie respectivement par des manifestations cliniques et biologiques secondaires à l'altération des fonctions hépatocytaires (transaminases >3N) ou une estimation du débit de filtration glomérulaire inférieure à 60

B.5 Étude clinique n° 5

Critères d'inclusion :

1. Patient = 18 ans et <75 ans
2. des douleurs abdominales survenant au moins 1 jour par semaine durant les 3 derniers mois associé à au moins 2 des critères suivants : en relation avec la défécation, survenue associée à une modification de la fréquence ou de la consistance des selles (à type de diarrhée)
3. Patient présentant une calprotectine fécale =200 µg/g dans un délai de 2 mois
4. Affilié à un régime de sécurité sociale
5. Patient ayant lu et compris la lettre d'information et signé le formulaire de consentement
6. Pour les femmes en âge de procréer prise d'une contraception efficace (oestro-progestatifs ou dispositif intra-utérin ou ligature de trompes) depuis 1 mois (test de grossesse négatif).
7. Pour les femmes ménopausées, un diagnostic de confirmation devra être obtenu

Critères d'exclusion :

1. Patient présentant une pathologie organique digestive et/ou inflammatoire évolutive
2. Patient présentant un syndrome de l'intestin irritable à forme constipée ou alternant diarrhée/constipation selon les critères de RomeIV
3. Patient ayant pour traitement un anti-inflammatoire (type 5-ASA, budésonide) ou un probiotique, ou l'ayant arrêté depuis moins de 3 mois.
4. Patient présentant des troubles de la crase sanguine connus ou décelés par un interrogatoire ciblé, sous anticoagulant ou sous anti-agrégant plaquettaire
5. Patient présentant une pullulation microbienne (test respiratoire au glucose)
6. Patient présentant une hypersensibilité connue au Normacol ou à l'un de ses constituants
7. Patient ayant une insuffisance rénale sévère

8. Patient ayant une pathologie anale de type fissure anale ou thrombose hémorroïdaire
9. Femme enceinte ou allaitante ou absence de contraception avérée
10. Personne privée de liberté par une décision administrative ou judiciaire ou personne faisant l'objet d'une mesure juridique de protection des majeurs (sauvegarde de justice ou tutelle ou curatelle).
11. Patient participant à un autre essai / ayant participé à un autre essai dans un délai de 2 semaines
12. Patient ne parlant ou ne comprenant pas le Français
13. Régime en particulier à base de raisins ou d'extraits de raisins

Annexe C

Annotation Sémantique des Documents Médicaux

Les deux tableaux ci-dessous reprennent les résultats de l'étude [59] relative à l'annotation sémantique des textes cliniques (i.e. documents médicaux) du CHU de Rouen effectuée dans le cadre de la création de l'EDSS. Afin de comprendre la signification des abréviations utilisées pour désigner les TOs dans les tableaux suivants merci de se référer à l'Index des systèmes d'organisation des connaissances p. 269.

Terminologies	**Nombre d'annotations après filtrage**	**Nombre d'annotations avant filtrage**	**Facteur de filtrage**
SNOMED–CT®	394 133 994	881 884 314	2,2
NCIt	319 853 952	843 195 067	2,6
MeSH	295 537 298	1 024 585 229	3,5
SNOMED 3.5	219 706 745	440 228 408	2,0
TSP	179 747 539	454 354 922	2,5
MedDRA	137 653 806	225 100 880	1,6
PHA	106 616 463	171 231 559	1,6
RadLex®	80 197 479	150 406 338	1,9
FMA	55 350 010	123 777 281	2,2
CiSMeF	51 051 547	138 204 239	2,7
Drug list	33 355 617	37 554 068	1,1
ICNP	30 775 599	50 502 115	1,6
PASCAL	28 520 543	38 917 274	1,4
CIM–10	27 688 468	41 208 901	1,5
HPO	27 526 442	40 038 338	1,5
MedlinePlus	18 951 750	24 168 261	1,3
CIM-9	13 445 266	24 023 060	1,8
DRC	13 319 633	17 741 845	1,3
IUPAC	11 942 501	31 473 471	2,6
CLADIMED	9 867 474	27 025 227	2,7

TABLE C.1 – Nombre d'annotations unitaire des documents médicaux du CHU de Rouen par terminologies. Seules les 20 premières terminologies ayant le plus d'annotations unitaires sont présentées.

Terminologies	**Nombre total de concepts**	**Nombre de concepts traduits**		**Nombre de concepts uniques identifiés**	**ratio de couverture terminologique**
SNOMED–CT®	326 946	194 611	(59,5%)	59 330	30,5
SNOMED 3.5	100 908	100 908	(100%)	36 229	35,9
NCIt	93 925	68 776	(73,2%)	25 315	36,8
MedDRA (LLT)	44 226	44 226	(100%)	22 711	51,4
MeSH (D)	28 329	28 329	(100%)	18 288	64,6
MedDRA (TP)	21 612	21 612	(100%)	13 580	62,8
MeSH (CPT)	365 731	102 116	(27,9%)	12 625	12,4
FMA (ENT)	81 041	16 629	(20,5%)	8 084	48,6
RadLex®	42 313	10 259	(24,2%)	6 114	59,6
TSP	7 087	7 087	(100%)	6 089	85,9

LTT : Termes de plus bas niveau
D : Descripteurs
TP : Termes préférés
CPT : Concepts
ENT : Entités

TABLE C.2 – Couverture terminologique des annotations des documents médicaux du CHU de Rouen. Seules les dix premières terminologies ayant le plus de concepts uniques identifiés sont présentées.

Annexe D

Rappels sur les langages formels

Le langage de requêtes dont la syntaxe à été décrite dans la section section 6.1 p. 130 est dans la pratique basé sur une **grammaire formelle** dont la construction a été réalisée dans le cadre de cette thèse.

Une grammaire formelle est assimilable à un ensemble de règles permettant, par « application » récursive de ces dernières, de construire un ensemble de **mots** qui constitue un **langage formel**.

Cette annexe rappelle les fondements mathématiques permettant d'aboutir à la notion et l'objet mathématique de **grammaire formelle**.

Ces derniers sont indispensables à la bonne compréhension des règles de grammaire constituant le langage de requête exploité par le SSE_{SQL} et le SSE_{NoSQL}.

De plus, la grammaire formelle de ce langage est en soit, bien qu'un élément a priori purement formel, un composant informatique essentiel ayant servi de support à l'implémentation de ces deux moteurs de recherche.

Dans cette annexe, les notations suivantes seront employées :

Notation 8 : — *On notera pour tout entier naturel* $n \in \mathbb{N}$ *:*

$$I_n = \begin{cases} \varnothing & si\ n = 0 \\ [\![1\,;\ n]\!] & si\ n > 0 \end{cases}$$

Où $[\![1\,;\ n]\!]$ *désigne l'ensemble des entiers naturels compris entre* 1 *et* n *(viz.*$[\![1\,;\ n]\!] = \{i \in \mathbb{N} : 1 \leqslant i \leqslant n\}$*).*

- *Pout tout ensemble* E *et tout ensemble* F *on notera* $\mathcal{A}(E, F)$ *l'ensemble des applications de* E *dans* F*.*
- *Pour tout ensemble* E *et tout entier naturel* $n \in \mathbb{N}$ *on notera* $E_n = \mathcal{A}(I_n, E)$ *l'ensemble des applications de* I_n *dans* E*.*
- *Pour tout ensemble* E *on notera* $\mathcal{P}(E)$ *l'ensemble des parties de* E*.*
- *Pour tout ensemble fini* E*, on notera* $card(E)$ *le cardinal de l'ensemble* E*.*
- *Pour tout ensemble* E *et tout ensemble* F *on notera* $f : E \longrightarrow F$ *tout application* f *de* E *dans* F*.*
- *Pour toute fonction* f *d'un ensemble* E *dans un ensemble* F *on notera :*
  - $dom(f) = E$ *l'ensemble de départ (ou domaine) de l'application* f*.*
  - $codom(f) = F$ *l'ensemble d'arrivé (ou le codomaine) de l'application* f*.*

D.1 Alphabet et mots

Définition 34 (Alphabet) :
On appelle ***Alphabet*** *tout ensemble A fini et non vide.*
Les éléments de A sont alors appelés ***symboles*** *ou encore* ***lettres****.*

> ***Remarque 7 :***
> *Pour tous alphabets A_1 et A_2, $A_1 \cup A_2$ et $A_1 \cap A_2$ sont des alphabets.*

Un alphabet est donc un ensemble fini contenant les éléments syntaxiques de base à partir desquels les expressions d'un langage sont construites. Si l'on considère les langages informatiques, ces éléments syntaxiques correspondent notamment aux mots-clés du langage en question.

