Une approche sémantique pour l'exploitation de données environnementales : application aux données d'un observatoire

Ba-Huy Tran

THÈSE

présentée par

Ba-Huy TRAN

23/11/2017

pour l'obtention du

Doctorat de l'université de La Rochelle
Spécialité : Informatique et Applications

Une approche sémantique pour l'exploitation de données environnementales
Application aux données d'un observatoire

Soutenue publiquement le 23/11/2017 devant le Jury composé de :

M^{me} Thérèse LIBOUREL	Rapporteur	Pr. émérite, Université de Montpellier
M. Jérôme GENSEL	Rapporteur	Pr., Université Grenoble Alpes
M. Christophe CLARAMUNT	Examinateur	Pr., École Navale
M^{me} Christine PLUMEJEAUD	Invitée	Dr., IR, Université de La Rochelle
M. Alain BOUJU	Directeur de thèse	MCF, HDR, Université de La Rochelle

Remerciements

Je souhaiterais tout d'abord remercier mon directeur de thèse, Alain Bouju, pour l'encadrement de mon travail et ses conseils avisés et ses remarques pertinentes tout au long de de cette thèse. Je souhaite également le remercier pour son soutien et sa confiance.

Je tiens également à remercier Thérèse Libourel et Jérôme Gensel pour m'avoir fait l'honneur d'être rapporteurs de cette thèse. Je leur suis très reconnaissant d'avoir accepté cette lourde tâche. Je remercie également Christophe Claramunt d'avoir accepté d'examiner mon travail, ainsi que Christine Plumejeaud-Perreau qui a accepté de participer à mon jury de thèse.

J'adresse un remerciement particulier à Vincent Bretagnolle et les membres de CEBC pour leurs jeux de données et leur collaboration précieux.

Je voudrais remercier très chaleureusement tous les membres du laboratoire L3i pour leur amitié et leur aide pendant ces années de thèse.

Bien évidemment, je remercie mes chers parents et ma chère sœur qui ont toujours cru en moi et qui m'ont toujours soutenu. Malgré la distance qui nous a séparé pendant ces dernières années, vous m'avez encouragé dans les moments difficiles.

Pour finir, je souhaite remercier ma femme qui m'épaule maintenant depuis cinq ans et sans qui rien n'aurait été possible.

Résumé

La nécessité de collecter des observations sur une longue durée pour la recherche sur des questions environnementales a entrainé la mise en place de Zones Ateliers par le CNRS. Ainsi, depuis plusieurs années, de nombreuses bases de données à caractère spatio-temporel sont collectées par différents équipes de chercheurs. Afin de faciliter les analyses transversales entre différentes observations, il est souhaitable de croiser les informations provenant de ces sources de données. Néanmoins, chacune de ces sources est souvent construite de manière indépendante l'une de l'autre, ce qui pose des problèmes dans l'analyse et l'exploitation.

De ce fait, cette thèse se propose d'étudier les potentialités des ontologies à la fois comme objets de modélisation, d'inférence, et d'interopérabilité. L'objectif est de fournir aux experts du domaine une méthode adaptée permettant d'exploiter l'ensemble de données collectées. Étant appliquées dans le domaine environnemental, les ontologies doivent prendre en compte des caractéristiques spatio-temporelles de ces données. Vu le besoin d'une modélisation des concepts et des opérateurs spatiaux et temporaux, nous nous appuyons sur la solution de réutilisation des ontologies du temps et de l'espace. Ensuite, une approche d'intégration de données spatio-temporelles accompagnée d'un mécanisme de raisonnement sur leurs relations a été introduite. Enfin, les méthodes de fouille de données ont été adaptées aux données spatio-temporelles sémantiques pour découvrir de nouvelles connaissances à partir de la base de connaissances.

L'approche a ensuite été mise en application au sein du prototype Geminat qui a pour but d'aider à comprendre les pratiques agricoles et leurs relations avec la biodiversité dans la zone atelier Plaine et Val de Sèvre. De l'intégration de données, à l'analyse de connaissances, celui-ci offre les éléments nécessaires pour exploiter des données spatio-temporelles hétérogènes ainsi qu'en extraire de nouvelles connaissances.

Abstract

The need to collect long-term observations for research on environmental issues led to the establishment of "Zones Ateliers" by the CNRS. Thus, for several years, many databases of a spatio-temporal nature are collected by different teams of researchers. To facilitate transversal analysis of different observations, it is desirable to cross-reference information from these data sources. Nevertheless, these sources are constructed independently of each other, which raises problems of data heterogeneity in the analysis.

Therefore, this thesis proposes to study the potentialities of ontologies as both objects of modeling, inference, and interoperability. The aim is to provide experts in the field with a suitable method for exploiting heterogeneous data. Being applied in the environmental domain, ontologies must take into account the spatio-temporal characteristics of these data. As the need for modeling concepts and spatial and temporal operators, we rely on the solution of reusing the ontologies of time and space. Then, a spatial-temporal data integration approach with a reasoning mechanism on the relations of these data has been introduced. Finally, data mining methods have been adopted to spatio-temporal RDF data to discover new knowledge from the knowledge-base.

The approach was then applied within the Geminat prototype, which aims to help understand farming practices and their relationships with the biodiversity in the "zone atelier Plaine and Val de Sèvre". From data integration to knowledge analysis, it provides the necessary elements to exploit heterogeneous spatio-temporal data as well as to discover new knowledge.

Table des matières

1 Introduction **1**
- 1.1 Contexte . . . 2
- 1.2 Approches proposées . . . 3
- 1.3 Objectifs et proposition . . . 4
- 1.4 Plan du manuscrit . . . 4

I Positionnement scientifique **7**

2 Intégration de données **9**
- 2.1 Intégration de données . . . 11
  - 2.1.1 Hétérogénéité des données . . . 12
  - 2.1.2 Approches d'intégration de données . . . 13
- 2.2 Intégration sémantique de données . . . 18
  - 2.2.1 Ontologies . . . 18
  - 2.2.2 Intégration sémantique de données . . . 23
  - 2.2.3 Synthèse . . . 29

3 Web sémantique et Base de connaissances **31**
- 3.1 Web sémantique . . . 33
  - 3.1.1 Modélisation de connaissances . . . 34
  - 3.1.2 Interrogation . . . 41
  - 3.1.3 Inférence . . . 43
- 3.2 Base de connaissances . . . 46
  - 3.2.1 Peuplement de l'ontologie . . . 47
  - 3.2.2 Stockage de données géospatiales . . . 52
  - 3.2.3 Interrogation de données géospatiales . . . 54

4 Modélisation et raisonnement spatio-temporel **59**
- 4.1 Le temps . . . 62
  - 4.1.1 Représentation du temps . . . 62
  - 4.1.2 Représentation du changement . . . 62
  - 4.1.3 Raisonnement temporel . . . 64
  - 4.1.4 Synthèse . . . 66

4.2 Modélisation et raisonnement temporel dans le Web sémantique . 66
4.2.1 Ontologies du temps . . . 67
4.2.2 Représentation des changements . . . 68
4.2.3 Raisonnement temporel . . . 72
4.2.4 Synthèse . . . 72
4.3 L'espace . . . 73
4.3.1 Représentation de l'espace . . . 73
4.3.2 Modèles de représentation . . . 74
4.3.3 Modèles de données spatiales . . . 76
4.3.4 Raisonnement spatial . . . 80
4.3.5 Synthèse . . . 84
4.4 Modélisation et raisonnement spatial dans le Web sémantique . . 85
4.4.1 Ontologies de l'espace . . . 85
4.4.2 Raisonnement spatial . . . 89
4.4.3 Synthèse . . . 93
4.4.4 Modélisation et raisonnement spatio-temporels dans le Web sémantique . . . 93

5 Fouille de donnés sémantiques 99
5.1 Extraction de connaissances à partir de données . . . 101
5.1.1 Présentation . . . 101
5.1.2 Algorithmes de fouille de données . . . 103
5.2 Fouille de donnés sémantiques . . . 107
5.2.1 Approches statistiques . . . 108
5.2.2 Rôles de l'ontologie dans l'extraction de connaissances . . 109
5.2.3 Synthèse . . . 110

II Proposition 113

6 Une ontologie pour l'environnement 115
6.1 Contexte d'application . . . 117
6.1.1 Zone atelier Plaine et Val de Sèvre . . . 117
6.1.2 Corpus de données . . . 118
6.1.3 Expression de besoins d'analyse . . . 122
6.2 Une ontologie pour l'environnement . . . 123
6.2.1 Représentation de l'information spatio-temporelle . . . 124
6.2.2 Réutilisation des ontologies du temps et de l'espace . . . 125
6.2.3 Conception et développement des ontologies locales . . . 128
6.2.4 Une ontologie pour l'environnement . . . 130
6.2.5 Études de cas . . . 134

7 Analyse de données environnementales hétérogènes 139
7.1 Intégration de données . . . 141
7.2 Intégration de données par un médiateur . . . 142
7.3 Intégration de données par l'entrepôt . . . 144
7.3.1 Architecture du système . . . 145
7.3.2 Construction de la base de connaissances . . . 147
7.3.3 Applications . . . 148
7.4 Fouille de données spatio-temporelles sémantiques . . . 152
7.4.1 Architecture du système . . . 153
7.4.2 Applications . . . 156

III Conclusions et perspectives 161

8 Conclusion et perspectives 163
8.1 Bilan . . . 164
8.2 Perspectives . . . 166

IV Annexes 171

A Intégration sémantique de données 173
A.1 Construction des ontologies . . . 174
A.2 Enregistrement des ontologies . . . 174
A.3 Traduction des données . . . 175
A.4 Chargement de données . . . 177

B Prototype et exemples d'analyse 179
B.1 Présentation du prototype . . . 179
B.2 Exemples d'analyse . . . 180
B.2.1 Rotation de cultures en deux années . . . 180
B.2.2 Croisement entre l'assolement et les observations . . . 181
B.2.3 Croisement entre les observations . . . 184
B.2.4 Détection des événements territoriaux . . . 185

C Règles temporelles 187
C.1 time:intervalBefore et time:intervalAfter . . . 187
C.2 time:intervalContains et time:intervalDuring . . . 187
C.3 time:intervalFinishes et time:intervalFinishedBy . . . 188
C.4 time:intervalMeets et time:intervalMetBy . . . 189
C.5 time:intervalOverlaps et time:intervalOverlappedBy . . . 189
C.6 time:intervalStarts et time:intervalStartedBy . . . 190
C.7 time:intervalEquals . . . 190
C.8 time:inside . . . 191

Table des figures **193**

Bibliographie **201**

Table des matières

Chapitre 1

Introduction

Sommaire

1.1 Contexte 2
1.2 Approches proposées 3
1.3 Objectifs et proposition 4
1.4 Plan du manuscrit 4

1.1 Contexte

La recherche sur les écosystèmes à long terme (LTER [1]) est une composante essentielle des efforts mondiaux visant à mieux comprendre les écosystèmes et l'environnement dont nous dépendons. Grâce à l'observation à long terme de sites représentatifs à travers le monde, LTER améliore notre compréhension de la structure et des fonctions des écosystèmes, qui fournissent des services essentiels aux hommes.

Participant au mouvement de LTER Europe, le CNRS a développé des zones ateliers [2] (ZA) qui forment un vaste réseau inter-organismes de recherches interdisciplinaires sur l'environnement et les anthropo-écosystèmes en relation avec les questions sociétales d'intérêt national. Les ZA se focalisent autour d'une unité fonctionnelle et y développent une démarche scientifique spécifique en s'appuyant sur des observations et expérimentations sur des sites ateliers, pour y mener des recherches pluridisciplinaires sur le long terme.

Chaque ZA fédère plusieurs laboratoires et équipes de chercheurs qui relèvent leurs propres bases de données pour leurs études. Par exemple, la ZA Alpes [3] possèdent 27 jeux de données, quant à la ZA Val et Plane de Sèvre [4], 6 bases de données principales sont déclarées, concernant l'assolement, la diversité végétale, l'avifaune, les insectes et les micromammifères, ou encore pour la ZA Bassin du Rhône [5], plus de 700 jeux de données sont disponibles en ligne. Les données concernent notamment des relevées ou des observations sur le terrain qui se composent en général d'informations temporelles, spatiales et thématiques.

Il existe conséquemment un fort besoin de partager ou de croiser les sources de données disponibles afin d'analyser, de vérifier ou de découvrir des relations entre elles. Par exemple, l'apparition ou la disparation d'une espèce vivant sur une zone agricole, ces informations sont fournies par deux sources différentes. Dès lors, on s'appuie sur la procédure d'intégration de données. Or, dans ce contexte, les systèmes d'informations, conçus et développés par des équipes différentes, constituent généralement des sources de données autonomes et hétérogènes. De ce fait, l'interopérabilité entre ces systèmes devient complexe puisque les applications doivent être adaptées afin d'interagir avec eux. En effet, pour chaque requête, elles doivent déterminer tout d'abord les sources de données pertinentes, la syntaxe et la terminologie requises pour l'interrogation et ensuite combiner les fragments de résultats issus de chaque source pour construire le résultat final. Ce processus d'adaptation peut être plus ou moins complexe: exploitation des résultats d'une source pour interroger une autre, élimination des redondances, etc..

1. https://www.ilternet.edu
2. http://www.za-inee.org
3. http://leca-bdgis.ujf-grenoble.fr
4. http://www.za.plainevalsevre.cnrs.fr
5. http://www.graie.org/zabr/index.htm

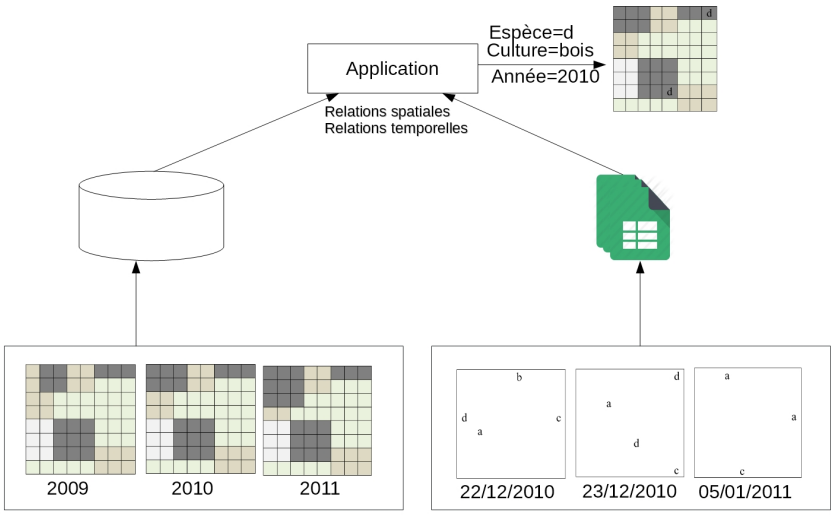


Figure 1.1 – Intégration de deux sources avec un raisonnement spatio-temporel.