Bien que, formellement, les éléments d'un alphabet soient appelés « symboles », ils peuvent parfaitement dans la pratique davantage désigner des « mots » ou même éventuellement des « phrases » au sens commun de ces deux termes.

La notion de langage formel est en réalité « générale » et ne se limite pas à la définition des seuls langages informatiques mais vise plus généralement à la validation d'une expression formée de symboles appartenant à un alphabet quelconque préalablement défini.

On donne ci-dessous trois exemples simples d'alphabets. Ces trois alphabets seront réutilisés par la suite sans être redéfinis.

Exemple 35 :

$☯ = \{0, 1\}$, l'alphabet « binaire » :
L'alphabet ☯ ne contient que deux symboles qui sont « 0 » et « 1 ». Ces deux symboles sont les seuls nécessaires à l'écriture des nombres binaires tel que le nombre « 11111100001 ».

$🖩 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, \times, \div\}$, l'alphabet « calculette » :
🖩 est un alphabet de 14 symboles. Il contient les symboles des 10 premiers entiers naturels ainsi que les symboles des 4 opérations de base qu'il est possible d'effectuer sur ces entiers (addition, soustraction, multiplication et division). Ces 14 symboles peuvent servir de base à la construction d'expressions arithmétiques simples tel que par exemple « $4 \times 1010 \div 2 - 3$ ».

$💡 = \{façonne, la, langage, le, pensée\}$, l'alphabet « philosophe » :
L'alphabet 💡 est formé de 6 symboles. Ici, les symboles de 💡 sont en réalité des mots au sens commun du terme. Cet alphabet sera utilisé par la suite pour former des phrases comme par exemple la phrase « le langage façonne la pensée ».

D'un point de vue formel, « 11111100001 », « $4 \times 1010 \div 2 - 3$ » et « *le langage façonne la pensée* » désignent respectivement des **mots sur les alphabets ☯, 🖩 et 💡**. Intuitivement un mot sur un alphabet est une suite ordonnée de symboles de cet alphabet.

La définition suivante formalise cette notion intuitive de suite ordonnée d'éléments d'un ensemble. Un mot sur un alphabet est alors vu formellement comme une association entre les différentes positions des lettres du mot avec le symbole de l'alphabet correspondant à cette lettre. En d'autres termes, un mot sur un alphabet n'est autre qu'un n-uplet d'éléments de l'alphabet ou encore une famille d'éléments d'un alphabet indexé sur I_n.

Définition 35 (Mot sur un alphabet) :
Étant donné un alphabet A, on appelle ***mot sur*** *A tout n-uplet d'éléments de A où n est un entier naturel. On appelle donc* ***mot sur*** *A tout application w de I_n dans A où $n \in \mathbb{N}$:*

$$w : \begin{cases} I_n & \longrightarrow & A \\ i & \longrightarrow & w_i \end{cases}$$

Remarque 8 :

- *Pour tout alphabet A, il existe un unique 0-uplet d'éléments de A appelé le* ***mot vide*** *et noté ε_A. Le mot vide sur A correspond alors à l'unique application vide $\varepsilon_A : \varnothing \longrightarrow A$.*
- *Pour tout alphabet A, l'ensemble $A_n = \mathcal{A}(I_n, A)$ des applications de I_n dans A est l'ensemble des mots sur A composés de n symboles de A.*

Bien que la notion de mot sur un alphabet A soit défini à l'aide de la notion d'application, on utilisera dans la pratique des notations simples pour désigner un mot. En fonction de la confusion que peut engendrer la notation, un mot w sur un alphabet A composé de n symboles pourra être noté au choix :

$$\begin{array}{lcll} & (w_1, w_2, \ldots, w_n) & \text{avec} & \forall i \in [\![1\,;\,n]\!], w_i = w(i) \\ \text{ou bien :} & w_1\ w_2\ \ldots\ w_n & \text{avec} & \forall i \in [\![1\,;\,n]\!], w_i = w(i) \\ \text{ou encore :} & w_1 w_2 \ldots w_n & \text{avec} & \forall i \in [\![1\,;\,n]\!], w_i = w(i) \end{array}$$

Ainsi, pour tout alphabet A et tout symbole $w \in A$ de cet alphabet, la notation « w » peut alors suivant le contexte désigner soit le mot (w) en tant que 1-uplet d'élément de A soit le symbole de A lui même.
Le mot vide sera quant à lui noté ε_A ou $()$.

Exemple 36 :

- *$w^{◕} = 11111100001$ est un mot sur ◕. C'est un 11-uplet d'éléments de l'alphabet ◕ tel que pour tout $i \in \{1,2,3,4,5,6,11\}$, $w^{◕}(i) = 1$ et pour tout $i \in \{7,8,9,10\}$, $w^{◕}(i) = 0$. On peut le noter au choix :*

$$11111100001 \equiv 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1 \equiv (1,1,1,1,1,1,0,0,0,0,1)$$

$$\begin{array}{rcccccccccccc} i & = & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ w_i^{◕} & = & w_1^{◕} & w_2^{◕} & w_3^{◕} & w_4^{◕} & w_5^{◕} & w_6^{◕} & w_7^{◕} & w_8^{◕} & w_9^{◕} & w_{10}^{◕} & w_{11}^{◕} \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ w^{◕}(i) & = & (1, & 1, & 1, & 1, & 1, & 1, & 0, & 0, & 0, & 0, & 1) \end{array}$$

- *$w^{▦} = 4 \times 1010 \div 2 - 3$ est un mot sur ▦. C'est un 10-uplet d'éléments de ▦ dont les différentes notations sont :*

$$4 \times 1010 \div 2 - 3 \equiv 4\ \times\ 1\ 0\ 1\ 0\ \div\ 2\ -\ 3 \equiv (4, \times, 1, 0, 1, 0, \div, 2, -, 3)$$

$$\begin{array}{rcccccccccc} i & = & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ w_i^{▦} & = & w_1^{▦} & w_2^{▦} & w_3^{▦} & w_4^{▦} & w_5^{▦} & w_6^{▦} & w_7^{▦} & w_8^{▦} & w_9^{▦} & w_{10}^{▦} \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ w^{▦}(i) & = & (4, & \times, & 1, & 0, & 1, & 0, & \div, & 2, & -, & 3) \end{array}$$

— $w^{\text{💡}} = lelangagefaçonnelapensée$ *est un mot sur* 💡. *C'est une* 5*-uplet d'éléments de* 💡 *dont les différentes notations sont :*

$$\underbrace{lelangagefaçonnelapensée}_{\text{notation inadaptée !}} \equiv le\ langage\ façonne\ la\ pensée \equiv (le, langage, façonne, la, pensée)$$

$$\begin{array}{rcccccc} i & = & 1 & 2 & 3 & 4 & 5 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ w_i^{\text{💡}} & = & w_1^{\text{💡}} & w_2^{\text{💡}} & w_3^{\text{💡}} & w_4^{\text{💡}} & w_5^{\text{💡}} \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ w^{\text{💡}}(i) & = & le & langage & façonne & la & pensée \end{array}$$

Un mot w sur un alphabet A est un n-uplet d'éléments de A où n est un entier naturel. n correspond intuitivement au nombre de symboles qui composent le mot w et donc à sa longueur noté $|w|$. La définition suivante formalise cette notion de « longueur d'un mot ».

Définition 36 (Longueur d'un mot) :
Étant donné un alphabet A et un mot w sur A, la ***longueur de*** *w noté $|w|$ est définie par :*

$$|w| = card\,(dom\,(w))$$

Remarque 9 :
Pour tout alphabet A :

— $|\varepsilon_A| = card\,(dom\,(\varepsilon_A)) = card\,(\varnothing) = 0$.

— $\forall n \in \mathbb{N}^*, \forall w \in A_n,\ |w| = card\,(dom\,(w)) = card\,([\![1\,;\,n]\!]) = n$.

Ainsi, tout n-uplet d'éléments d'un alphabet A est de longueur n. Inversement, l'ensemble A_n est l'ensemble des mots sur A de longueurs n.

Propriété 4 (Égalité de deux mots) :
Étant donné un alphabet A et deux mots x et y sur A on a :

$$x = y \Leftrightarrow \begin{cases} |x| = |y| \\ \forall i \in I_{|x|},\ x(i) = y(i) \end{cases}$$

Remarque 10 :
En d'autres termes, deux mots sur un même alphabet A sont égaux si il sont de mêmes longueurs et que leurs symboles sont identiques et dans le même ordre. Si x et y s'écrivent respectivement $x = x_1x_2\dots x_n$ et $y = y_1y_2\dots y_m$ où n et m sont deux entiers naturels on alors :

$$x = y \Leftrightarrow (n = m\ et\ x_1 = y_1\ et\ x_2 = y_2\ et\ \dots\ et\ x_n = y_n)$$

.

Preuve :
Triviale (simple réécriture de la condition d'égalité de deux applications).

Exemple 37 :

— 0 *et* 1 *sont deux mots de* ☯ *de longueur* 1. *Ils correspondent aux* 1*-uplet* (0) *et* (1) *d'éléments de A et sont assimilables aux symboles de l'alphabet A. Par extension*

l'ensemble A_1 des 1-uplets d'éléments de A est assimilable à l'alphabet A.

— 11111100001 *est un mot sur* ◕ *de longueur* $|11111100001| = 11$

— $4 \times 1010 \div 2 - 3$ *est un mot sur* 🖩 *de longueur* $|4 \times 1010 \div 2 - 3| = 10$

— *le langage façonne la pensée est un mot sur* 💡 *de longueur*

$$|\textit{le langage façonne la pensée}| = 5$$

La notion d'alphabet, de mot sur un alphabet et de longueur d'un mot étant défini, il est désormais possible de définir l'ensemble A^* des mots sur un alphabet A. Les ensembles A_0, A_1, A_2, etc.correspondent respectivement aux ensembles des mots de tailles 0, 1, 2, etc. Ainsi, l'ensemble de tous les mots n'est rien d'autre que la réunion de tous ces ensembles.

Définition 37 (Ensemble des mots sur un alphabet) :
Étant donné un alphabet A on note :

— A^+ *l'ensemble des mot non vide sur A défini par :*

$$A^+ = \bigcup_{n \in \mathbb{N}^*} A_n$$

— A^* *l'ensemble des mots sur A défini par :*

$$A^* = \bigcup_{n \in \mathbb{N}} A_n$$

Remarque 11 :
Pour tout alphabet A on a clairement : $A^* = A^+ \cup \{\varepsilon_A\}$.

Exemple 38 :

$$\begin{aligned}
◕^* &= \bigcup_{n \in \mathbb{N}^*} ◕_n \\
&= ◕_0 \cup ◕_1 \cup ◕_2 \cup ◕_3 \cup \dots \\
&= \{\varepsilon_◕\} \cup \{0,1\} \cup \{00,01,10,11\} \cup \{000,001,010,011,100,101,110,111\} \cup \dots \\
&= \{\varepsilon_◕, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \dots\}
\end{aligned}$$

$$\begin{aligned}
🖩^* &= \bigcup_{n \in \mathbb{N}^*} 🖩_n \\
&= 🖩_0 \cup 🖩_1 \cup 🖩_2 \cup \dots \\
&= \{\varepsilon_🖩\} \cup \{0,1,\dots,9,+,-,\times,\div\} \cup \{00,01,\dots,09,0+,0-,0\times,\dots,9\div,\dots,\times\div,\div\div\} \\
&\quad \cup \dots \\
&= \{\varepsilon_🖩, 0,1,\dots,9,+,-,\times,\div,00,01,\dots,09,0+,0-,0\times,\dots,9\div,\dots,\times\div,\div\div,\dots\}
\end{aligned}$$

D.2 Monoïde libre

La notion de langage sur un alphabet ne requiert en elle même aucun élément théorique supplémentaire pour être définie. Cependant, la notion de grammaire fait, elle, appelle à certaines opérations sur les langages qui nécessite l'introduction de l'opération de « concaténation » sur les mots d'un alphabet.

Cette sous-section s'attache à définir la loi de composition interne de concaténation définie sur l'ensemble des mots d'un alphabet quelconque. Cette loi permet la construction de nouveaux mots à partir de mots initiaux et par simple concaténation des symboles de ces derniers.

Définition 38 (Concaténation) :
Soit A un alphabet. On note $\bowtie_A$: $A^* \times A^* \longrightarrow A^*$ la loi de composition interne de concaténation sur A^* qui à tout couple $(x, y) \in A^* \times A^*$ de mots sur A associe le mot $x \bowtie_A y$ défini comme l'unique mot de $A_{|x|+|y|}$ tel que que :

$$x \bowtie_A y : \left\{ \begin{array}{lll} I_{|x|+|y|} & \longrightarrow & A \\ i & \longrightarrow & \left\{ \begin{array}{ll} x(i) & \text{si } i \leqslant |x| \\ y(i-|x|) & \text{sinon} \end{array} \right. \end{array} \right.$$

De manière plus simple, si x et y s'écrivent respectivement $x = x_1 \dots x_n$ et $y = y_1 \dots y_m$ avec n et m deux entiers naturels alors $x \bowtie_A y$ est le mot sur A dont les symboles sont ceux de x concaténés à ceux de y :

$$x_1 \dots x_n \bowtie_A y_1 \dots y_m = x_1 \dots x_n y_1 \dots y_m$$

$$\begin{array}{ccccccccccc} x \bowtie_A y & = & x_1 & & x_2 & \cdots & x_n & & y_1 & y_2 & \cdots & y_m \\ & & \uparrow & & \uparrow & & \uparrow & & \uparrow & \uparrow & & \uparrow \\ x \bowtie_A y(i) & = & x(1) & & x(2) & \cdots & x(n) & & y(n+1-n) & y(n+2-n) & \cdots & y(n+m-n) \\ & & \uparrow & & \uparrow & & \uparrow & & \uparrow & \uparrow & & \uparrow \\ i & = & 1 & & 2 & \cdots & n & & n+1 & n+2 & \cdots & n+m \\ & & \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{si } i \leqslant n} & & & & & & \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{sinon}} & & & \end{array}$$

Remarque 12 :

- *$\bowtie_A$ est par définition une fonction de $A^* \times A^*$ dans A^* vérifiant $\forall (x, y) \in (A^*)^2, \exists! z \in A^* : z = x \bowtie_A y$. C'est donc bien une loi de composition interne sur A^*.*
- *La loi de composition interne de concaténation est compatible avec la concaténation de deux mots x et y sur deux alphabets distincts A_1 et A_2. x et de y sont alors vus comme deux mots de l'alphabet $A_1 \cup A_2$ et leur concaténation comme le mot $x \bowtie_{A_1 \cup A_2} y$ sur ce même alphabet. Par la suite on n'indicera donc plus la loi de composition interne de concaténation avec l'alphabet sous-jacent et on notera simplement $x \bowtie y$ au lieu de $x \bowtie_{A_1 \cup A_2} y$ (ou de $x \bowtie_A y$ si x et y sont deux mot sur un même alphabet A).*

L'exemple suivant illustre le principe de concaténation de mots sur chacun des trois alphabet ☯, 🖩 et 💡.