De plus, les processus d'intégration de données environnementales sont souvent complexes car les applications doivent prendre en compte les relations spatio-temporelles. Effectivement, ces relations sont nécessaires pour le croisement de données provenant de différentes sources. Par exemple, on peut vérifier si une espèce a été observée sur un terrain à une période donnée (Figure 1.1), cela implique la prise en compte de la relation spatiale entre le point d'observation et la géométrie du terrain, et la relation temporelle entre les deux observations. Autrement dit, un raisonnement spatio-temporel est nécessaire pour répondre à une telle requête. Le problème posé devient alors d'intégrer de données hétérogènes avec la capacité de raisonnement spatio-temporel.

1.2 Approches proposées

Pour valoriser des données environnementales hétérogènes, nous nous appuyons sur les approches suivantes:

— **Intégration sémantique de données** : L'utilisation de modèles ontologiques prenant en compte les dimensions spatiale et temporelle favorise l'intégration et l'analyse de données environnementales hétérogènes. En effet, les ontologies permettent une représentation de connaissances qui satisfait aux exigences de la modélisation spatio-temporelle. Elles donnent aussi un moyen de raisonner sur les informations de contexte via des règles et des moteurs d'inférences. D'autre part, elles peuvent être utilisées comme un support pour l'intégration de données hétérogènes. Afin de concrétiser ces propositions, on se tourne vers les technologies du Web sémantique.

— **Fouille de données sémantiques** : L'application des méthodes de fouille

aux données intégrées, appelées alors données spatio-temporelles sémantiques, favorise la découverte de nouvelles connaissances. Ces méthodes peuvent être adaptées pour découvrir de nouvelles connaissances à partir d'une base de connaissances spatio-temporelle. Dans ce contexte, les ontologies et les technologies du Web sémantique sont utilisées comme un support pour les processus d'extraction de connaissances.

1.3 Objectifs et proposition

L'objectif principal de ce travail est de proposer une approche permettant l'intégration et l'exploitation de données environnementale hétérogènes. Plus précisément, les objectifs sont de :

— Proposer une modélisation des données environnementales, par l'approche ontologique, en prenant en compte les concepts et les relations spatiaux et temporels. Cela permet d'offrir un moyen d'intégrer des sources de données hétérogènes et de les exploiter à l'aide du raisonnement spatio-temporel. Pour cela, ce travail se propose d'étudier la modélisation spatiale et temporelle, en particulier celle basée sur les ontologies. Dès lors, les ontologies du temps et de l'espace doivent être examinées afin d'en choisir pour la réutilisation.

— Concevoir un système intégrant cette modélisation afin d'intégrer et d'exploiter des données spatio-temporelles hétérogènes. À cet égard, les technologies du Web sémantiques sont étudiées afin de proposer une méthode pour la construction et la gestion de la base de connaissances dont les concepts et l relations sont modélisés par des ontologies. Un mécanisme de raisonnement spatio-temporel doit être proposé.

— Incorporer dans le système les méthodes de fouille de données permettant l'analyse automatique des données et la découverte de nouvelles connaissances. Un prototype doit être conçu et développé pour mettre en synergie, les technologies du Web sémantique et les processus d'extraction de connaissances. Pour cela, de différentes approches visant à adapter des méthodes de fouilles aux données sémantiques doivent être examinées.

1.4 Plan du manuscrit

Ce manuscrit est organisé en trois parties. La première partie fixe le contexte et le positionnement scientifique relatifs aux problématiques de notre travail en rappelant brièvement de nombreux sujets, plus particulièrement, les ontologies et les technologies du Web sémantique ; la modélisation spatio-temporelle et enfin l'extraction de connaissances. La deuxième partie présente notre contribution. Il

se termine par une conclusion et des perspectives. Nous détaillons ci-dessous le contenu de chacun de ces chapitres.

Première partie

Chapitre 2 Ce chapitre est consacré à une présentation de l'intégration de données en général et l'intégration sémantique de données en particulier. Nous présentons les ontologies, leurs notions, leurs concepts et leurs rôles en soulignant leur application en tant que format d'échange pour faciliter l'interopérabilité entre diverses sources de données.

Chapitre 3 Ce chapitre se concentre sur deux sujets : les technologies du Web sémantique ainsi que la construction et la gestion de la base de connaissances.

Chapitre 4 Dans ce chapitre, nous présentons quelques travaux importants autour de la modélisation et du raisonnement sur des données spatiales et temporelles en utilisant des approches classiques et le Web sémantique.

Chapitre 5 Ce chapitre présente les processus d'extraction de connaissances à partir de données en se concentrant sur l'adaptation de ces processus aux données sémantiques.

Deuxième partie

Chapitre 6 Ce chapitre présente notre contexte d'application, la zone atelier Plaine et Val de Sèvre. Nous détaillons les sources de données spatio-temporelles disponibles et les besoins d'intégration et d'analyse de ces données. À partir de ce cas d'étude, nous présentons une modélisation de données environnementales par l'approche ontologique prenant en compte leur composante temporelle et spatiale. Pour cela, les ontologies du temps et de l'espace sont réutilisées.

Chapitre 7 Dans ce chapitre, nous présentons deux prototypes d'intégration et d'analyse de données spatio-temporelles hétérogènes. Chacun correspond à une approche d'intégration sémantique de données différente et possède donc ses propres avantages et inconvénients. Ensuite, l'incorporation des méthodes de fouille de données a été réalisée. À l'aide de celle-ci, nous pouvons montrer le rôle de l'extraction de données et du Web sémantique et leur relation dans la découverte de nouvelles connaissances à partir d'une base de connaissances.

Troisième partie

Enfin, nous concluons ce document en revenant sur les différents travaux présentés et la manière dont ils répondent aux objectifs initiaux. Nous envisagerons également certains travaux futurs qu'il nous semble important de garder à l'esprit.

Première partie

Positionnement scientifique

Chapitre 2

Intégration de données

Sommaire

2.1 Intégration de données 11
2.1.1 Hétérogénéité des données 12
2.1.2 Approches d'intégration de données 13
2.1.2.1 Approche médiateur 14
2.1.2.2 Approche matérialisée 16
2.1.2.3 Approche hybride 18
2.2 Intégration sémantique de données 18
2.2.1 Ontologies 18
2.2.1.1 Les composantes d'une ontologie 20
2.2.1.2 Typologie des ontologies 20
2.2.1.3 Le rôle des ontologies 22
2.2.2 Intégration sémantique de données 23
2.2.2.1 Construction des ontologies 24
2.2.2.2 Médiation des ontologies 27
2.2.3 Synthèse 29

Introduction du chapitre

L'intégration de données implique de combiner des données résidant dans différentes sources et de fournir aux utilisateurs une vue unifiée de ces données. Ce processus devient significatif dans une variété de situations qui inclut à la fois des domaines commerciaux et scientifiques, par exemple la combinaison des résultats de recherche de différentes sources. L'intégration de données apparaît avec une fréquence croissante à mesure que le volume et la nécessité de partager les données explosent. L'intégration de données sémantiques est le processus d'utilisation d'une représentation conceptuelle de données et de leurs relations pour éliminer les hétérogénéités possibles. Au cœur de cette intégration se trouve le concept d'ontologie [Cruz et Xiao, 2005] qui est un élément de la pile du Web sémantique (Figure 3.1).

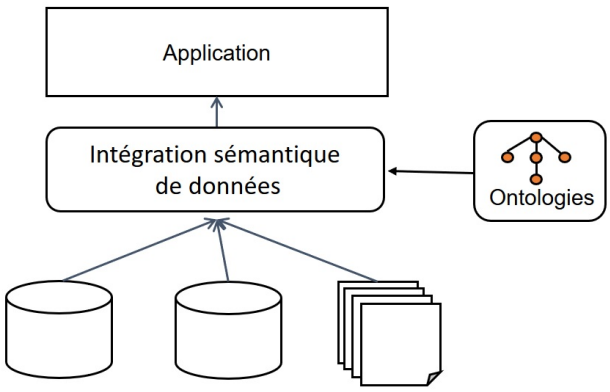


Figure 2.1 – Intégration sémantique de données.

Ce chapitre commence par la présentation de la notion d'ontologie et des principaux concepts d'intérêt. Parmi les rôles de l'ontologie, nous nous intéressons à son rôle dans le processus d'intégration de données hétérogènes. Pour ces raisons, nous décrivons ensuite différents types d'hétérogénéité de données, en nous concentrant sur l'hétérogénéité sémantique. Enfin, nous terminons par une étude sur les approches d'intégration de données hétérogènes à l'aide des ontologies.

2.1 Intégration de données

Le développement très rapide des données disponibles localement ou en ligne augmente la difficulté de retrouver les informations pertinentes. En effet, les informations peuvent être représentées et stockées dans une multitude de sources de données. Or, celles-ci sont souvent hétérogènes car elles sont conçues de manière indépendante par différents concepteurs. L'intégration de données est un processus ayant pour objectif de combiner d'abord les données résidant dans différentes sources en éliminant les conflits entre elles et de fournir ensuite aux utilisateurs une vue unifiée de celles-ci. Ce processus devient significatif dans une variété de situations, qui incluent à la fois des domaines commerciaux et scientifiques. L'intégration de données apparaît avec une fréquence croissante à mesure que le volume et la nécessité de partager les données existantes explosent.

Un système d'information est homogène si le logiciel qui gère les données est le même sur tous les sites, les données ont les mêmes format et structure et appartiennent à un même univers de discours. Un système hétérogène est celui qui n'adhère pas à toutes les caractéristiques d'un système homogène, c'est-à-dire, tout système d'information qui utilise des langages de programmation et d'interrogation, des modèles, des SGBD différents [Solar et Doucet, 2002].

Les hétérogénéités dans les systèmes d'information peuvent survenir à deux niveaux : systèmes et données (information) (Figure 2.2). Les hétérogénéités de systèmes sont dues aux différences de technologies utilisées, par exemple les différences de matériel, de logiciel ou de systèmes de communication. Les hétérogénéités de données se composent de l'hétérogénéité sémantique, l'hétérogénéité structurelle et l'hétérogénéité syntaxique qui surviennent lors de la représentation et de la modélisation des données [Sheth, 1999].

Ainsi, il existe deux types d'interopérabilité :

— **Interopérabilité de systèmes** : Elle vise à définir les normes, les standards et les protocoles afin de résoudre l'hétérogénéité entre les sources d'information.

— **Interopérabilité de données** : Elle vise à donner l'accès unifié à des ressources distribuées.

Dans le cadre de cette thèse, nous nous intéressons à l'interopérabilité sémantique, plus particulièrement, à l'application des ontologies dans le processus de l'intégration de données hétérogènes. Pour ces raisons, nous décrivons dans la suite les approches d'intégration de données en soulignant les approches à base ontologique. Pourtant, nous commençons par l'introduction de différents types d'hétérogénéité de données.

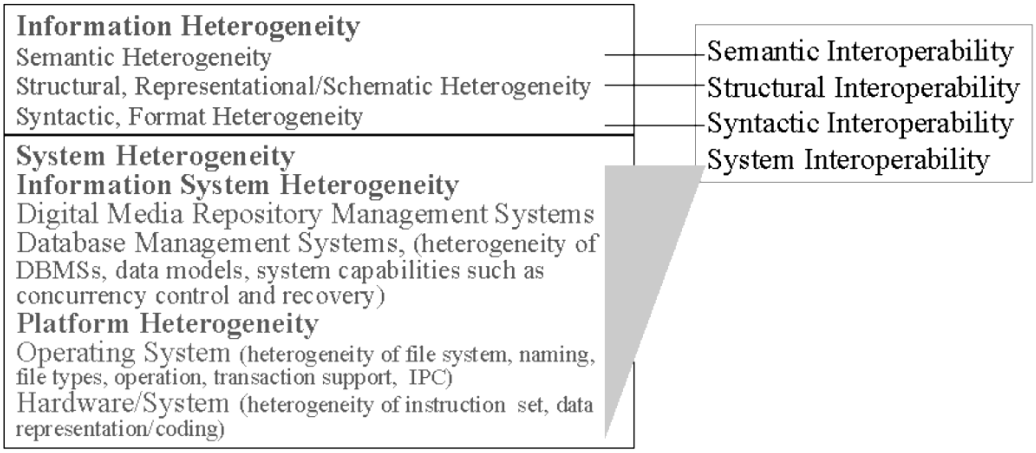


Figure 2.2 – Les hétérogénéités survenues dans un système d'information [Sheth, 1999].

2.1.1 Hétérogénéité des données

L'hétérogénéité des données se réfère à la façon dont les données sont organisées ou interprétées dans les systèmes. Elle concerne les différences dans le modèle de données, le langage de manipulation de données, le mécanisme de contrôle de la concurrence... De toute évidence, l'autonomie de la conception constitue un obstacle majeur à la réalisation de l'interopérabilité sémantique, ou à l'échange d'information entre différents systèmes. Nous pouvons distinguer trois types d'hétérogénéité de données :

— **Hétérogénéité syntaxique** : L'hétérogénéité syntaxique est causée par les différences entre plateformes logicielles, et les formats qu'elles manipulent. En effet, chaque sources de données adopte un modèle ou un schéma de données propre qui diffère d'une source à une autre. Elle se trouve dans les formats de stockage de données, dans les langages d'interrogation, dans les protocoles d'accès ou dans les interfaces. Par exemple, certaines sources de données sont accessibles à travers des formulaires d'interrogation alors que d'autres sont disponibles sous forme de fichiers plats.

— **Hétérogénéité structurelle (ou schématique)** [Kim et Seo, 1991] : L'hétérogénéité structurelle résulte d'une structuration ou d'une classification différente des informations. Elle survient lorsque des concepts équivalents sont représentés différemment dans les sources de données. Elle est associée au choix des noms, des types de données, des attributs, ou des unités pour construire le schéma des sources.

— **Hétérogénéité sémantique** [Sheth et Kashyap, 1993, Naiman et Ouksel, 1995, García-Solaco *et al.*, 1995] : Elle se réfère aux différences dans la signification, l'interprétation ou l'utilisation d'une même donnée entre les

domaines d'application. Le problème d'avoir des différences dans la compréhension par rapport à la même information vient parfois de la diversité des aspects géographiques et organisationnels qui les utilisent. Par exemple, la position d'un objet est mesurée différemment dans les systèmes si le système de référence n'est pas le même.

Hors ces trois catégories d'hétérogénéités présentés ci-dessus, il existe aussi l'hétérogénéité intentionnelle [Goh, 1997] qui se réfère aux différences dans le contenu informationnel des sources de données.

Les chercheurs et les développeurs ont travaillé pour résoudre ces hétérogénéités pendant plusieurs années. Depuis les années 1990, trois générations d'interopérabilité ont vu le jour :

— **Première génération** : Avant 1990, il y avait des progrès significatifs dans la gestion de l'hétérogénéité et le support de l'interopérabilité ou de l'intégration dans les environnements avec des bases de données structurées et des SGBD traditionnels. Les recherches de cette génération se concentraient sur les architectures de bases de données multiples ou fédérées pour traiter l'hétérogénéité associée aux modèles de données ou aux problèmes de schéma.