Exemple 39 :

sur ☯ : *par exemple* $111111 \bowtie 00001 = 11111100001$

sur 🖩 : *par exemple* $((((4 \bowtie \times) \bowtie 101) \bowtie 0) \bowtie \div 2 -) \bowtie 3 = 4 \times 1010 \div 2 - 3$

sur 💡 : $(le \bowtie langage) \bowtie façonne\ la\ pensée = le\ langage\ façonne\ la\ pensée$

L'opération de concaténation de mots possède un certain nombre de propriétés élémentaire qui sont synthétisée et démontrée ci-après.

Ces propriétés permettrons dans la propriété 6 de munir l'ensemble des mots A^* d'une structure algébrique de Monoïde libre.

Propriété 5 (Propriétés élémentaires de $\bowtie$) :
Étant donné un alphabet A et deux mots $(x,y) \in (A^*)^2$ on a :

1. $\varepsilon_A \bowtie \varepsilon_A = \varepsilon_A$
2. $x \bowtie \varepsilon_A = \varepsilon_A \bowtie x = x$
3. $|x \bowtie y| = |x| + |y|$

Preuve :

1. $I_{|\varepsilon_A|+|\varepsilon_A|} = I_0$ *donc* $\varepsilon_A \bowtie \varepsilon_A$ *est une application de* $\varnothing$ *dans* A. *Cette application étant unique* $\varepsilon_A \bowtie \varepsilon_A = \varepsilon_A$.
2. *Si* $x = \varepsilon_A$ *alors on a trivialement d'après ce qui précède* $x \bowtie \varepsilon_A = \varepsilon_A \bowtie x = \varepsilon_A \bowtie \varepsilon_A = \varepsilon_A = x$. *On suppose désormais que* $x \neq \varepsilon_A$:

   $x \bowtie \varepsilon_A = x$: *Par définition* $x \bowtie \varepsilon_A$ *est une application de* $I_{|x|+|\varepsilon_A|} = I_{|x|}$ *dans* A^*. *Ainsi :*
   $$\left\{\begin{array}{l} |x \bowtie \varepsilon_A| = |x| \\ \forall i \in I_{|x|}, i \leqslant |x| \end{array}\right. \Rightarrow \left\{\begin{array}{l} |x \bowtie \varepsilon_A| = |x| \\ \forall i \in I_{|x|}, x \bowtie \varepsilon_A(i) = x(i) \end{array}\right. \Rightarrow x \bowtie \varepsilon_A = x$$
   $\varepsilon_A \bowtie x = x$: $\varepsilon_A \bowtie x$ *est une application de* $I_{|\varepsilon_A|+|x|} = I_{|x|}$ *dans* A^*. *Ainsi :*
   $$\left\{\begin{array}{l} |\varepsilon_A \bowtie x| = |x| \\ \forall i \in I_{|x|}, i > 0 \end{array}\right. \Rightarrow \left\{\begin{array}{l} |x \bowtie \varepsilon_A| = |x| \\ \forall i \in I_{|x|}, x \bowtie \varepsilon_A(i) = x(i-0) = x(i) \end{array}\right. \Rightarrow \varepsilon_A \bowtie x = x$$
3. *Par définition* $x \bowtie y$ *est une application de* $I_{|x|+|y|}$ *dans* A^* *donc* $|x \bowtie y| = card\left(dom\left(x \bowtie y\right)\right) = card\left(I_{|x|+|y|}\right) = |x| + |y|$

Propriété 6 (Monoïde libre) :
Pour tout alphabet A, $(A^*, \bowtie)$ est un monoïde libre de neutre ε_A et de base A_1.

Preuve :
En effet, pour tout alphabet A :

$\bowtie$ **est une loi de composition interne sur** A^* **:** *Par définition.*

ε_A **est neutre pour** $\bowtie$ **:** *en effet d'après la propriété 5 p. 259,* $\forall x \in A^*, \varepsilon_A \bowtie x = x \bowtie \varepsilon_A = x$.

$\bowtie$ **est associative :** *Pour tout* $(x,y,z) \in (A^*)^3$ *et tout* $i \in [\![1\,;\ |x|+|y|+|z|]\!]$ *on a :*
$$\begin{array}{rcl} ((x \bowtie y) \bowtie z)(i) & = & \left\{\begin{array}{ll} (x \bowtie y)(i) & si\ i \leqslant |x|+|y| \\ z(i-|x|-|y|) & sinon \end{array}\right. \\ & = & \left\{\begin{array}{ll} x(i) & si\ i \leqslant |x| \\ y(i-|x|) & si\ i-|x| \leqslant |y| \\ z(i-|x|-|y|) & sinon \end{array}\right. \\ & = & \left\{\begin{array}{ll} x(i) & si\ i \leqslant |x| \\ (y \bowtie z)(i-|x|) & sinon \end{array}\right. \\ ((x \bowtie y) \bowtie z)(i) & = & (x \bowtie (y \bowtie z))(i) \end{array}$$

A_1 **une base :** *Tout mot x se décompose de manière unique comme une concaténation de mots de A_1 (1-uplets d'éléments de A). On prouve en effet par récurrence sur $n \in \mathbb{N}^*$ que pour tout tout $x \in A_n$, $x = \bowtie_{i=1}^{n} (x_i)$ (où $\forall i \in I_n, x_i = x(i)$). Le mot vide ε_A est quant à lui obtenu par concaténation vide. L'unicité de cette décomposition est assurée par la propriété 4 p. 256.*

Par la suite, on omettra parfois le symbole $\bowtie$ pour désigner la concaténation de deux mots. Ainsi pour deux mots u et v sur un alphabet quelconque on notera simplement uv pour désigner la concaténation $u \bowtie v$.

Exemple 40 :
11111100001 *est un mot du monoïde libre* $(☯^*, \bowtie)$. *Sa décomposition unique sur* $☯_1$ *est :*

$$11111100001 = 1 \bowtie 1 \bowtie 1 \bowtie 1 \bowtie 1 \bowtie 1 \bowtie 0 \bowtie 0 \bowtie 0 \bowtie 0 \bowtie 1$$

D.3 Langages

On s'intéresse dans cette sous-section à la notion de langage. Dans la pratique, un langage sur un alphabet consiste simplement en un sous-ensemble de l'ensemble des mots sur cet alphabet. La définition d'un langage permet ainsi d'établir au sein de l'ensemble de tous les mots qu'il est possible de former sur un alphabet quelconque, une distinction entre des mots « valides » qui appartiennent alors à ce langage et d'autres qui ne le sont pas.

Prenons à titre d'exemple l'alphabet 🖩. Cet alphabet permet de former des expressions algébriques telles que $1 + 2$ ou encore $1 \div 2$ mais aussi le mot $0 \div \times - 3$ qui lui ne correspond pas à une expression algébrique « syntaxiquement correcte ». La définition d'un langage sur 🖩 ne contenant que les expressions arithmétiques syntaxiquement valides peut alors être envisagée.

La notion de langage ne définit pas de règles de construction de « mot valide » mais s'intéresse simplement à leur regroupement au sein d'un même ensemble.

Définition 39 (Langage) :
On appelle langage sur un alphabet A toute partie $\mathcal{L}$ de l'ensemble A^*.

Remarque 13 :

- *Tout ensemble $\mathcal{L} \in \mathcal{P}(A)$ où A est un alphabet est donc un langage sur A.*
- *Le langage vide $\mathcal{L} = \varnothing$ et indépendant de l'alphabet A.*
- *Pour tout $n \in \mathbb{N}$, A_n est le langage contenant l'ensemble des mots de longueur n.*

La définition suivante introduit l'opération de produit pour les langages. De manière analogue à l'opération de concaténation sur les mots d'un alphabet, le produit de langages permet de construire de nouveaux langages à partir de langages existants. Cette opération permet dans un second temps d'introduire l'étoile de Kleene d'un langage.

Ces deux notions formelles sont particulièrement utiles pour définir les grammaires formelles et plus précisément pour définir les règles de production d'une grammaire.

Définition 40 (Produit de langages) :
On appelle ***produit de langages*** *la relation qui à tout couple de langage $(\mathcal{L}_1, \mathcal{L}_2) \in \mathcal{P}(A_1) \times \mathcal{P}(A_2)$ sur des alphabets A_1 et A_2 associe le langage $\mathcal{L}_1 \bullet \mathcal{L}_2$ sur l'alphabet $A_1 \cup A_2$ définit par :*

$$\mathcal{L}_1 \bullet \mathcal{L}_2 = \{x \bowtie y \mid (x, y) \in \mathcal{L}_1 \times \mathcal{L}_2\}$$

Remarque 14 :

- *Le produit $\mathcal{L}_1 \bullet \mathcal{L}_2$ n'est donc rien d'autre que le langage contenant tous les mots formés de la concaténation d'un mot de $\mathcal{L}_1$ avec un mot de $\mathcal{L}_2$.*
- *Le produit de langage est une relation associative. Cette associativité est héritée de celle de la loi de composition interne de concaténation pour les mots. On pourra donc se passer de parenthèses : pour tout langage $\mathcal{L}_1$, $\mathcal{L}_2$ et $\mathcal{L}_3$ sur trois alphabets A_1, A_2 et A_3 on a donc $(\mathcal{L}_1 \bullet \mathcal{L}_2) \bullet \mathcal{L}_3 = \mathcal{L}_1 \bullet (\mathcal{L}_2 \bullet \mathcal{L}_3) = \mathcal{L}_1 \bullet \mathcal{L}_2 \bullet \mathcal{L}_3$.*

Exemple 41 :
$\mathcal{L}_1 = \{un, mot\}$ *et* $\mathcal{L}_2 = \{ion, if\}$ *sont deux langages sur l'alphabet français classique (26 lettres). On a :*

$$\mathcal{L}_1 \bullet \mathcal{L}_2 = \{un \bowtie ion, un \bowtie if, mot \bowtie if, mot \bowtie ion\} = \{union, unif, motif, motion\}$$

Définition 41 (Puissance d'un langage et opération de Kleene) :
Soit A un alphabet et $\mathcal{L} \in \mathcal{P}(A)$ un langage sur A :

puissance d'un langage : *Pour tout $k \in \mathbb{N}$, on appelle* ***puissance k-iéme du langage*** *$\mathcal{L}$ notée $\mathcal{L}^k$ le langage sur A définit de manière récursive par :*

$$\begin{cases} \mathcal{L}^0 = \{\varepsilon_A\} \\ \forall k \in \mathbb{N}^*,\ \mathcal{L}^k = \mathcal{L}^{k-1} \bullet \mathcal{L} \end{cases}$$

En d'autre termes on a : $\mathcal{L}^0 = \{\varepsilon_A\}$ et pour tout entier k non nul $\mathcal{L}^k = \overset{k}{\underset{i=1}{\bullet}} \mathcal{L}$.

étoile propre d'un langage : *On appelle* ***étoile propre*** *ou* ***itéré stricte*** *du langage $\mathcal{L}$ le langage $\mathcal{L}^+$ définit par :*

$$\mathcal{L}^+ = \bigcup_{k \in \mathbb{N}^*} \mathcal{L}^k = \mathcal{L} \cup \mathcal{L}^2 \cup \dots$$

étoile d'un langage : *On appelle* ***fermeture de Kleene*** *ou* ***étoile*** *ou* ***itéré*** *du langage $\mathcal{L}$ le langage $\mathcal{L}^*$ définit par :*

$$\mathcal{L}^* = \bigcup_{k \in \mathbb{N}} \mathcal{L}^k = \{\varepsilon_A\} \cup \mathcal{L} \cup \mathcal{L}^2 \cup \dots$$

Remarque 15 :

- *Pour tout alphabet A le langage A_n des mots de longueur $n \in \mathbb{N}$ est la puissance n-iéme du langage A_1 des mots de longueur 1. En d'autres termes pour tout alphabet A et pour tout $n \in \mathbb{N}$, $A_n = A_1^n$. En particulier $A_0 = \{\varepsilon_A\} = A_1^0$.*
- *Pour tout alphabet A l'ensemble A^* des mots sur A est la fermeture de Kleene du langage A_1 des mots de longueur 1. En d'autres termes pour tout alphabet A, on a : $A^* = A_1^*$. La notation employée pour la fermeture de Kleene reste ainsi cohérente à l'assimilation de l'alphabet A et de A_1 près.*

D.4 Grammaire

Définition 42 (Grammaire) :
Une grammaire est un quadruplet $\mathcal{G} = (V_T, V_N, S, R)$ tel que :

- *V_T est un alphabet appelé **vocabulaire terminal** et dont les symboles sont appelés **terminaux**,*
- *V_N est un alphabet disjoint de V_T ($V_T \cap V_N = \varnothing$) appelé **vocabulaire non-terminal** et dont les symboles sont appelés **non-terminaux** ou **variables**,*
- *$V = V_T \cup V_N$ est appelé **vocabulaire** ou **vocabulaire général** de la grammaire $\mathcal{G}$,*
- *$S \in V$ est un non-terminal particulier appelé **source** ou **axiome**,*
- *R est une partie finie de $V^* \bullet V_N \bullet V^* \times V^*$ dont les éléments sont appelés **règles de production**. Une règle de production $(u, v) \in R \subseteq V^* \bullet V_N \bullet V^* \times V^*$ est alors noté :*

$$u \to v$$

Remarque 16 :

- *Les terminaux sont classiquement notés en minuscule.*
- *Les non-terminaux sont classiquement notés en majuscule.*
- *La construction de l'ensemble des règles de production impose la présence d'un non-terminal dans le mot entrant de la règle de production. Il est alors également possible de définir l'ensemble R comme une partie finie de $V^* \setminus V_T^*$*
- *V_N et V_T forment une partition du vocabulaire général V de la grammaire.*

D'un point de vue général, une grammaire peut être vue comme un objet formel permettant de « construire » un langage. Le langage alors reconnu et un langage sur l'ensemble V_T des terminaux et non pas sur l'ensemble V_N des variables qui ne sert que dans le processus interne de construction des mots du langage.

Les « règles de construction » des mots du langage sont données par l'intermédiaire des règles de production (ensemble R) qui sont en réalité des règles de « réécriture » de mots composés à la fois de variables et de terminaux.

Une règle de production $x \to y$ peut intuitivement être comprise comme la possibilité de remplacer une occurrence du mot x dans un mot quelconque $u = \dots x \dots$ par le mot y. Ce processus réécrit alors le $u = \dots x \dots$ en un mot $v = \dots y \dots$. On dit alors que le mot v dérive de u par la grammaire et on note $u \Rightarrow v$ ou de manière plus détaillée $u \overset{x \to y}{\Longrightarrow} v$.