— **Deuxième génération** : Au cours des années 1990, les recherches ont notamment favorisé l'interopérabilité syntaxique et structurelle. Elles ont permis de traiter les problèmes au niveau des données avec l'apparition des architectures de médiation de sources de données.

— **Troisième génération** : À partir des années 2000, l'objectif principal des études est l'interopérabilité sémantique basée sur les approches logiques, les ontologies et les transformations de modèles.

2.1.2 Approches d'intégration de données

Un système d'intégration est défini comme un système donnant l'impression d'interroger plusieurs sources distribuées et hétérogènes comme une seule. Pour cela, les données de différentes sources sont intégrées dans un schéma global qui répond aux besoins des utilisateurs. Toutes les hétérogénéités sont cachées pour les utilisateurs, ceux-ci interrogent le schéma global comme un simple schéma de base de données.

Un système d'intégration est composé en général de trois couches principales comme illustré par la Figure 2.3 :

— **Couche des données** : Elle contient l'ensemble des sources de données à intégrer.

— **Couche des adaptateurs ou des chargeurs** : Elle permet d'accéder aux sources de données, d'en extraire les données et de représenter ces données dans le schéma global. C'est aussi le moyen avec lequel la source peut interagir avec les autres composants de l'architecture.

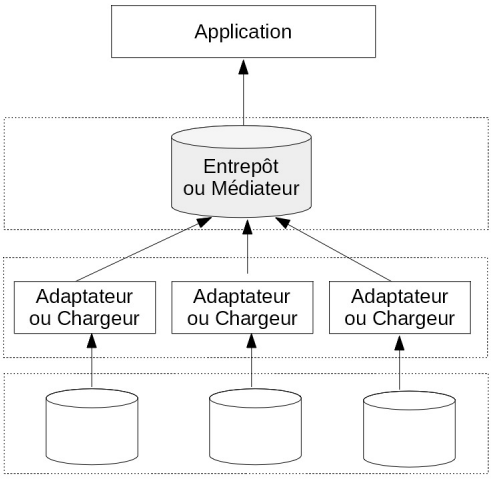


Figure 2.3 – Système d'intégration de données.

— **Couche entrepôt ou médiateur** : Elle contient les éléments nécessaires permettant d'interroger les différentes sources à travers le schéma global. Celui-ci peut être virtuel ou matérialisé dans un entrepôt de données.

Il existe trois approches pour proposer la vue unifiée, présenté ci-dessous, les approches médiateurs, les approches entrepôts de données et les approches hybrides.

2.1.2.1 Approche médiateur

Le schéma global d'un système d'intégration de données peut être virtuel. Dans ce cas, toutes les données demeurent dans les sources locales et on y accède par une infrastructure intermédiaire, généralement appelée médiateur, contenant le schéma global. L'objectif de cette approche est de donner à l'utilisateur l'illusion d'interroger un système homogène et centralisé alors que les sources interrogées sont réparties, autonomes et hétérogènes. L'intégration de données est fondée sur :

— La définition du schéma global unifiant les schémas de chaque source hétérogène à intégrer.

— La description homogène et abstraite du contenu de ces sources par des vues.

L'architecture médiateur a été proposée par [Wiederhold, 1992]. Elle est composée de trois couches (Figure 2.4) : médiateur, adaptateurs et sources.

Le médiateur est chargé de la localisation des sources de données par rapport à une requête à l'aide des connaissances sur ces sources. Au niveau du médiateur, le schéma global fournit un vocabulaire pour l'expression des requêtes et la description des sources par un ensemble de vues abstraites sur ces dernières.

Étant un modèle du domaine d'application, l'ontologie peut être utilisée en tant que schéma global. En effet, elle fournit un vocabulaire structuré servant de support à l'expression des requêtes. Par ailleurs, elle établit une connexion entre les sources accessibles en décrivant de façon homogène et uniforme leur contenu.

Les adaptateurs, appelés aussi «wrappers», sont des outils permettant aux médiateurs d'accéder au contenu des sources en un langage uniforme. Ils décomposent ou reformulent (réécrivent) une requête en langage de requêtes spécifique accepté par chaque source. Ensuite ils recomposent les réponses partielles en une seule réponse conforme au schéma global du médiateur.

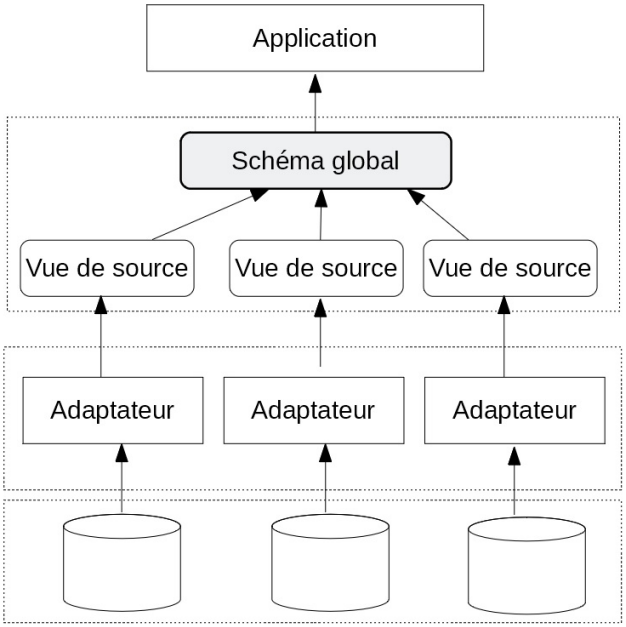


Figure 2.4 – Architecture d'un système médiateur

L'avantage des approches médiateur est la facilité de la mise en œuvre :

— En effet, il n'existe pas de coût d'actualisation des données car elles sont systématiquement à jour.

— De plus, ces approches ne nécessitent pas d'important investissement en termes de matériel, de déploiement et d'administration.

Pourtant, plusieurs inconvénients subsistent comme :

— Difficulté dans la conception des adaptateurs.

— Modification ou nettoyage de données.

— Performances limitées par celles des sources et du réseau.

— Risque d'incomplétude du résultat causée par l'indisponibilité d'une certaine source au moment de l'interrogation.

— Difficulté dans l'établissement des contraintes et des jointures entre les sources [Davidson *et al.*, 1995].

— Manque de contrôle sur l'historique et les versions des données.

— Interruption de l'accès aux données à cause de l'évolution d'une source de données.

2.1.2.2 Approche matérialisée

Dans cette approche, le schéma global d'un système d'intégration de données est entièrement matérialisé. Comme la conception d'un tel système est similaire à celle des entrepôts de données, l'approche est aussi appelée approche entrepôt. En effet, il est nécessaire qu'une nouvelle base, c'est-à-dire un entrepôt, soit développée via un SGBD avec une reproduction de toutes les données sources correspondant au schéma global. Dans ce cas, les requêtes envers le schéma global de l'entrepôt s'appuient sur les techniques d'interrogation classiques du domaine des bases de données. Des lors, l'utilisateur interagit avec l'entrepôt par des requêtes d'interrogation directes aux données.

L'intégration de données est alors fondée sur le schéma global de l'entrepôt fournissant une vue intégrée des sources. Pour être exploitable, les données sont extraites à partir de sources hétérogènes, ensuite transformées au format de représentation des données de l'entrepôt par des extracteurs, et enfin stockées dans l'entrepôt. Elles sont éventuellement filtrées pour éliminer les données peu pertinentes. Ainsi, le processus d'intégration de données se décompose en quatre étapes principales [Hacid et Reynaud, 2004] qui correspondent à celles de l'approche d'ETL (Extract, Transform and Load) ci-dessous (Figure 2.5) :

1. Extraction de données à partir des sources.

2. Transformation de données au niveau structurel et sémantique.

3. Intégration de données.

4. Stockage de données intégrées dans le système cible.

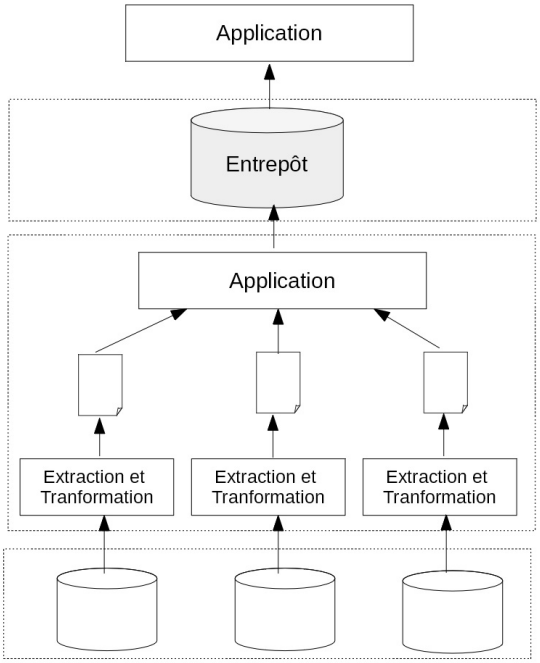


Figure 2.5 – Processus de construction d'un entrepôt de données d'après [Hacid et Reynaud, 2004].

Dans la pratique, un seul outil, tel qu'un adaptateur ou un outil de migration de données peut être utilisé dans les deux premières étapes. Dans certain cas, par exemple au sein d'un système multibases, toutes les étapes de traitement peuvent aussi être groupées dans un seul logiciel. Le rôle de l'adaptateur est d'extraire les données et de les transformer en un même format. Chaque adaptateur fournit ainsi une interface d'accès et des requêtes à la source. Les données extraites sont ensuite intégrées en éliminant les conflits entre elles puis stockées dans l'entrepôt. Lorsque les étapes d'extraction et d'intégration sont réalisées séparément, les données extraites sont stockées de manière temporaire en utilisant un média par source ou un média pour toutes les sources.

L'avantage des approches matérialisées est la réactivité et les performances supérieures du système.

— Comme les données sont regroupées dans une seule source semblable à celle d'une base de données simple, le traitement des requêtes se fait d'une façon rapide et efficace grâce au langage d'interrogation puissant et aux techniques d'optimisation de requêtes. En outre, la phase de réécriture de requêtes n'est plus nécessaire.

— Cette approche évite tout problème lié à la communication à distance et aux caractéristiques des sources locales. Les performances du système ne sont plus dépendantes de celles de chaque source ou des délais de communication. De plus, en cas d'incident, l'indisponibilité d'une source n'implique

pas le résultat des requêtes.

Néanmoins, elles possèdent des inconvénients majeurs comme :

— Nécessité d'un support de stockage volumineux et fiable et des outils ETL spécifiques pour le traitement préalable des données [Laïla et Dalila, 2006].

— Coût de maintenance causé par les opérations de mises à jour au niveau des sources de données. Le problème de maintenance du système est similaire au problème de maintenance des vues matérialisées [Chen *et al.*, 2004].

2.1.2.3 Approche hybride

Les approches hybrides sont également considérées comme une troisième solution possible pour l'intégration de données. Dans ces approches, la vue globale est divisée en parties matérialisées et virtuelles, autrement dit, certains objets ou relations sont choisis pour la matérialisation alors que d'autres résident dans les sources locales et seront extraits au moment de la requête. De cette façon, d'une part, ces approches fournissent un accès rapide aux données matérialisées qui sont rarement modifiées. D'autres part, les données qui sont mises à jour souvent sont interrogées directement depuis les sources si nécessaire.

Les critères considérés par ces approches sont la fréquence des requêtes, le temps de réponse à ces requêtes et la fréquence de mise à jour des données. D'autres paramètres peuvent être considérés dans un scénario d'intégration comme la puissance d'interrogation des sources, leur disponibilité, ou l'importance de leurs données.

2.2 Intégration sémantique de données

Quelle que soit l'approche d'intégration utilisée, une ontologie peut être utilisée en tant que schéma global. En effet, l'ontologie permet l'interopérabilité entre diverses sources de données hétérogènes en fournissant un vocabulaire structuré décrivant de façon homogène et uniforme leur contenu. L'utilisation des ontologies est aussi l'objectif des études des systèmes d'intégration de données de troisième génération. Dans la suite, nous présentons les approches à base ontologique visant à intégrer des sources hétérogènes.

2.2.1 Ontologies

Le terme «Ontologie» est emprunté à la philosophie, dans laquelle, il est considéré comme une branche de la métaphysique qui s'intéresse à la nature et à l'organisation de la réalité. Il signifie:« la science de ce qui existe: on ne cherche pas à expliquer le monde, mais à le construire sur l'empirie et la logique » [Christophe, 2005].
Dans les années 1980, l'ontologie est apparu dans le domaine de l'intelligence

artificielle pour résoudre les problèmes de modélisation des connaissances et, plus précisément, en ingénierie des connaissances. Dans ce domaine, Neches et ses collègues ont défini une ontologie comme suit : « Une ontologie définit les termes de base et les relations comprenant le vocabulaire d'une zone thématique ainsi que les règles pour combiner des termes et des relations pour définir des extensions au vocabulaire » [Neches *et al.*, 1991].

Cette définition explique comment procéder pour construire une ontologie en donnant des lignes directrices : identifier les termes de base et les relations entre eux, identifier les règles en vue de combiner ces termes, fournir des définitions de ces termes et ces relations. Selon cette définition, une ontologie inclut non seulement les termes qui y sont explicitement définis, mais aussi des termes qui peuvent être déduits à l'aide de règles. Plus tard, la définition de Gruber devient la plus couramment citée, considérant une ontologie comme : « une spécification explicite d'une conceptualisation » [Gruber, 1993].

Sur la base de la définition de Gruber, de nombreuses définitions des ontologies ont été proposées dans la littérature. On l'a légèrement modifiée en y ajoutant la notion de partage : « Une ontologie est un accord sur une conceptualisation partagée et éventuellement partielle. » [Guarino et Giaretta, 1995], ou « une spécification formelle d'une conceptualisation partagée » [Borst, 1997], ou encore « une spécification formelle et explicite d'une conceptualisation partagée » [Studer *et al.*, 1998].

D'après nous, ces définitions s'appuient globalement sur quatre notions importantes :

1. **Formel** : L'ontologie est représentée sous une forme traitable par machine. Cette description permet de rendre automatique certains traitements.
2. **Explicit** : Les concepts utilisés sont définis de façon déclarative.
3. **Conceptualisation** : Une ontologie n'est qu'une abstraction du monde réel. Dès lors, les concepts ainsi que leurs relations utilisés doivent être décrits sans ambiguïté.
4. **Partage** : L'ensemble des membres d'une communauté se sont mis d'accord sur les concepts définis dans l'ontologie. Ceux-ci pourront donc être utilisés pour se comprendre.

Ces notions d'ontologies et de modèles de données diffèrent par deux aspects importants :

— Les ontologies se fondent sur une compréhension partagée au sein d'une communauté. Cette compréhension représente un accord sur les concepts et leurs relations existants dans un domaine.

— Les ontologies utilisent des représentations basées sur la logique et retraitable par machine. Elles permettent la manipulation informatique, y compris le transfert des ontologies entre des machines, le stockage d'ontologies,la vérification de la cohérence des ontologies, le raisonnement sur les ontologies, etc..

2.2.1.1 Les composantes d'une ontologie

L'ontologie permet la formalisation de connaissances générales d'un domaine et la représentation des individus en capturant la sémantique de ce domaine. Pour cela, elle définit les concepts et leurs relations, et les organise de manière hiérarchique. Ainsi, les connaissances intégrées dans une ontologie sont formalisées en s'appuyant principalement sur quatre types de composants : les concepts, les relations, les instances et les axiomes [Gruber, 1993] :

— **Concepts** : Ils représentent des ensembles ou des classes d'entités ou d'objets d'un domaine. Il existe deux types de concept : les concepts primitifs qui ne peuvent pas être définis en termes d'autres concepts ; et les concepts définis.

— **Relations** : Elles décrivent les interactions entre les concepts ou les propriétés d'un concept. Ces relations peuvent se diviser en deux catégories :

  — Relation taxonomique (ou hiérarchique) : Elle organise les concepts en structure arborescente. Les formes les plus courantes sont les relations de spécialisation et les relations partielles qui décrivent des concepts faisant partie d'autres concepts.

  — Relation associative (ou sémantique) : Comme la relation «partie-tout», ou parfois appelée «partie-de», elle est une relation hiérarchique qui existe entre un couple de concepts dont l'un dénote une partie et l'autre dénote le tout.

  Comme les concepts, les relations peuvent être organisées en taxonomies. Une fois que cette conceptualisation a été concrétisée, une ontologie a été produite.

— **Instances** : Ils sont des individus ou des objets. Ils peuvent inclure des individus concrets ou abstraits. Chaque individu correspond à une instance de la classe à laquelle il appartient. Si l'ontologie contient des instances, ou autrement-dit, si elle est peuplée, elle devient alors une base de connaissances.

— **Axiomes** : Ils sont des assertions présentées sous une forme logique qui constituent la théorie globale de l'ontologie. Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférence. En d'autres termes, les axiomes sont utilisés pour modéliser des phrases qui sont toujours vraies. Les types d'axiome peuvent être classés selon leur sens sémantique [Staab et Maedche, 2000].

2.2.1.2 Typologie des ontologies

Dans la littérature, les ontologies peuvent être classifiées selon plusieurs critères :

— Degré de formalisme de la représentation [Uschold et Gruninger, 1996]

— Objet de conceptualisation [Guarino, 1997]

— Quantité, type de structure de la conceptualisation et sujet de la conceptualisation [van Heijst *et al.*, 1997]

— Niveau de détail [Guarino, 1998]

— Niveau de complétude [Bachimont, 2000]

— Niveau d'expressivité [Mizoguchi, 2003]

Nous adoptons la classification de [Guarino, 1998] qui se compose de quatre niveaux dépendant du degré de conceptualisation (Figure 2.6) :

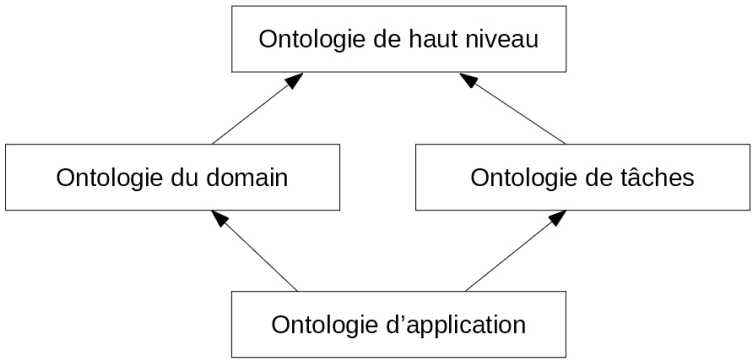


Figure 2.6 – Typologie des ontologies selon leur niveau de conceptualisation.

— **Ontologies de haut niveau** : Elles sont dédiées aux utilisations générales. En étudiant les catégories des choses qui existent dans le monde, comme les concepts de haut niveau d'abstraction, ces ontologies sont réutilisables d'un domaine à un autre et sont conçues pour réduire les incohérences des termes définis en aval de la hiérarchie.

— **Ontologies de domaine** : Elles sont spécialisées pour un certain type d'artefact et s'attachent à décrire le vocabulaire relatif à un domaine. Elles permettent de spécialiser les termes et les notions des ontologies de haut niveau.

— **Ontologies d'application** : Elles sont dédiées à un champ d'application précis à l'intérieur d'un domaine et décrivent le rôle particulier des entités de l'ontologie de domaine dans ce champ. Elles sont à la fois une union et une spécialisation des ontologies de tâches et de domaines [Maedche et Staab, 2001]. Ces ontologies précisent les connaissances nécessaires pour une application donnée en prenant en compte le vocabulaire spécialisé des experts. Généralement, les ontologies d'application ne sont pas réutilisables et possèdent donc un intérêt plus limité.

— **Ontologies de tâches** : Elles fournissent un vocabulaire systématique des termes utilisés afin de résoudre les problèmes associés aux tâches d'un domaine. Les ontologies de tâches caractérisent l'architecture computationnelle d'un système à base de connaissances qui réalise une tâche [Mizoguchi et Bourdeau, 2000].

2.2.1.3 Le rôle des ontologies

Les ontologies ont été employées dans divers domaines et pour différents objectifs. Leurs applications les plus répandues peuvent être classées en trois catégories [Gruber, 1993, Uschold et Gruninger, 1996] :

— **Communication** : Les ontologies peuvent être utilisées dans la communication entre les hommes et/ou les systèmes pour le partage et la compréhension dans des contextes particuliers et selon les besoins.

— **Interopérabilité** : Les ontologies facilitent la compréhension et l'interprétation des informations échangées, en se présentant comme un format d'échange.

— **Ingénierie des systèmes** : Les ontologies aident également au processus de construction et de maintenance des systèmes. En particulier, elles jouent un rôle important pour les trois aspects suivants :

  — Réutilisabilité : Les ontologies peuvent être réutilisables, adaptables ou partagées dans différents systèmes lorsqu'elles sont représentées dans un langage formel.

  — Fiabilité : Une représentation formelle des ontologies facilite la vérification automatique de leur cohérence.

  — Spécification : Les ontologies facilitent l'identification des besoins du système et la compréhension des liens et relations entre ses composants. Elles définissent la spécification déclarative du système.

Ainsi, dans un système informatique, les ontologies peuvent être utilisées à différents niveaux [Guarino, 1998] :

— Spécification et l'analyse des besoins du système.

— Maintenance du système en favorisant la documentation ou en permettant la vérification d'incohérences.

— Coopération et partage en tant que format d'échange.

— Recherche d'informations en servant de base d'index ou de métadonnées.

— Interopérabilité entre diverses sources de données hétérogènes.

— Compréhension du schéma conceptuel et du vocabulaire du système à travers sa visualisation.

— Exécution et traitement de requêtes exprimées en langue naturelle.

Dans ces travaux, nous nous intéressons à l'utilisation des ontologies comme un format d'échange en vue de faciliter l'interopérabilité entre diverses sources de données hétérogènes. Par conséquent, nous présentons dans la suite l'intégration de données par l'approche ontologique et les techniques appliquées dans ce processus.

2.2.2 Intégration sémantique de données

Nous appelons systèmes d'intégration sémantique, les systèmes utilisant les ontologies dans leur processus d'intégration. En effet, en raison de leurs potentiels de description de la sémantique des connaissances, les ontologies peuvent jouer plusieurs rôles dans le processus d'intégration. Les premières approches pour l'intégration sémantique s'appuyaient principalement sur l'utilisation de thésaurus à des fins de traduiction entre des vocabulaires spécifiques [Visser et Schuster, 2002]. Récemment, les ontologies sont utilisées pour gérer le problème de connaissances implicites et cachées en permettant la conceptualisation explicite d'un domaine. L'avantage de cette approche est qu'elle représente une norme qui est largement acceptée par la communauté de la recherche [Durbha et King, 2005].

L'ontologie est une spécification explicite d'une conceptualisation. Par conséquent, les ontologies peuvent être utilisées dans le processus d'intégration des données pour décrire la sémantique des sources d'information et rendre ainsi le contenu explicite. En ce qui concerne l'intégration des sources de données, elles peuvent être utilisées pour l'identification et l'association des concepts et les informations sémantiques correspondantes. [Uschold et Gruninger, 1996] ont également considéré l'interopérabilité comme une application clé des ontologies et de nombreuses approches ontologiques pour l'intégration de l'information.

[Cruz et Xiao, 2005] ont synthétisé quatre rôles principaux des ontologies dans le processus d'intégration de données :

— **Représentation des métadonnées (ou le schéma des sources)** : Les ontologies fournissent un vocabulaire qui permet de représenter de façon explicite chaque source de données en utilisant le même langage indépendamment de leur structure native.

— **Conceptualisation commune** : Avec leur capacité de représentation des connaissances, les ontologies peuvent être utilisées pour décrire le schéma global et fournir une conceptualisation suffisamment complète pour modéliser en un même formalisme des sources impliquées dans le processus d'intégration.

— **Support pour l'expression de requêtes de haut-niveau** : Le langage fourni par les ontologies est assez expressif pour exprimer des requêtes complexes sans besoin de connaissances explicites de la structure des données, ni de leurs sémantiques et encore moins de leur répartition. Des lors, les

requêtes sont exécutées de manière efficace grâce aux correspondances sémantiques établies entre l'ontologie globale et les ontologies locales.

— **Support pour les correspondances** : Les ontologies peuvent être utilisées pour établir des liens sémantiques entre les éléments appartenant aux différentes sources. Elles permettent de gérer des données incohérentes en fournissant un moyen pour résoudre automatiquement les conflits sémantiques.

L'intégration de données à base ontologique peut être divisée en trois étapes (Figure 2.7) :

1. **Construction des ontologies locales** : Cette étape a pour l'objectif de construire des ontologies locales à partir du schémas des sources. Ainsi, on interprète le sens de chaque source en l'associant à une ontologie locale. Ce processus est effectué d'une façon manuelle ou semi-automatique. En réalité, la sémantique de chaque source peut être découverte semi-automatiquement grâce aux techniques de fouilles de données. Pourtant, comme son résultat est approximatif, il nécessite donc la présence de l'expert pour valider ce résultat. Cette étape est négligeable si chaque source est déjà associée à une ontologie a priori.
2. **Intégration des ontologies** : L'intégration des ontologies consiste à établir les relations sémantiques (équivalence, subsomption) entre les éléments des ontologies.
3. **Intégration des données** : L'étape vise à peupler les données dans un entrepôt pour les systèmes matérialisés ou à construire des médiateurs pour les systèmes virtuels en s'appuyant sur les correspondances établies dans les étapes précédentes.

2.2.2.1 Construction des ontologies

Dans presque toutes les approches d'intégration sémantique, les ontologies sont utilisées pour la description explicite des sémantiques des sources [Wache *et al.*, 2001]. Il existe plusieurs méthodologies, en général, trois directions distinctes sont identifiées : les approches à ontologie unique, les approches à ontologies multiples et les approches hybrides.

Approches à ontologie unique

Ces approches utilisent une ontologie globale fournissant un vocabulaire commun pour la spécification de la sémantique (Figure 2.8). Toutes les sources d'information sont liées à l'ontologie globale. Ces approches peuvent être appliquées aux problèmes d'intégration où toutes les sources d'information fournissent à peu près le même point de vue sur un domaine.

Cette approche est naturelle lorsqu'on dispose d'une ontologie globale fournissant un vocabulaire partagé pour la spécification sémantique, ou lorsque

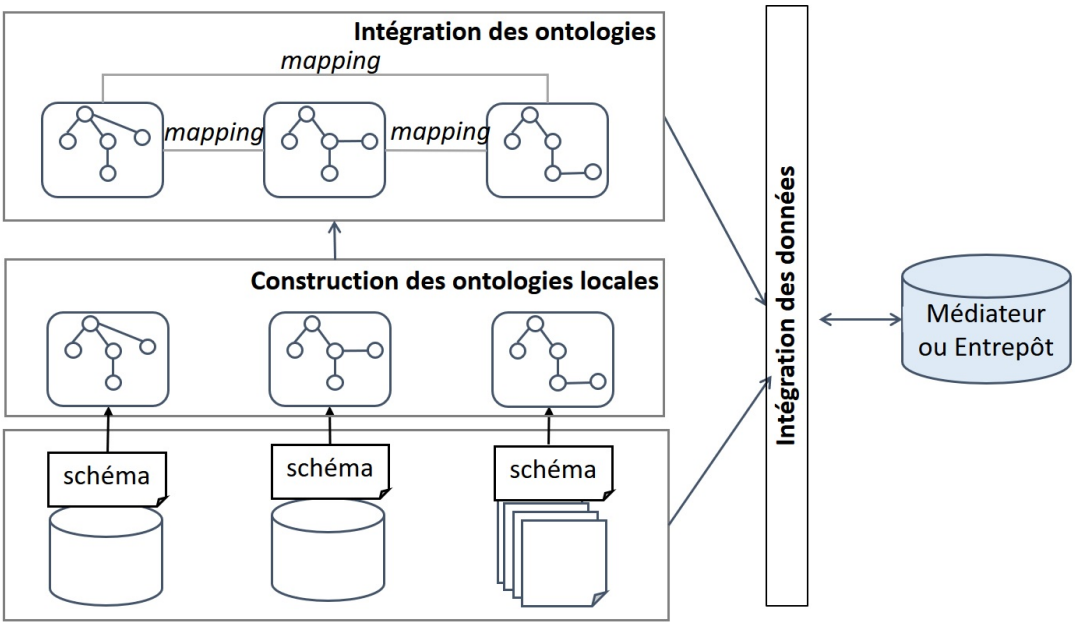


Figure 2.7 – Intégration sémantique de données.

plusieurs ontologies existent pour décrire un même domaine, avec une granularité proche, et sont réalisées dans une même optique. Elle est à éviter quand une des ontologies évolue de manière indépendante, car dans ce cas, on doit régulièrement modifier l'ontologie globale.

Approches à ontologies multiples

Dans ces approches, chaque source d'information est décrite par sa propre ontologie (Figure 2.9). L'avantage de cette approche est que chaque ontologie source peut être définie sans prendre en considération les autres sources ou les autres ontologies. Cette architecture peut simplifier les modifications dans une source ou l'ajout et la suppression des sources. Dans ce cas, il suffit d'ajouter ou de supprimer les correspondances entre les ontologies. Pourtant, le manque d'un vocabulaire commun conduit à une difficulté extrême pour comparer les ontologies locales. Afin de surmonter ce problème, un formalisme de représentation additionnel définissant le mapping inter-ontologies est requis. Ce dernier identifie sémantiquement les termes correspondant aux ontologies. Néanmoins, ce mapping est difficile à définir à cause de nombreux problèmes d'hétérogénéité sémantique qui peuvent se produire. Parmi les principales difficultés, il y a les cas de synonymie et d'homonymie mais surtout l'ambiguïté provoqué par le manque d'information.