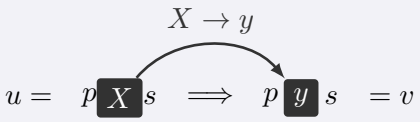
Par application récursive et aléatoire de ces règles de production à partir de l'axiome S on peut ainsi construire une multitude de mots qui constituent ainsi un langage.

$$\begin{array}{lllllllllll}
S & \overset{a\to a'}{\Longrightarrow} & u_1 & \overset{b\to b'}{\Longrightarrow} & v_1 & \overset{c\to c'}{\Longrightarrow} & w_1 & \overset{\cdots\to\cdots}{\Longrightarrow} & \cdots & \overset{\cdots\to\cdots}{\Longrightarrow} & x_1 \\
S & \overset{e\to e'}{\Longrightarrow} & u_2 & \overset{f\to f'}{\Longrightarrow} & v_2 & \overset{g\to g'}{\Longrightarrow} & w_2 & \overset{\cdots\to\cdots}{\Longrightarrow} & \cdots & \overset{\cdots\to\cdots}{\Longrightarrow} & x_2 \\
S & \overset{\cdots\to\cdots}{\Longrightarrow} & \cdots & \overset{\cdots\to\cdots}{\Longrightarrow} & \cdots & \overset{\cdots\to\cdots}{\Longrightarrow} & \cdots & \overset{\cdots\to\cdots}{\Longrightarrow} & \cdots & \overset{\cdots\to\cdots}{\Longrightarrow} & \cdots
\end{array} \longrightarrow \text{Langage sur } V_T$$

Avant d'illustrer les propos précédents par des exemples, on formalise la notion de dérivation :

Définition 43 (Dérivation élémentaire) :
*Étant donné une grammaire $\mathcal{G} = (V_T, V_N, S, R)$ et $(u, v) \in (V^*)^2$ (avec $V = V_T \cup V_N$) un couple de mot, on dit que v **dérive directement de** u **par la grammaire** $\mathcal{G}$ et on note alors $u \Rightarrow v$ si et seulement si $u = v$ ou bien si il existe une règle de production $X \to y$ de R et deux mots $(p, s) \in (V^*)^2$ (préfixe et suffixe) tels que :*

- *$u = pXs$ (X est un sous-mot de u)*
- *$v = pys$ (y est un sous-mot de v)*

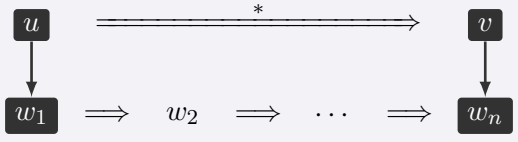


Remarque 17 :

- *On emploiera également la notation $u \xRightarrow{X \to y} v$ pour signifier que v dérive directement de u par l'intermédiaire de la règle de production $X \to y$.*
- *Le mot X d'une règle de production $X \to y$ comporte nécessairement une variable, c'est pour cette raison qu'on le note en majuscule.*
- *Lorsque $u \xRightarrow{X \to y} v$, les mots p et s tel que $u = pXs$ et $v = pys$ appartiennent à V^* et peuvent être le mot vide ε_V.*

Définition 44 (Dérivation) :
*Étant donné une grammaire $\mathcal{G} = (V_T, V_N, S, R)$ et $(u, v) \in (V^*)^2$ (avec $V = V_T \cup V_N$) un couple de mots, on dit que v **dérive de** u **par la grammaire** $\mathcal{G}$ et on note alors $u \overset{*}{\Rightarrow} v$ si et seulement si il existe un entier $n \geqslant 2$ et un n-uplet $w = (w_1, \ldots, w_n) \in \mathcal{A}(I_n, V^*)$ de mots sur V tel que :*

- *$w_1 = u$*
- *$\forall i \in I_{n-1}, w_i \Rightarrow w_{i+1}$*
- *$w_n = v$*



Remarque 18 :
Un mot v dérive d'un mot u si il peut être obtenu par un enchaînement de dérivations élémentaires à partir du mot u.

Remarque 17 (Méta-symboles dans les règles de production) :
*Pour plus de clarté et de concision dans l'écriture des règles de production d'une grammaire, on emploiera parfois des **méta-symboles** permettant de fusionner certains ensembles de règles de production en une seule. Ces simplifications sont issues du métalangage BNF et de ses extensions bien que les méta-symboles que l'on emploiera dans ce mémoire ne soient pas syntaxiquement rigoureusement identiques à ceux de cette notation. Ainsi, Étant donnée $\mathcal{G} = (V_T, V_N, S, R)$ une grammaire formelle ou $V = V_T \cup V_N$ est la grammaire générale de $\mathcal{G}$ on pourra notamment employer :*
Le méta-symbole « | » et des parenthèses pour « fusionner » l'ensemble des règles de réécri-

ture possible d'un même mot Y en une seule règle de production :

$$\begin{cases} X \to w_a Y w_b \\ Y \to Y_0 \\ Y \to Y_1 \\ \cdots \\ Y \to Y_n \end{cases} \Leftrightarrow \quad X \to w_a(Y_0 \mid Y_1 \mid \cdots \mid Y_n)w_b \quad \text{avec} \quad \begin{cases} (X,Y) \in (V^* \bullet V_N \bullet V^*)^2 \\ n \in \mathbb{N}^* \\ (w_a, w_b) \in (V^*)^3 \\ \forall i \in [\![0\,;\ n]\!], Y_i \in V^* \end{cases}$$

Le méta-symbole « ? » et des parenthèses pour fusionner un ensemble de règles de production rendant intrinsèquement un mot Y optionnel :

$$\begin{cases} X \to w_a Z w_b \\ Z \to Y \\ Z \to \varepsilon \end{cases} \Leftrightarrow \quad X \to w_a\,(Y)?w_b \qquad \text{avec} \quad \begin{cases} (X,Z) \in (V^* \bullet V_N \bullet V^*)^2 \\ (w_a, w_b, Y) \in (V^*)^3 \end{cases}$$

Le méta-symbole « $*$ » et des parenthèses pour fusionner un ensemble de règles de production permettant intrinsèquement à un mot Y d'apparaître un nombre indéfini de fois (éventuellement 0) :

$$\begin{cases} X \to w_a Z w_b \\ Z \to YZ \\ Z \to \varepsilon \end{cases} \Leftrightarrow \quad X \to w_a\,(Y)^*\,w_b \qquad \text{avec} \quad \begin{cases} (X,Z) \in (V^* \bullet V_N \bullet V^*)^2 \\ (w_a, w_b, Y) \in (V^*)^3 \end{cases}$$

Le méta-symbole « $+$ » et des parenthèses pour fusionner un ensemble de règles de production permettant intrinsèquement à un mot Y d'apparaître au moins une fois puis un nombre indéfini de fois (éventuellement 0) :

$$\begin{cases} X \to w_a Z w_b \\ Z \to Y Z' \\ Z' \to Y \\ Z' \to \varepsilon \end{cases} \Leftrightarrow \quad X \to w_a\,(Y)^+\,w_b \qquad \text{avec} \quad \begin{cases} (X,Z,Z') \in (V^* \bullet V_N \bullet V^*)^3 \\ (w_a, w_b, Y) \in (V^*)^3 \end{cases}$$

Exemple 42 :

On définit dans cette exemple une grammaire permettant de construire les mots correspondant aux nombres binaires. Le vocabulaire terminal de la grammaire est donc l'alphabet $\textcircled{\scriptsize 2}$. On impose ici qu'un nombre binaire ne peut pas commencer par le symbole « 0 » sauf si le nombre en question est 0 (le mot 010 ne peut donc pas être « construit » par cette grammaire contrairement au mot 10).

On définit pour cela deux variables : S qui sera également l'axiome et une variable B. L'ensemble des non terminaux est donc l'ensemble $V_{N\textcircled{\scriptsize 2}} = \{S, B\}$ et le vocabulaire général de la grammaire est donc l'ensemble $V_{\textcircled{\scriptsize 2}} = \textcircled{\scriptsize 2} \cup V_{N\textcircled{\scriptsize 2}} = \{0, 1, S, B\}$.