Approches hybrides

Ces approches sont introduites pour surmonter les inconvénients des deux approches précédentes. Similaire à l'approche à ontologies multiples, dans

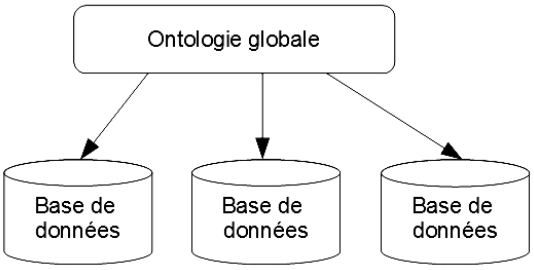


Figure 2.8 – Approches à ontologie unique.

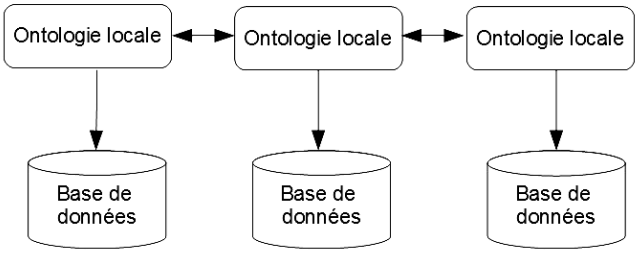


Figure 2.9 – Approches à ontologies multiples.

ces approches, la sémantique de chaque source est décrite par sa propre ontologie mais elle est construite grâce à un vocabulaire partagé afin d'être comparable entre elles (Figure 2.10). L'avantage de ces approches est que de nouvelles sources peuvent être facilement ajoutées avec moins de modifications des correspondances ou du vocabulaire partagé. Par conséquence, elles permettent l'acquisition et l'évolution des ontologies.

Les approches hybrides nécessitent cependant de commencer par créer un vocabulaire commun ainsi que les règles de combinaison des termes. De plus, si les sources sont indépendantes, les ontologies devront utiliser un langage commun. Dès lors, il est nécessaire d'avoir un consensus au préalable ou de construire une ontologie pour chaque source.

Dans ces approches, l'intégration des ontologies peut être réalisée a posteriori [Mitra *et al.*, 2000] ou a priori [Bellatreche *et al.*, 2004]. Dans le cas de l'intégration a posteriori, comme les ontologies locales sont indépendantes, la liaison entre une ontologie locale et l'ontologie partagée est effectuée d'une façon manuelle ou semi-automatique. Au contraire, l'intégration a priori est effectuée de manière automatique grâce aux relations sémantiques existantes entre les ontologies locales, celles-ci sont définies a priori dans l'ontologie globale qui est aussi l'ontologie du domaine [Nguyen, 2006].

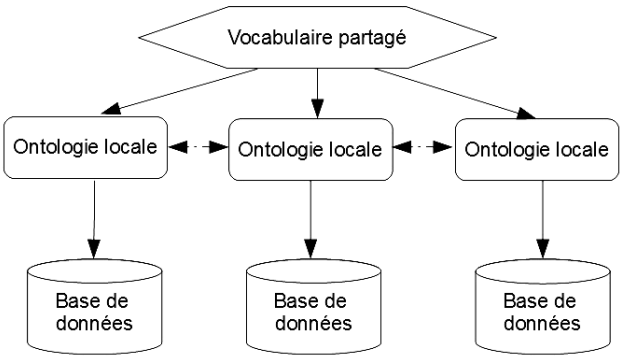


Figure 2.10 – Approches hybrides.

2.2.2.2 Médiation des ontologies

Les trois approches d'intégration des données hétérogènes ci-dessus s'appuient sur l'utilisation des ontologies décrivant des terminologies partagées pour permettre l'interopérabilité entre les sources de données. Dans la littérature, un certain nombre de techniques ont été proposées pour découvrir des correspondances entre les terminologies ou prendre en compte les différences entre les ontologies utilisées. Nous distinguons deux catégories principales : la correspondance d'ontologies («mapping d'ontologie») et la fusion d'ontologies [Bruijn *et al.*, 2006].

Mapping d'ontologies

Un *mapping* de deux ontologies est une spécification déclarative du chevauchement sémantique entre ces ontologies (Figure 2.11). C'est la sortie du processus de *mapping* qui spécifie une convergence sémantique entre différentes ontologies afin d'en extraire les correspondances entre certaines entités [Noy, 2004]. Celles-ci sont exprimées par des axiomes. Trois phases principales peuvent être distinguées dans ce processus :

1. Découverte des *mapping*.
2. Représentation des *mapping*.
3. Exploitation et exécution des *mapping*.

Alignement d'ontologies

La découverte (semi-) automatisée de ces correspondances est appelée alignement d'ontologies. En général, il est décrit comme une application de l'opérateur *Match* [Rahm et Bernstein, 2001] dont l'entrée est constituée d'un ensemble d'ontologies et la sortie, formée des spécifications des correspondances entre ces ontologies. L'alignement d'ontologie est réalisé lorsque les sources deviennent cohérentes les unes avec les autres, mais elles sont maintenues séparées ou quand il existe des domaines complémentaires [Choi *et al.*, 2006].

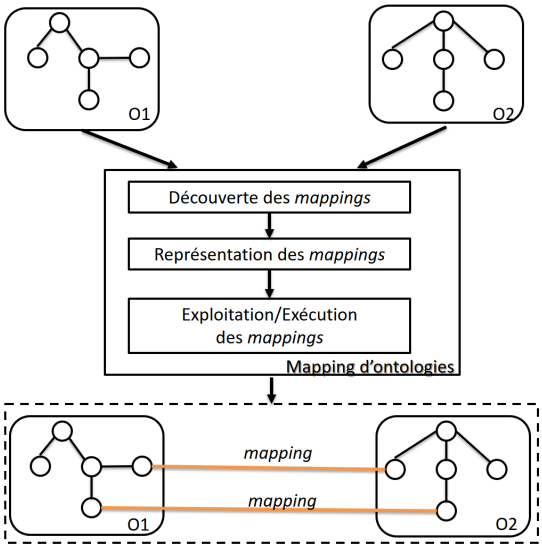


Figure 2.11 – *Mapping* de deux ontologies.

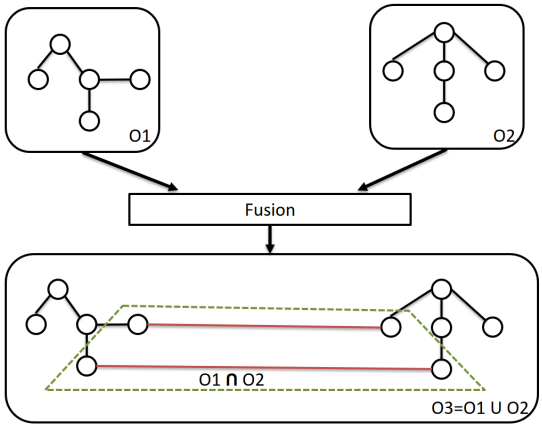


Figure 2.12 – Fusion d'ontologies.

Fusion d'ontologies

Les techniques de fusion d'ontologies permettent de créer une nouvelle ontologie, appelée ontologie fusionnée, capturant les connaissances décrites par les ontologies d'origine. L'ontologie fusionnée unifie et remplace les ontologies sources. Dès lors, il est nécessaire d'assurer que toutes les correspondances et différences entre ces ontologies soient correctement prises en compte dans l'ontologie résultante.

Deux techniques peuvent être appliquées (Figure 2.12) :

— **Union d'ontologies** : Dans l'union d'ontologies, l'ontologie résultante contient l'union des entités provenant des ontologies d'origine et résout les différences de représentation d'un même concept.

	Approches à ontologie unique	**Approches à ontologies multiples**	**Approches hybrides**
Effort d'implémentation	Simple	Élevé	Raisonnable
Ajout, enlèvement des sources	Besoin d'adaptation de l'ontologie globale	Ajouter une autre ontologie locale, mettre à jour l'ontologie locale associée. Liaison avec les autres ontologies locales	Ajouter une autre ontologie locale, mettre à jour l'ontologie locale associée
Hétérogénéité sémantique	Non (Sémantique unifiée du domaine)	Oui	Oui
Comparaison d'ontologies	Non car il n'existe qu'une seule ontologie	Difficile du fait qu'il n'existe pas de vocabulaire partagée	Simple du fait qu'il existe un vocabulaire partagé

Tableau 2.1 – Évaluation des approches ontologiques [Wache *et al.*, 2001].

— **Intersection d'ontologies** : À l'inverse, dans l'intersection d'ontologie, l'ontologie résultante ne contient que les parties communes des ontologies sources.

médiation pour les données distribuées. En permettant de garder leur contenu, il peut fournir l'interopérabilité entre les ontologies locales quand celles-ci ne peuvent pas être intégrées ou fusionnées en raison de la contradiction mutuelle de leurs informations.

2.2.3 Synthèse

La table 2.1 présente une comparaison de trois approches appliquées dans l'intégration sémantique de données hétérogènes. À partir des caractéristiques comparées, l'approche hybride possède des avantages remarquables. En effet, elle permet la maintenance des ontologies avec un effort d'implémentation raisonnable.

En fonction de l'approche utilisée, les techniques de *mapping* d'ontologies peuvent être utilisées en vue de trouver des correspondances entre les ontologies ou la fusion d'ontologies peuvent être réalisée pour former l'ontologie globale. Comme il existe de nombreux problèmes d'hétérogénéité des vocabulaires utilisés dans les ontologies, dans la littérature, tous les systèmes de *mapping* d'ontologie existants jusqu'à présent nécessitent une intervention des experts du domaine [Elbyed, 2009].

Conclusion du chapitre

Dans ce chapitre, nous présentons les ontologies et leurs rôles en nous concentrant sur la résolution de problèmes d'hétérogénéité de données. Trois approches d'intégration de données ont été étudiées : l'approche médiateur, l'approche matérialisée et l'approche hybride. Chacune possède ses propres avantages et inconvénients. Pour cet raison, deux prototypes différents ont été successivement développés pour vérifier leur capacité au niveau du fonctionnement et des performances dans l'intégration de données à composantes spatio-temporelles.

Enfin, les approches d'intégration de données à base ontologique ont été étudiées : les approches à ontologie unique, les approches à ontologies multiples et les approches hybrides. Quelle que soit l'approche utilisée, il est nécessaire de posséder des ontologies à priori, soit une seule ontologie du domaine utilisée en tant qu'ontologie globale connectant les sources, soit un ensemble d'ontologies locales correspondant chacune à une source. Ensuite, les techniques de fusion d'ontologies et de *mapping* d'ontologies peuvent être utilisées pour trouver des correspondances entre ces ontologies locales. Dans le cas des approches hybrides, c'est grâce à ces correspondances que nous pouvons construire l'ontologie globale et modifier les ontologies locales au fur et à mesure pour nous conformer à cette ontologie.

Chapitre 3

Web sémantique et Base de connaissances

Sommaire

3.1 Web sémantique 33
3.1.1 Modélisation de connaissances 34
3.1.1.1 RDF 34
3.1.1.2 RDFS 36
3.1.1.3 OWL 39
3.1.2 Interrogation 41
3.1.3 Inférence 43
3.1.3.1 SWRL 44
3.1.3.2 SPARQL comme langage de règles 45
3.2 Base de connaissances 46
3.2.1 Peuplement de l'ontologie 47
3.2.1.1 Les outils de traduction 49
3.2.1.2 Traduction de sources non relationnelles . . . 52
3.2.2 Stockage de données géospatiales 52
3.2.2.1 Parliament 52
3.2.2.2 Strabon 53
3.2.3 Interrogation de données géospatiales 54
3.2.3.1 GeoSPARQL 54
3.2.3.2 stSPARQL 56
3.2.3.3 Synthèse 57

Introduction du chapitre

Le W3C[1] a standardisé une pile stratifiée de langages ontologiques qui possèdent les avantages des formalismes de représentation des connaissances et des méthodes de modélisation conceptuelle pour les bases de données. La normalisation a encouragé la création de nouvelles ontologies et le portage d'ontologies existantes dans le Web sémantique.

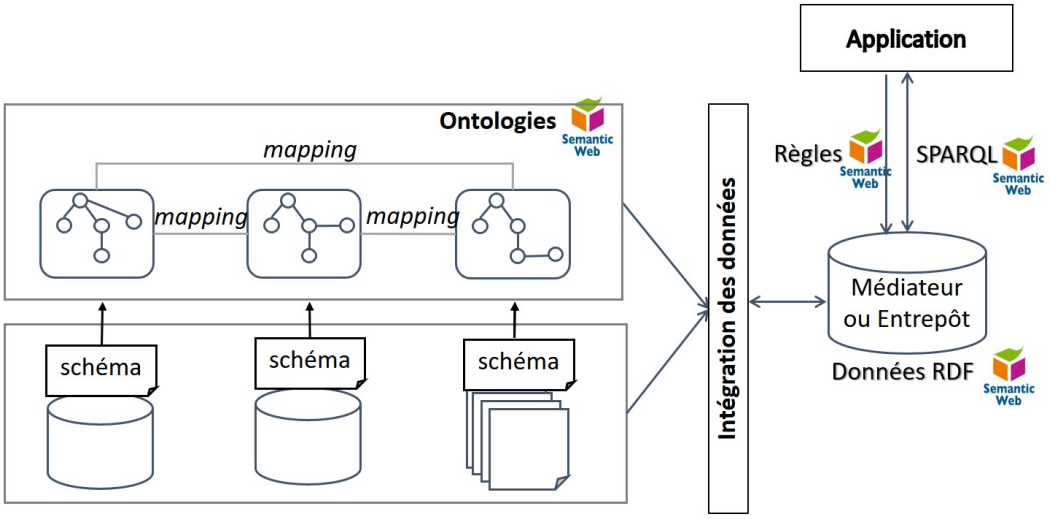


Figure 3.1 – Intégration sémantique de données à l'aide du Web sémantique.

Dans ce chapitre, nous présentons les technologies du Web sémantique qui visent à développer des ontologies modélisant les connaissances d'un domaine et à construire des ressources partageables. Ces technologies pourront servir à l'intégration sémantique de donnnées (Figure 3.1). La deuxième partie consiste à étudier les techniques permettant la construction d'une base de connaissances géospatiale. Nous abordons d'abord le peuplement de l'ontologie à l'aide des outils de traduction, ensuite le stockage de données spatiales en RDF grâce aux triplestores géospatiaux et finalement les langages permettant l'interrogation de ces données.

3.1 Web sémantique

Au début des années 1990, la majorité du contenu de la toile était conçu pour être lu par des êtres humains mais pas pour être manipulé symboliquement par des programmes informatiques. Cette conception du web était bien trop limitée. Elle ne permettait pas de partager du savoir et de rendre des connaissances directement utilisables. Il ne peut donc être facilement manipulable par des agents logiciels car il manque une dimension sémantique permettant de relier les différentes informations afin de permettre le traitement automatique.

L'expression «Web Sémantique» est due à Tim Berners-Lee qui fait référence par cette notion à l'évolution du web comme étant un vaste espace d'échange de données entre humains et machines [Berners-Lee, 1998]. Un des objectifs fondamentaux du Web sémantique est l'échange de ressources entre machines, afin de permettre l'exploitation de grands volumes d'informations et de services. Les ontologies jouent ici un rôle important car elles supportent la réalisation du Web sémantique. En effet, elles permettent de fournir des vues structurées et partageables des ressources et de définir l'information par une sémantique formelle.

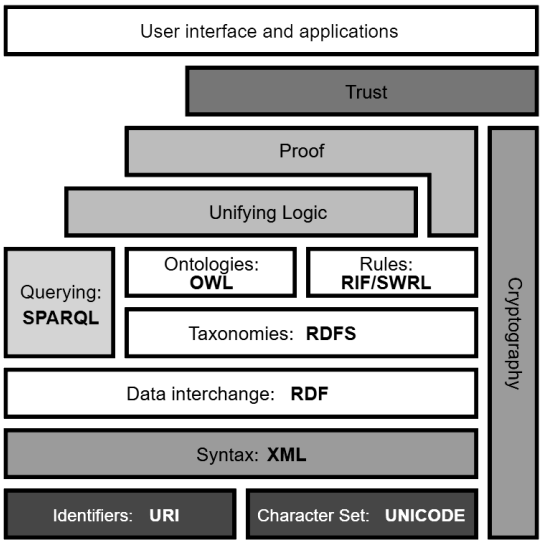


Figure 3.2 – Pile du Web sémantique.

L'ensemble des technologies du Web sémantique est organisé dans une architecture en couches, appelée «Pile du Web sémantique» (Figure 3.2). Cette architecture constitue la vision du W3C envers l'architecture du Web sémantique. Ainsi l'ensemble des technologies, y compris les langages et les protocoles, qu'elle comporte est standardisé et fait toujours l'objet de recherches et de travaux d'amélioration et de normalisation au sein du W3C.

Certaines couches sont déjà implémentées alors que les couches supérieures sont en cours de développement. Chaque couche réfère à la couche du dessous

et a une fonction bien déterminée dans l'architecture. Ainsi, les langages et protocoles permettant de remplir leurs fonctions existaient déjà ou ont été conçus et créés pour répondre aux spécifications de chaque couche. Dans cette section, nous présentons brièvement les principales composantes du Web sémantique qui visent à modéliser les connaissances de façon sémantique ainsi qu'à interroger et raisonner sur ces connaissances.

3.1.1 Modélisation de connaissances

Le XML est utilisé dans la pile du Web sémantique en tant que syntaxe élémentaire pour structurer le contenu des documents mais il ne décrit pas la sémantique des documents. Dès lors, les couches au niveau supérieur sont proposées en vue de leur apporter la sémantique requise. Nous parlons ainsi du modèle RDFS et OWL permettant de définir des ontologies dites légères ou lourdes respectivement pour modéliser les connaissances d'un domaine, tout en se basant sur le modèle de données RDF.

En vue de représenter les connaissances dans le Web sémantique, il est nécessaire d'exprimer d'abord en RDF le modèle sémantique à utiliser. Une fois que ce schéma est publié, les primitives qui y sont décrites peuvent être utilisées dans une page web si on y inclut une référence à l'URI du schéma. L'application utilisera le schéma d'interprétation pour accéder à la sémantique de cette page. Dans le Web sémantique, un tel schéma de base incluant les primitives sémantiques peut être décrit en RDFS ou en OWL. Dans les paragraphes suivant, nous présentons le langage RDF et ses deux extensions, RDFS et OWL.

3.1.1.1 RDF

RDF [2] (Resource Description Framework) est une famille de spécifications du W3C conçue à l'origine comme un modèle de données de métadonnées. Il est devenu une méthode générale pour la description conceptuelle ou la modélisation de l'information implémentée dans les ressources web en utilisant une variété de notations de syntaxe et de formats de sérialisation des données. Ainsi, RDF facilite l'interopérabilité entre les applications qui échangent des informations sur le Web et les rend compréhensibles par les machines.

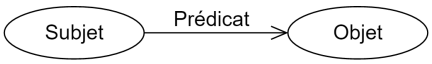


Figure 3.3 – Un graph RDF.

La syntaxe de RDF repose sur le langage XML. Celui-ci fournit une syntaxe pour encoder des donnés alors que RDF fournit un mécanisme pour décrire leur

2. https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

sens. Un des buts de RDF est de rendre possible la spécification de la sémantique de données en se basant sur XML d'une manière standardisée et interopérable.

En RDF, les ressources sont associées à des URI pour l'identification. Ainsi, on peut définir des relations entre deux ressources ou entre une ressource et un littéral. Pour cela, RDF représente les données sous forme d'un graphe qui est un ensemble de triplets ; chacun est composé de trois éléments comme illustrés par la Figure 3.3 :

— **Sujet** : Un sujet représente une ressource à décrire, qui est une entité accessible via une URI.

— **Prédicat** : Un prédicat représente un type de propriété applicable à un sujet. Il définit une relation binaire entre un sujet et un objet permettant ainsi d'associer de l'information sémantique au sujet.

— **Objet** : Un objet représente la valeur d'une propriété qui peut être une autre ressource ou une valeur littérale.

Le sujet et l'objet peuvent être identifiés par un URI ou un nœud anonyme mais le prédicat doit être identifié absolument par un URI.

La Figure 3.4 représente le graph RDF correspondant à l'extrait des données de la ville de Paris sur Geonames. Le sujet est la ville de Paris identifié par l'URI *http://sws.geonames.org/2988507/*.

Il est lié à deux objets : *http://dbpedia.org/resource/Paris* par les prédicats *rdfs:seeAlso* et *http://www.geonames.org/ontology#P* par *ns0:featureClass*.

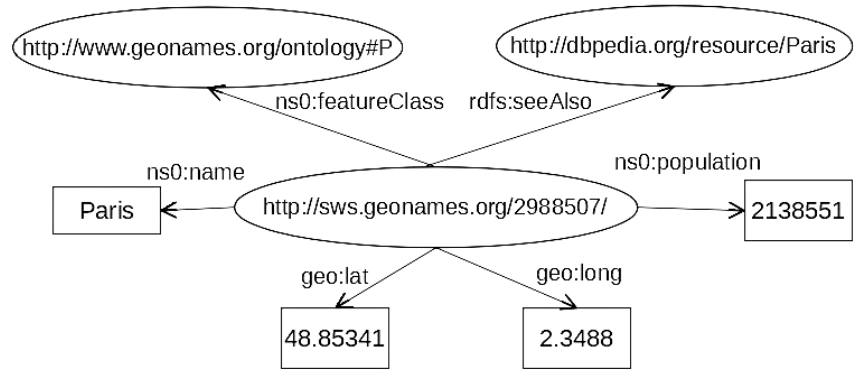


Figure 3.4 – Un extrait du graph RDF représentant la ville de Paris sur Geonames.

Les documents RDF peuvent être écrits en différentes syntaxes comme N3 [3], N-triple [4], Turtle [5] et RDF/XML [6]. Bien que Turtle permette une lecture plus

3. https://www.w3.org/TeamSubmission/n3/
4. https://www.w3.org/TR/2014/REC-n-triples-20140225/
5. https://www.w3.org/TR/turtle/
6. https://www.w3.org/TR/rdf-syntax-grammar/

facile pour les utilisateurs humains, seule RDF/XML, définie par le W3C pour exprimer un graphe RDF en tant que document XML, est recommandée. Le code 3.1 provenant de Geonames est un exemple de la représentation des données de la ville de Paris en RDF/XML.