Enfin, on définit l'ensemble $R_{\textcircled{\scriptsize 2}}$ des règles de production tel que $R_{\textcircled{\scriptsize 2}}$ contient les règles de production suivantes :

$$\begin{aligned} S &\to (0 \mid 1B) \\ B &\to (0B \mid 1B \mid \varepsilon_V) \end{aligned}$$

La grammaire ainsi définie est la grammaire $\mathcal{G}_{\textcircled{\scriptsize 2}} = (\textcircled{\scriptsize 2}, V_{N\textcircled{\scriptsize 2}}, S, R_{\textcircled{\scriptsize 2}})$. On peut ainsi obtenir :

le mot 0 : $S \overset{*}{\Rightarrow} 0$ *car :* $S \xrightarrow{S\to 0} 0$.

le mot 1 : $S \overset{*}{\Rightarrow} 1$ *car :* $S \xrightarrow{S\to 1B} 1B \xrightarrow{B\to\varepsilon_{V_{\textcircled{\scriptsize 2}}}} 1\varepsilon_{V_{\textcircled{\scriptsize 2}}} = 1$.

le mot 10 : $S \overset{*}{\Rightarrow} 10$ *car :* $S \xrightarrow{S\to 1B} 1B \xrightarrow{B\to 0B} 10B \xrightarrow{B\to\varepsilon_{V_{\textcircled{\scriptsize 2}}}} 10\varepsilon_{V_{\textcircled{\scriptsize 2}}} = 10$

le mot 11 : $S \overset{*}{\Rightarrow} 11$ *car :* $S \xrightarrow{S\to 1B} 1B \xrightarrow{B\to 1B} 11B \xrightarrow{B\to\varepsilon_{V_{\textcircled{\scriptsize 2}}}} 11\varepsilon_{V_{\textcircled{\scriptsize 2}}} = 11$

le mot 100 : $S \overset{*}{\Rightarrow} 100$ *car :* $S \xrightarrow{S\to 1B} 1B \xrightarrow{B\to 0B} 10B \xrightarrow{B\to 0B} 100B \xrightarrow{B\to\varepsilon_{V_{\textcircled{\scriptsize 2}}}} 100$

Exemple 43 :

On définit ici la grammaire $\mathcal{G}_{♀} = \{♀, N_{♀}, P, R_{♀}\}$ *où :*

- $N_{♀} = \{P, G_n, V, D_{♀}, D_{♂}, N_{♀}, N_{♂}\}$. *Chacune de ces variables désignent une catégorie grammaticale de la langue française :* P *: « phrase »,* G_n *: « groupe nominal »,* V *: « verbe »,* $D_{♀}$ *: « déterminant féminin »,* $D_{♂}$ *: « déterminant masculin »,* $N_{♀}$ *: « nom féminin ».* $N_{♂}$ *: « nom masculin ».*
- *La variable phrase* P *est l'axiome de* $\mathcal{G}_{♀}$*,*
- *L'ensemble* $R_{♀}$ *contient les règles de production suivantes :*

$$\begin{array}{ll} P \to G_n\ V\ G_n & V \to façonne \\ G_n \to (D_{♀}\ N_{♀} \mid D_{♂}\ N_{♂}) & D_{♀} \to la \\ & N_{♂} \to langage \\ & D_{♂} \to le \\ & N_{♀} \to pensée \end{array}$$

Cette grammaire permet notamment de dériver les deux mots

$$le\ langage\ façonne\ la\ pensée$$

et

$$la\ pensée\ façonne\ le\ langage$$

En effet :

$$\begin{array}{ll} P & \\ \xRightarrow{P \to G_n\ V\ G_n} & G_n\ V\ G_n \\ \xRightarrow{V \to façonne} & G_n\ façonne\ G_n \\ \xRightarrow{G_n \to D_{♂}\ N_{♂}} & D_{♂}\ N_{♂}\ façonne\ G_n \\ \xRightarrow{D_{♂} \to le} & le\ N_{♂}\ façonne\ G_n \\ \xRightarrow{N_{♂} \to langage} & le\ langage\ façonne\ G_n \\ \xRightarrow{G_n \to D_{♀}\ N_{♀}} & le\ langage\ façonne\ D_{♀}\ N_{♀} \\ \xRightarrow{D_{♀} \to la} & le\ langage\ façonne\ la\ N_{♀} \\ \xRightarrow{N_{♀} \to pensée} & le\ langage\ façonne\ la\ pensée \end{array} \qquad \begin{array}{ll} P & \\ \xRightarrow{P \to G_n\ V\ G_n} & G_n\ V\ G_n \\ \xRightarrow{V \to façonne} & G_n\ façonne\ G_n \\ \xRightarrow{G_n \to D_{♀}\ N_{♀}} & D_{♀}\ N_{♀}\ façonne\ G_n \\ \xRightarrow{D_{♀} \to la} & la\ N_{♀}\ façonne\ G_n \\ \xRightarrow{N_{♀} \to pensée} & la\ pensée\ façonne\ G_n \\ \xRightarrow{G_n \to D_{♂}\ N_{♂}} & la\ pensée\ façonne\ D_{♂}\ N_{♂} \\ \xRightarrow{D_{♂} \to le} & la\ pensée\ façonne\ le\ N_{♂} \\ \xRightarrow{N_{♂} \to langage} & la\ pensée\ façonne\ le\ langage \end{array}$$

elle permet également de dériver les mots

$$la\ pensée\ façonne\ la\ pensée$$

et

$$le\ langage\ façonne\ le\ langage$$

Une grammaire permet en somme de construire des mots. Ces mots sont construits en dérivant de manière récursive l'axiome de cette grammaire. La relation de dérivation permet de construire des mots sur le vocabulaire général.

Ces derniers peuvent ainsi contenir des variables appartenant à l'alphabet non terminal cependant une grammaire permet in fine de définir un langage sur le vocabulaire terminal uniquement.

Ainsi, les mots contenant un symbole appartenant au vocabulaire non-terminal doivent être vus comme des mots intermédiaires dans le processus de dérivation qui a comme but ultime la construction d'un mot composé uniquement de symbole terminaux.

Les variables d'une grammaire permettent en réalité de définir des catégorie syntaxique qui peuvent correspondre à plusieurs mots sur l'alphabet des terminaux. Dans l'exemple précédent, ces catégories syntaxiques correspondaient à différentes catégories grammaticales de la langue française (nom masculin, groupe nominal, etc.).

Le langage d'une grammaire est donc défini comme l'ensemble des mots sur le vocabulaire terminal qu'il est possible de former par dérivation de l'axiome de la grammaire.

Définition 45 (Langage engendré par une grammaire) :
Étant donné $\mathcal{G} = (V_T, V_N, S, R)$ une grammaire, on appelle **langage engendré par** *$\mathcal{G}$ le langage $\mathcal{L}(\mathcal{G})$ sur V_T des mots sur V_T qui dérivent de l'axiome S par la grammaire $\mathcal{G}$:*

$$\mathcal{L}(\mathcal{G}) = \{w \in V_T^* : S \overset{*}{\Rightarrow} w\}$$

Exemple 44 :
On définit la grammaire $\mathcal{G}_{\blacksquare} = \left(\blacksquare, V_{\blacksquare}, E, R\right)$ de telle sorte que le langage $\mathcal{L}\left(\mathcal{G}_{\blacksquare}\right)$ engendré par la grammaire $\mathcal{G}_{\blacksquare}$ contienne les expressions arithmétiques syntaxiquement correctes qu'il est possible de former à partir des nombres entiers positifs et des quatre opérations de base sur les entiers. On définit pas à pas les différentes catégories syntaxiques et non-terminaux correspondant permettant l'engendrement de ce langage :

Catégories syntaxiques « d'opération » et de « chiffre » : *On sépare dans un premier temps les symboles de l'alphabet $\blacksquare$ en deux catégories syntaxiques. Les symboles correspondant à des chiffres sont définis à travers la variable D (digit) tandis que les opérations sont définies à travers une variable O. Les 14 règles de production assurant cette séparation sont alors les suivantes :*