Code 3.1 – Données RDF de la ville de Paris récupérées sur Geonames

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <rdf:RDF>
3  <gn:Feature rdf:about="http://sws.geonames.org/2988507/">
4  <rdfs:isDefinedBy rdf:resource="http://sws.geonames.org/2988507/
       about.rdf"/>
5  <gn:name>Paris</gn:name>
6  ...
7  <gn:featureClass rdf:resource="http://www.geonames.org/ontology#P
       "/>
8  <gn:featureCode rdf:resource="http://www.geonames.org/ontology#P.
       PPLC"/>
9  <gn:countryCode>FR</gn:countryCode>
10 <gn:population>2138551</gn:population>
11 <wgs84_pos:lat>48.85341</wgs84_pos:lat>
12 <wgs84_pos:long>2.3488</wgs84_pos:long>
13 <gn:parentFeature rdf:resource="http://sws.geonames.org
       /6455259/"/>
14 <gn:parentCountry rdf:resource="http://sws.geonames.org
       /3017382/"/>
15 ...
16 </rdf:RDF>
```

La structure de RDF est extrêmement générique et sert de base à un certain nombre de schémas ou vocabulaires dédiés à des applications spécifiques. Une partie de ces vocabulaires est spécifiée par le W3C, comme les langages d'ontologie RDFS et OWL ou le vocabulaire SKOS [7] pour la représentation des thésaurus. D'autres vocabulaires non spécifiés par le W3C sont néanmoins largement utilisés et constituent des standards dans la communauté du Web sémantique. FOAF [8], un vocabulaire destiné à la représentation des informations personnelles, en est un exemple.

3.1.1.2 RDFS

RDFS [9] a pour but d'étendre RDF en décrivant de façon précise les ressources utilisées pour étiqueter les graphes. Il permet de déclarer en RDF les propriétés

7. Simple Knowledge Organization System : https://www.w3.org/TR/swbp-skos-core-spec
8. Friend Of A Friend : http://xmlns.com/foaf/spec/
9. Resource Description Framework Schema : https://www.w3.org/TR/rdf-schema/

et les classes utilisées pour décrire des données RDF et les relations hiérarchiques qui existent entre elles.

Dans RDFS, toutes les ressources peuvent être regroupées en classes. Ces dernières sont aussi des ressources de sorte qu'elles sont identifiées par des URI et peuvent être décrites en utilisant des propriétés. Les membres d'une classe sont des instances de cette classe. Cette relation est indiquée grâce à la propriété *rdf:type*.

Les classes principales de RDFS sont (Figure 3.5) :

— **rdfs:Resource** : C'est la classe de tout. Toutes les choses décrites par RDF sont des ressources.

— **rdfs:Class** : Elle déclare une ressource comme une classe pour les autres ressources.

— **rdfs:Literal** : Elle décrit des valeurs littérales telle que des chaînes de caractères ou des nombres entiers.

— **rdfs:Datatype** : Il s'agit de la classe des types de données. La classe est à la fois une instance et une sous-classe de *rdfs:Class*. Chaque instance de cette classe est une sous-classe de *rdfs:Literal*.

— **rdfs:Property** : C'est la classe des propriétés.

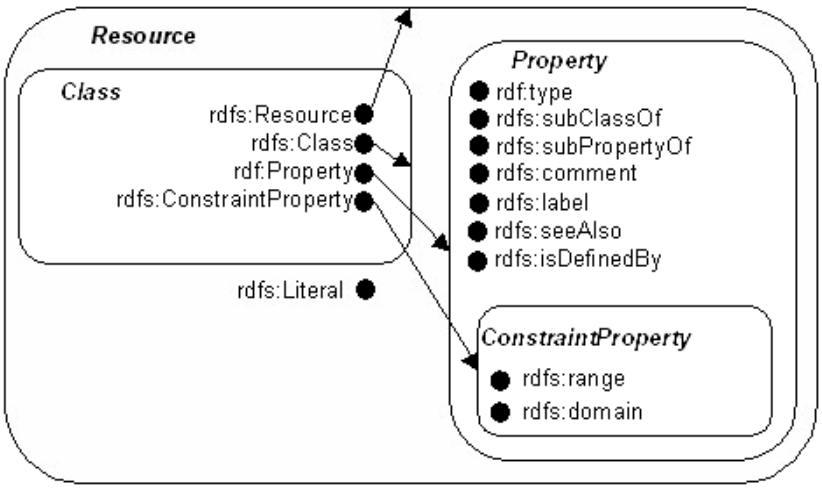


Figure 3.5 – Les composantes du langage RDFS.

De cette façon, RDFS considère que :

— Toutes les ressources sont des instances de la classe *rdfs:Resource*.

— Toutes les classes sont des instances de la classe *rdfs:Class* et sous-classes de *rdfs:Resource*.

— Tous les littéraux sont des instances de *rdfs:Literal*. Toutes les propriétés sont des instances de *rdf:Propriété*.

Une propriété de RDFS est un prédicat représentant une relation entre un sujet et un objet. Chaque propriété possède un domaine (domain) et une image (range) définis. Les propriétés les plus connues de RDFS sont :

— **rdfs:range** : Elle définit la classe ou le type de données de la valeur d'une propriété.

— **rdfs:domain** : Elle définit la classe du sujet lié à une propriété.

— **rdf:type** : Elle indique qu'une ressource est une instance d'une classe. Un Qname [10] communément accepté pour cette propriété est «a».

— **rdfs:subClassOf** : Elle déclare les hiérarchies des classes. Toutes les instances d'une classe sont des instances d'une autre.

— **rdfs:subPropertyOf** : Elle déclare les hiérarchies des propriétés. Toutes les ressources liées par une propriété sont également liées par une autre.

— **rdfs:label** : Elle fournit une version lisible par l'homme du nom d'une ressource.

— **rdfs:comment** : Elle fournit une description lisible par l'homme d'une ressource.

La Figure 3.6 illustre les propriétés *foaf:homepage* et *foaf:name* de l'ontologie FOAF [11]. Le sujet de ces propriétés sera typé par *foaf:Person* et leur valeur sera typée par *foaf:Document* et *rdfs:Literal* respectivement.

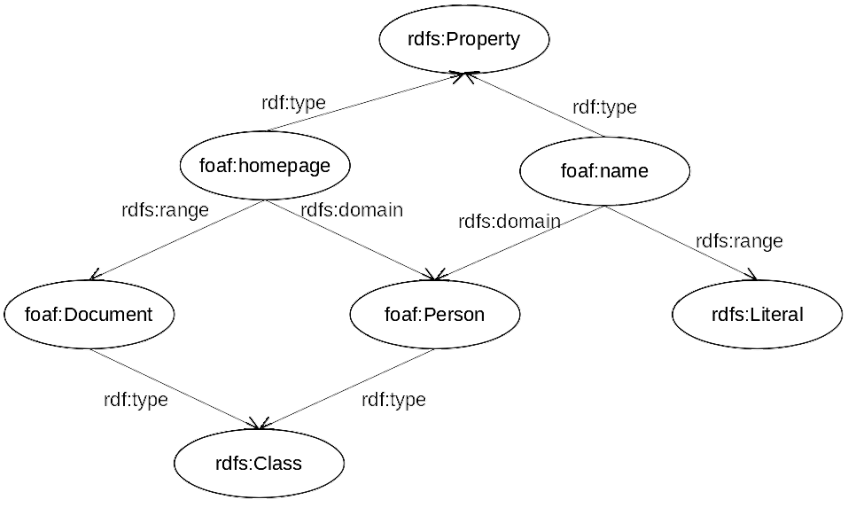


Figure 3.6 – Un extrait de l'ontologie FOAF en RDFS.

Le langage RDFS a certaines limites dans la description de situations complexes due à la seule définition des relations entre objets par des assertions. Par

10. https://www.w3.org/2001/tag/doc/qnameids-2004-01-14.html
11. Friend of a friend : http://xmlns.com/foaf/spec/

exemple, il est impossible de spécifier des propriétés d'unicité ou de définir des restrictions de cardinalité [Antoniou et van Harmelen, 2004] ou l'impossibilité de raisonner et de mener des raisonnements automatisés sur les modèles de connaissances établis par RDF/RDFS. Afin de pallier ces limites, de nouveaux langages plus expressifs, dont OWL, ont été introduits.

3.1.1.3 OWL

Tout comme RDFS, OWL [12] est un langage pour représenter des ontologies dans le Web sémantique. Il offre aux machines de plus grandes capacités d'interprétation du contenu Web que celles permises par RDF et RDFS grâce à un vocabulaire supplémentaire et une sémantique formelle. Il se différencie du couple RDF/RDFS par le fait qu'il est justement un langage d'ontologies, contrairement à RDFS. Si RDFS apportent la capacité de décrire des classes et des propriétés, OWL intègre en plus des outils de comparaison des propriétés et des classes comme l'identité, l'équivalence, le contraire, la cardinalité, la symétrie, la transitivité ou la disjonction, etc. Par conséquent, il est plus expressif que RDFS et apporte une meilleure intégration, une évolution, un partage et une inférence plus facile des ontologies. Il offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grâce à un vocabulaire plus large et à une vraie sémantique formelle.

La première recommandation d'OWL comporte trois sous-langages d'expressivités différentes. Chacun de ces sous-langages étant lui-même une extension de son prédécesseur.

— **OWL Lite** : Il est le sous langage d'OWL le plus simple. Il permet d'exprimer des hiérarchies de classes et de propriétés, certaines propriétés algébriques et des contraintes de cardinalité sur les propriétés. Dès lors, ce langage est destiné aux utilisateurs qui ont besoin d'une hiérarchie de concepts simple. Il est adapté, par exemple, aux migrations rapides à partir de thésaurus.

— **OWL DL** : OWL DL est fondé sur la logique descriptive (d'où son nom, OWL Description Logics). Il est plus complexe qu'OWL Lite en permettant une expressivité bien plus importante. Il garantit la complétude des raisonnements (toutes les inférences sont calculables) et leur décidabilité (leur calcul se fait en une durée finie).

— **OWL Full** : C'est la version la plus complexe d'OWL mais également celle qui permet le plus haut niveau d'expressivité. Il est compatible avec RDFS alors que les deux premières familles ci-dessus ne le sont pas puisque RDFS ne distingue pas individus, classes et propriétés. Toutefois, OWL Full ne garantit pas la complétude et la décidabilité.

Voici la liste des principaux énoncés que permet d'exprimer OWL :

12. Web Ontology Language : https://www.w3.org/TR/owl-features/

— Equivalence de classes en utilisant la propriété *owl:equivalentClass* : cette propriété lie une classe à une autre. Les deux classes contiennent exactement le même ensemble d'individus.

— Equivalence de propriétés en utilisant la propriété *owl:equivalentProperty*.

— Propriété fonctionnelle en utilisant *owl:FunctionalProperty* : Une propriété fonctionnelle est une propriété qui ne peut avoir qu'une (unique) valeur pour chaque instance.

— Relations inverses en utilisant la propriété *owl:inverseOf*.

— Relations symétriques en utilisant *owl:SymmetricProperty*, Si la paire (x, y) est une instance de P, alors la paire (y, x) est aussi une instance de P.

— Relations transitives en utilisant *owl:TransitiveProperty*, si une paire (x, y) est une instance de P, et la paire (y, z) est aussi instance de P, alors nous pouvons en déduire le couple (x, z) est également une instance de P.

— Contraintes de cardinalité en utilisant les propriétés *owl:maxCardinality* et *owl:minCardinality* : Une classe de tous les individus qui ont au moins N valeurs sémantiquement distinctes (individus ou valeurs de données) pour la propriété concernée, où N est la valeur de la contrainte de cardinalité.

— Liaison entre individus en utilisant la propriété *owl:sameAs*. Une telle déclaration indique que deux URI se réfèrent réellement à la même chose, autrement dit, les individus ont la même identité.

La Figure 3.7 représente un extrait de l'ontologie Geonames en OWL qui modélise les données de la ville de Paris dont une extraite est décrite par la Figure 3.4 et le Code 3.1.

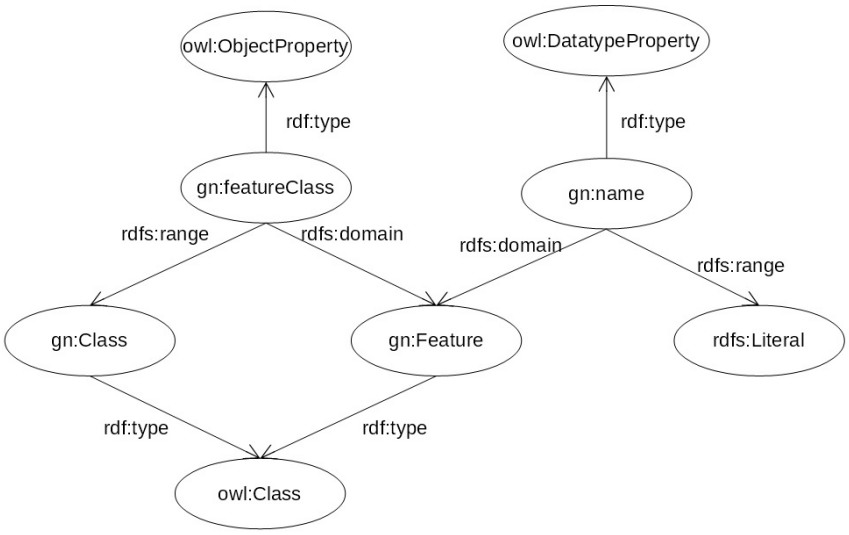


Figure 3.7 – Un extrait de l'ontologie Geonames en OWL.

OWL 2 [13] a été introduit pour répondre aux besoins d'évolution d'OWL [Grau *et al.*, 2008]. Il comprend trois profils [14] : OWL EL, OWL RL, et OWL QL qui restreignent les caractéristiques de modélisation disponibles afin de simplifier le raisonnement.

3.1.2 Interrogation

SPARQL [15] est un langage de requête et un protocole recommandé par le W3C pour interroger des graphes RDF. Ce langage de requête permet d'exprimer des requêtes interrogatives ou constructives :

— **Select (interrogative)** : Il permet d'extraire du graphe RDF un sous-graphe correspondant à un ensemble des ressources vérifiant les conditions définies dans sa clause Where.

— **Construct (constructive)** : Il engendre un nouveau graphe qui complète le graphe interrogé.

— **Ask** : Il permet de poser des requêtes à réponse booléenne sur un graphe RDF.

— **Describe** : Il permet de demander la description RDF d'une ressource ; son résultat est un ensemble de triplets RDF décrivant la ressource ciblée.

Il existe aussi des opérateurs qui permettent de modifier la séquence des solutions :

— **Distinct** pour éliminer les doublons.

— **Order by** pour trier les solutions.

— **Limit** pour couper le nombre de solutions en limitant le nombre de solutions retournées.

— **Offset** pour couper le nombre de solutions en indiquant de commencer les solutions générées après le nombre de solutions indiqué.

Par exemple, le code 3.2 représente une requête SPARQL visant à récupérer des villes de la France sur Geonames par un filtrage ?code="FR".

Code 3.2 – Récupération des villes de la France sur Geonames par une requête SPARQL

```
1 PREFIX ns0: <http://www.geonames.org/ontology#>
2 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
3 SELECT ?ville ?name ?lat ?long
4 WHERE
```

13. https://www.w3.org/TR/owl2-conformance/
14. https://www.w3.org/TR/owl2-profiles/
15. SPARQL Protocol and RDF Query Language : https://www.w3.org/TR/rdf-sparql-query/

```
 5 {
 6  ?ville ns0:featureClass ns0:P.
 7  ?ville ns0:name ?name.
 8  ?ville ns0:countryCode ?code.
 9  ?ville geo:lat ?lat.
10  ?ville geo:long ?long.
11 FILTER(?code="FR")
12 }
```

Depuis 2013, SPARQL 1.1 est la recommandation pour interroger des données RDF. Elle apporte beaucoup de nouveautés à la version précédente de SPARQL, notamment :

— **Agrégation** : Elle introduit différentes fonctions d'agrégation comme *Count, Sum, Min, Max, AVG, Group by, Having*, qui permettent d'effectuer des calculs sur un ensemble de résultats.

— **Projection d'expression** : il est possible d'assigner et créer de nouvelles valeurs avec le constructeur *AS*. Ce-ci peut être utilisée avec des agrégats.

— **Négation** : Deux opérateurs *Not exists* et *Minus* sont introduits pour exprimer la négation. Le premier permet de tester l'absence d'un certain pattern de graphe dans les graphes solutions recherchés. Le deuxième permet de supprimer de l'ensemble des graphes solutions ceux qui suivraient un certain pattern.

— **Sous-requêtes** : Une sous-requête est une requête de la forme *Select* imbriquée dans une autre et dont son résultat est utilisé dans la requête principale.

Aux côtés du langage de requête, SPARQL 1.1 Update permet de modifier des données RDF avec les opérateurs suivants :

— *Create* et *Drop* pour créer ou supprimer des graphes.

— *Insert* et *Delete* pour ajouter ou supprimer les triplets dans un graphe.

— *Insert data* et *Delete data* pour ajouter ou supprimer les données sans variables.

Par exemple, nous pouvons calculer le nombre des départements de la France dont la population est de plus de 500000 habitants par le code ci-dessous (Code 3.3).

Code 3.3 – Calcul du nombre des départements de la France dont la population est plus de 500000 habitants

```
1 PREFIX ns0: <http://www.geonames.org/ontology#>
2 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
3 PREFIX dbp: <http://dbpedia.org/property/>
4 PREFIX dbr: <http://dbpedia.org/resource/>
```

```
5  PREFIX dbo: <http://dbpedia.org/ontology/>
6  SELECT ?dep (COUNT(*) as ?nombre_ville)
7  WHERE
8  {
9   ?ville dbo:country dbr:France.
10  ?ville dbp:label ?name.
11  ?ville dbp:population ?pop.
12  ?ville dbp:department ?dep.
13 FILTER(?pop>500000)
14 }
15 GROUP BY ?dep
```

3.1.3 Inférence

L'inférence est la capacité de déduire de nouvelles données à partir de données disponibles. Dans le Web sémantique, les données sont modélisées comme un ensemble de ressources et leurs relations. L'inférence est alors un ensemble de procédures automatiques qui peut générer de nouvelles relations basées sur les données disponibles et certaines informations supplémentaires sous la forme d'un vocabulaire, par exemple, un ensemble de règles.

L'inférence est un des outils de choix pour améliorer la qualité des données en découvrant de nouvelles relations, en analysant le contenu de données ou en gérant les connaissances. Les techniques basées sur l'inférence sont aussi importantes pour découvrir les incohérences dans les données intégrées. Dans le Web sémantique, l'inférence se réalise grâce aux régimes d'inférence de base [16] de RDF, RDFS et OWL ou de règles personnalisées. L'implémentation de ce système par un moteur sémantique lors de l'interrogation de données RDF permet de récupérer des informations sans qu'elles soient explicites dans la base de connaissance.

Supposons les données de publications ci-dessous (Code 3.4) :

Code 3.4 – Exemple du régime d'inférence de RDFS

```
1 ex:livre1 rdf:type ex:Publication.
2 ex:livre2 rdf:type ex:Article.
3 ex:Article rdfs:subClassOf ex:Publication.
4 ex:publishes rdfs:range ex:Publication.
5 ex:Springer ex:publishes ex:livre3.
```

Grâce aux règles d'inférences RDFS [17], deux conséquences peuvent être déduites :

1. La règle rdfs9 peut être appliquée aux triplets (2) et (3) pour dériver un nouveau triplet: ex:livre2 rdf:type ex:Publication.

16. http://www.w3.org/TR/sparql11-entailment
17. https://www.w3.org/TR/rdf11-mt/#rdf-entailment

2. La règle rdfs3 peut être appliquée aux triplets (4) and (5) pour dériver un nouveau triplet: ex:livre3 rdf:type ex:Publication.

Les règles constituent une alternative ou un complément pour les ontologies. Une règle est vue comme une instruction conditionnelle de type «si la condition est vérifiée alors une conséquence doit être exécutée». Une nouvelle connaissance est ainsi ajoutée dans le cas de la satisfaction de la règle donnée. Ce processus va alors déduire de nouvelles informations à partir des observations et des règles.

De nombreux langages de règles ont vu le jour, tels que RuleML [Boley *et al.*, 2001], SWRL [Horrocks *et al.*, 2004], N3Logic [Berners-lee *et al.*, 2008], ou RIF [Kifer et Boley, 2013]. Ces langages ont des syntaxes et sémantiques différentes, ce qui pose des problèmes d'interopérabilité et d'échange. Or, celles-ci sont des éléments clés du Web sémantique. Parmi les solutions, les plus connues sont RIF et SWRL. La première est une recommandation du W3C depuis 2010 et a été conçue pour fournir un format d'échange de règles entre des systèmes de règles. La deuxième, quant à elle, apparaît comme la meilleure alternative entre l'expressivité, la standardisation et la décidabilité.

3.1.3.1 SWRL

SWRL [18] (Semantic Web Rule Language) [Horrocks *et al.*, 2004] est un langage de règles qui est implémenté dans de nombreux logiciel ou bibliothèque comme Protégé, Pellet ou encore OWL API. L'objectif de SWRL est d'étendre les potentialités du langage OWL-DL en permettant l'ajout de règles, tout en gardant une compatibilité maximale avec la syntaxe sémantique existante.

Prévu pour supporter les raisonnements basés sur les logiques de description et les règles de Horn, la structure d'une règle SWRL est de la forme: *antécédent => conséquence*. Elle signifie que si l'antécédent est vrai, alors la conséquence l'est aussi. L'antécédent et le conséquent d'une règle peuvent consister en zéro ou plusieurs atomes exprimés en utilisant des concepts OWL (classes, propriétés, individus, etc.). Toutefois, une règle avec plusieurs conséquents peut toujours être transformée en plusieurs règles avec un conséquent unique. En SWRL, seuls des prédicats unaires et binaires peuvent être utilisés, c'est-à-dire, tout atome est de la forme C(x) ou P(x,y).

En utilisant cette syntaxe, il est possible de définir que la combinaison des propriétés «parent» et «frère» implique une propriété «oncle» :

$$parent(?x, ?y) \bigwedge frère(?y, ?z) => oncle(?x, ?z)$$

SWRL propose une syntaxe facilement compréhensible pour l'utilisateur et possède également l'avantage de proposer des fonctions spécifiques utilisables au sein de règles appelées *built-ins*. Celles-ci permettent par exemple de réaliser

18. https://www.w3.org/Submission/SWRL/

des opérations mathématiques (*swrlb:add*, *swrlb:substract*, etc.), des comparaisons (*swrlb:equal*, *swrlb:greaterThan*, etc.) ou des manipulations de chaînes de caractères (*swrlb:contains*, *swrlb:replace*, etc.) ou encore des opérations sur les URI (*swrlb:anyURI*, *swrlb:resolveURI*, etc.).

Grâce à ces built-ins, on peut définir des règles d'inférences pour déduire de nouvelles informations. Le code 3.5 montre un exemple comment déduire la relation *inside* entre un instant et un intervalle de temps. Dans cet exemple, les concepts de l'ontologie OWL-Time et le *built-in* mathématique pour la comparaison des valeurs de temps sont utilisés.

Code 3.5 – Inférence de la relation *inside* entre un instant et un intervalle de temps par une règle SWRL