- $O \to + \mid - \mid \div \mid \times$
- $D \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Catégories syntaxique de « nombre » : *On définit le non-terminal N (number) désignant la catégorie syntaxique de nombre. Un nombre N peut alors être défini comme une suite de variable D (i.e. de chiffres) ou comme le mot vide ε_V afin de garantir la finitude de l'enchaînement de chiffre. Ceci est réalisé grâce à la règle de production $N \to DN \mid D\varepsilon_V$.*

$$N \to DN \mid D\varepsilon_V \Rightarrow N \overset{*}{\Rightarrow} DD\cdots D\varepsilon_V$$

Ces règles de production permettent notamment d'obtenir les dérivations suivantes : $N \overset{*}{\Rightarrow} 2$, $N \overset{*}{\Rightarrow} 3$, $N \overset{*}{\Rightarrow} 4$ ou encore $N \overset{*}{\Rightarrow} 1010$ ($N \overset{N\to DN}{\Longrightarrow} DN \overset{D\to 1}{\Longrightarrow} 1N \overset{N\to DN}{\Longrightarrow} 1DN \overset{D\to 0}{\Longrightarrow} 10N \overset{N\to DN}{\Longrightarrow} 10DN \overset{D\to 1}{\Longrightarrow} 101N \overset{N\to D\varepsilon_V}{\Longrightarrow} 101D\varepsilon_V \overset{D\to 0}{\Longrightarrow} 1010$).

Catégories syntaxique « d'expression arithmétique » : *Pour définir la syntaxe d'une expression arithmétique on définit la variable E. Une expression arithmétique est définie comme un enchaînement fini de nombres séparés par des opérateurs (nombre $\to$ opérateur $\to$ nombre $\to$ opérateur $\to \ldots \to$ opérateur $\to$ nombre). En d'autres termes, une expression arithmétique est constituée d'un suivi d'un nombre fini de « motifs » du type « opération nombre ». On définit ainsi une variable E' correspondant au motif « opération nombre ». En somme une expression arithmétique est définie à l'aide des règles de production suivantes :*

- $E \to NE'$
- $E' \to ONE' \mid \varepsilon_V$

$$\left.\begin{array}{l} E \to NE' \mid \varepsilon_V \\ E' \to ONE' \end{array}\right\} \Rightarrow E \overset{*}{\Rightarrow} NONON\cdots ON\varepsilon_V$$

On définit ainsi la grammaire $\mathcal{G}_{\boxplus} = \left(\boxplus, V_{\boxplus}, E, R\right)$ *où :*

— $\boxplus = \{0,1,2,3,4,5,6,7,8,9,+,-,\times,\div\}$

— $V_{\boxplus} = \{D, O, N, E', E\}$

— *R contient les règles de production :*

$$
\begin{aligned}
&D \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\\
&O \to + \mid - \mid \div \mid \times\\
&N \to DN \mid D\varepsilon_V\\
&E \to NE'\\
&E' \to ONE' \mid \varepsilon_V
\end{aligned}
$$

On peut notamment vérifier que $4 \times 1010 \div 2 - 3 \in \mathcal{L}\left(\mathcal{G}_{\boxplus}\right)$. *En effet on a* $E \overset{*}{\Rightarrow} 4 \times 1010 \div 2 - 3$ *:*

$$
\begin{aligned}
E &\xrightarrow{E \to NE'} NE'\\
&\xrightarrow{N \overset{*}{\Rightarrow} 4} 4E'\\
&\xrightarrow{E' \to ONE'} 4ONE'\\
&\xrightarrow{O \to \times} 4 \times NE'\\
&\xrightarrow{N \overset{*}{\Rightarrow} 1010} 4 \times 1010E'\\
&\xrightarrow{E' \to ONE'} 4 \times 1010ONE'\\
&\xrightarrow{O \to \div} 4 \times 1010 \div NE'\\
&\xrightarrow{N \overset{*}{\Rightarrow} 2} 4 \times 1010 \div 2E'\\
&\xrightarrow{E' \to ONE'} 4 \times 1010 \div 2ONE'\\
&\xrightarrow{O \to -} 4 \times 1010 \div 2 - NE'\\
&\xrightarrow{N \overset{*}{\Rightarrow} 3} 4 \times 1010 \div 2 - 3E'\\
&\xrightarrow{E' \to \varepsilon_V} 4 \times 1010 \div 2 - 3
\end{aligned}
$$

Index des systèmes d'organisation des connaissances

ADICAP **A**ssociation pour le **D**éveloppement de l'**I**nformatique en **C**ytologie et en **A**natomie **P**athologique. 65

ATC **A**natomical **T**herapeutic **C**hemical classification. 188, 189

BDSP **B**anque de **D**onnées en **S**anté **P**ublique. 85, 116

CCAM **C**lassification **C**ommune des **A**ctes **M**édicaux. 14, 47, 53–55, 84, 118, 131, 182, 194, 211

CdAM **C**atalogue **d**es **A**ctes **M**édicaux. 54

CIM **C**lassification **I**nternationale des **M**aladies. 62

CIM–10 **C**lassification **I**nternationale des **M**aladies (**10ème** révision). 47, 53–55, 65, 84, 119, 145, 146, 182, 187–189, 191, 201, 211, 217, 251

CIM–O–3 **C**lassification **I**nternationale des **M**aladies **O**ncologiques (**3ème** révision). 65

CIM-9 **C**lassification **I**nternationale des **M**aladies (9^{eme} révision). 60, 62, 251

CLADIMED **CLA**ssification des **DI**spositifs **MÉD**icaux. 207, 251

CPT **C**urrent **P**rocedural **T**erminology. 62, 84

DEWEY® Classification décimale de **Dewey**. 178

DRC **D**ictionnaire des **R**ésultats de **C**onsultation. 251

FMA **F**oundational **M**odel of **A**natomy. 84, 90, 251, 252

GENEONTOLOGY Gene Ontology. 84

HPO **H**uman **P**henotype **O**ntology. 33, 90, 251

ICNP **I**nternational **C**lassification for **N**ursing **P**ractice. 251

IUPAC **I**nternational **U**nion of **P**ure and **A**pplied **C**hemistry. 251

LOINC **L**ogical **O**bservation **I**dentifiers **N**ames and **C**odes. 50, 60, 65, 84

MedDRA **MED**ical **D**ictionary for **R**egulatory **A**ctivities terminologies. 251, 252

MedlinePlus Système d'Organisation de Connaissance **MedlinePlus** édité par la NLM. 178, 251

MeSH **M**edical **S**ubject **H**eadings. 33, 84, 85, 116, 125, 174, 178, 243, 244, 251, 252

MIDAS Nomencalture **Midas**. 178

NCIt **N**ational **C**ancer **I**nstitute **T**hesaurus. 33, 251, 252

NGAP **N**omenclature **G**énérale des **A**ctes **P**rofessionnels. 54

OMIM Online Mendelian Inheritance in Man. 33, 84

orphanet **Orphanet** Rare Disease Ontology. 84, 207

PASCAL Système d'organisation des connaissance exploité dans le cadre du **P**rogramme **A**ppliqué à la **S**élection et à la **C**ompilation **A**utomatique de la **L**ittérature de l'**IN**stitut de l'**I**nformation **S**cientifique et **T**echnique (INIST). 251

PHA Médicaments. 188, 189, 251

RadLex® Terminologie **RadLex** (radiologie). 251, 252

RxNorm **NORM**alized names for Clinical Drugs. 84

SNOMED 3.5 **S**ystematized **N**omenclature **O**f **MED**icine (version 3.5). 33, 62, 65, 178, 251, 252

SNOMED–CT® **S**ystematized **N**omenclature **O**f **MED**icine–**C**linical **T**erms. 47, 50, 60, 84, 90, 178, 194, 207, 251, 252

TSP **T**hesaurus **S**anté **P**ublique. 116, 251, 252

WHO–ART **W**orld **H**ealth **O**rganization **A**dverse Drug **R**eaction **T**erminology. 84