```
1  Instant(?x) ⋀ ProperInterval(?a) ⋀ hasBeginning(?a, ?b) ⋀ hasEnd
       (?a, ?c) ⋀ inXSDDateTime(?b, ?d) ⋀ inXSDDateTime(?c, ?e) ⋀
       inXSDDateTime(?x, ?y) ⋀ swrlb:lessThan(?y, ?e) ⋀ swrlb:
       greaterThan(?y, ?d) => inside(?x, ?a)
```

Ce mécanisme de raisonnement a été appliqué dans le modèle SOWL [Batsakis et Petrakis, 2011] qui a été par la suite amélioré par le système CHRONOS [Anagnostopoulos *et al.*, 2013]. [O'Connor et Das, 2009] a développé un ensemble de *built-ins*, appelé *SWRLTemporalBuiltIns*, à partir des *buil-ins* d'origine pour effectuer le raisonnement temporel dans leur ontologie temporelle.

3.1.3.2 SPARQL comme langage de règles

Dans la littérature, les requêtes SPARQL Construct/Update peuvent être utilisées à la place des règles pour déduire de nouvelles informations. Une requête SPARQL de la forme *Construct* produit un nouveau graphe RDF en remplaçant les variables du graphe de la clause *Construct* par les valeurs pour lesquelles le graphe requête de la clause *Where* s'apparie avec le graphe RDF interrogé. Une telle requête peut être vue comme une règle et son traitement comme l'application d'une règle en chaînage avant pour enrichir le graphe RDF. Les règles SPARQL s'établissent sous la forme ci-dessous, avec le conséquent et l'antécédent constitués d'un ou plusieurs triplets.

$$INSERT/CONSTRUCT \ \ conséquent \ \ [WHERE \ \ antécédent]$$

La règle définie dans 3.5 peut être représentée par une requête SPARQL Update comme ci-dessous (Code 3.6).

Code 3.6 – Inférence de la relation *inside* entre un instant et un intervalle de temps par une requête SPARQL Update

```
1  INSERT
2  {
3   ?x time:inside ?a.
```

```
4  }
5  WHERE
6  {
7   ?x rdf:type time:Instant.
8   ?x time:inXSDDateTime ?dt.
9   ?a rdf:type time:Interval.
10  ?a time:hasBeginning ?be.
11  ?a time:hasEnd ?end.
12  ?be time:inXSDDateTime ?dt1.
13  ?end time:inXSDDateTime ?dt2.
14 FILTER(?dt>?dt1 && ?dt<?dt2)
15 }
```

Cette approche a été appliquée dans plusieurs travaux de recherches. [Corby *et al.*, 2004] ont proposé Corese/KGRAM mettant en œuvre le SPARQL pour interroger les sources de données et appliquer des règles d'inférence sur celles-ci. [Polleres *et al.*, 2007] développe SPARQL++ qui utilisent SPARQL comme un langage de règles pour définir les correspondances entre les vocabulaires RDF et permettant de construire des requêtes étendues avec des fonctions d'agrégation intégrés. [Angles et Gutierrez, 2008] a prouvé que SPARQL est équivalent d'un point de vue expressivité à l'algèbre relationnelle. [Schenk et Staab, 2008] utilisent SPARQL comme langage de règles pour définir de nouvelles données RDF à partir de sources de données existantes. Le framework R2R [Bizer et Schultz, 2010], qui permet de publier des mappings sur le Web, est basé sur la forme CONSTRUCT de SPARQL pour exprimer des transformations de données. [Knublauch *et al.*, 2011] ont introduit SPIN [19] qui devient la norme de l'industrie *de-facto* pour représenter les règles et les contraintes SPARQL sur les modèles du Web sémantique.

3.2 Base de connaissances

Une base de connaissances regroupe des connaissances spécifiques d'un domaine, sous une forme exploitable par l'ordinateur. Elle peut contenir des règles (dans ce cas, on parle de base de règles), des faits ou d'autres représentations. Si elle contient des règles, un moteur d'inférence simulant les raisonnements déductifs logiques peut être utilisé pour déduire de nouveaux faits. Une autre manière de définir une base de connaissances est qu'il s'agit d'une ontologie peuplée par des individus [Baader *et al.*, 2003].

Une base de connaissances se compose d'une TBox représentant la terminologie du domaine et d'une ABox qui déclare les assertions et instances de ce domaine.

19. SPARQL inférences Notation : http://spinrdf.org/

— **TBox** : Elle est constituée du vocabulaire du domaine, représentée par des ontologies. Celles-ci décrivent des concepts et leurs relations.

— **ABox** : Elle permet de décrire des individus en spécifiant leur identité, les concepts auxquels ils appartiennent, et leurs relations avec d'autres individus.

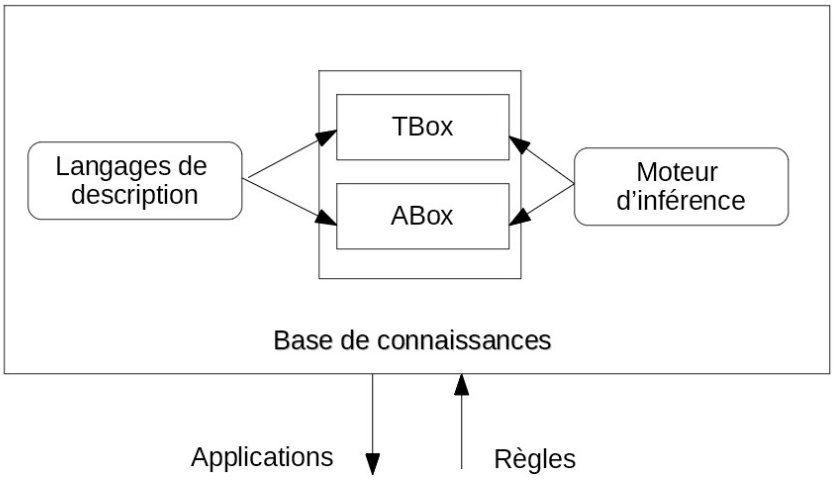


Figure 3.8 – Illustration d'une base de connaissances [Baader *et al.*, 2003]

Les logiques de description permettent d'effectuer le raisonnement sur la base de connaissances. D'un part, le raisonnement sur la TBox consiste à calculer les relations de subsomption qui existent entre les différents concepts et à vérifier la consistance des connaissances représentées. D'autre part, le raisonnement sur l'ensemble de la base permet de vérifier les instances ou de répondre à des requêtes.

3.2.1 Peuplement de l'ontologie

Comme présenté, une ontologie est considérée comme un ensemble de concepts, leurs relations et d'individus (ou instances) qui leurs sont associés. La conception d'une ontologie se fait en deux étapes :

1. **Conceptualisation d'ontologie** : Elle a pour but de faire émerger les concepts et relations du domaine ainsi que les axiomes permettant d'ordonner les concepts et les relations et de classer leurs instances. En d'autres termes, il s'agit de la construction de la TBox.
2. **Peuplement d'ontologie** : Elle consiste à associer des instances aux concepts et aux relations existantes. Autrement dit, il s'agit de la composition de l'ABox. La source de ces instances est variée, notamment des informations extraites d'un corpus de documents ou des données qui sont sauvegardées dans des bases de données.

Les techniques de peuplement d'ontologie s'appuient sur des approches de correspondance (appelées aussi «traduction» ou «mapping»), appelées RDB2RDF ou OBDA (Ontology-Based Data Access) qui visent à exposer les données en graphes RDF [Malhotra et Melton, 2008]. Ces derniers transforment les données relationnelles en graphe RDF et établissent par conséquent une correspondance entre les ontologies et les schémas de ces bases de données relationnelles.

On peut distinguer deux approches de correspondance de données :

— **Matérialisation de données** : Dans cette approche, on applique la transformation statique de la base de données source en une représentation RDF par des processus ETL, comme dans les approches d'entrepôt. Les règles de traduction sont appliquées aux données sources pour créer un graphe RDF équivalent. Cette étape est aussi appelé «graph dump», «RDF dump» ou «graph extraction». Lorsque la transformation est terminée, le graphe RDF obtenu peut être chargé dans un triplestore, une base de données dédiée pour les données RDF, et accessible via un moteur de requête SPARQL [Michel *et al.*, 2014]. Une telle approche est appliquée pour publier les données géospatiales sur le web [Chentout et Vaisman, 2013, Skjæveland *et al.*, 2013, Atemezing *et al.*, 2014, Koubarakis *et al.*, 2016] ou intégrer des données géospatiales hétérogènes dans l'agriculture [Pokharel *et al.*, 2014], dans la transport [Seliverstov et Rossetti, 2015] ou dans la tourisme [Lo Bue et Machì, 2015].

  Cette approche souffre de tous les problèmes usuels de réplication de données, tels que le maintien de plusieurs copies des données dans différents modèles de données et la mise à jour des données.

— **Traduction de données à la demande** : Dans cette approche, on utilise un «wrapper» qui expose le schéma de la base de données relationnelle comme un modèle RDF et traduit à la volée les requêtes SPARQL sur le modèle RDF exposé en requêtes SQL sur la source relationnelle. D'après [Gray *et al.*, 2009], il n'est pas encore possible d'utiliser les outils pour exposer les bases de données relationnelles existantes en tant que RDF à la demande. Le problème principal est que les performances de ces outils sont nettement plus faibles que l'accès aux données en utilisant soit des triplestore RDF, soit le modèle relationnel sous-jacent. Une grande partie des mauvaises performances de ces systèmes peut être expliquée par :

  — Les requêtes traduites n'exploitent pas les puissances du modèle relationnel ou de l'optimiseur de requêtes.

  — Les conditions de sélection sont calculées par ces systèmes plutôt que poussées à la base de données.

3.2.1.1 Les outils de traduction

Dans cette partie, nous présentons deux outils libres simples et efficaces. Il existe également d'autres outils comme Datalift [20] [Scharffe *et al.*, 2012] et Morph [21][Priyatna *et al.*, 2014].

§1 D2RQ

D2RQ [22] [Bizer, 2004] est une plateforme permettant l'accès à des bases de données relationnelles comme les graphes RDF virtuels en lecture seulement. Il offre un accès basé sur RDF pour le contenu des bases de données relationnelles, sans avoir à les reproduire dans un triplestore. La figure 3.9 décrit l'architecture de l'outil dont fonctionnalités principales sont :

— Interroger une base de données non-RDF en utilisant RDQL.

— Publier le contenu d'une base de données non-RDF sur le web en utilisant l'API RDFNet.

— Réaliser des inférences RDFS et OWL sur le contenu d'une base de données non-RDF en utilisant l'API Jena.

— Accéder aux informations dans une base de données non-RDF en utilisant l'API Jena.

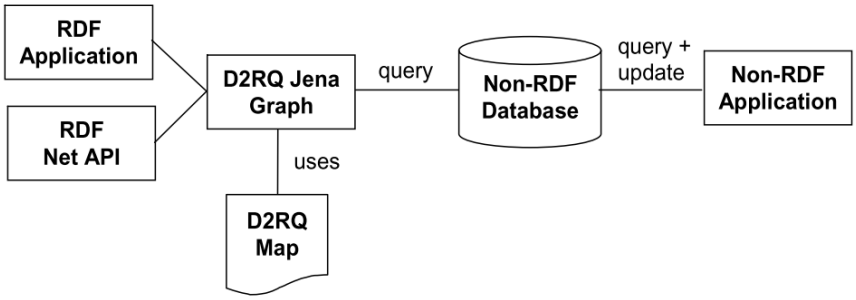


Figure 3.9 – Archecture du système D2RQ [Bizer, 2004].

L'objet principal de D2RQ est le *ClassMap* qui représente une classe ou un groupe de classes similaires de l'ontologie. Il indique si les instances sont identifiées en utilisant des valeurs de colonne à partir de la base de données en utilisant un URI.

Le code ci-dessous (Code 3.7) est un exemple d'un mapping entre la colonne *esp_nom* de la table *Espece* à la propriété *gem:name* de la classe *gem:Species* :

20. http://datalift.org/
21. http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/technologies/315-morph-rdb/index.html
22. http://d2rq.org/

Code 3.7 – Mise en correspondance entre une ontologie et un schéma relationel

```
map:Name a  d2rq:PropertyBridge;
d2rq:belongsToClassMap map:Species;
d2rq:property gem:name;
d2rq:column "espece.esp_nom";
```

Il existe quelques inconvénients :

— Ce système fournit certaines optimisations de requêtes mais celles-ci sont souvent insuffisantes. Par exemple, les requêtes SQL générées peuvent contenir un nombre excessif de jointures [Priyatna *et al.*, 2014].

— Il offre son propre langage de *mapping* et prend en charge uniquement un fragment de R2RML.

— Aucun mécanisme d'inférence n'est inclus.

Plusieurs outils ont été développés sur l'expérience acquise avec le framework D2RQ. D2R Server [Bizer et Cyganiak, 2006, Cyganiak et Bizer, 2006] est un outil permettant la publication du contenu des bases de données relationnelles sur le web. Il permet aux agents Web de récupérer des ressources en RDF et XHTML et d'interroger des bases de données non RDF à l'aide du langage de requête SPARQL. [Eisenberg et Kanza, 2012] ont introduit D2RQ/Update, une extension de D2RQ permettant l'exécution des requêtes SPARQL/Update sur les données RDF virtuelles. [Valle *et al.*, 2010] ont présenté un langage de traduction déclarative ainsi qu'un prototype, appelé G2R, qui permet d'exécuter une requête SPARQL impliquant des calculs spatiaux.

La Figure 3.10 représente l'architecture serveur D2R en basé sur les fonctionnalités de D2RQ.

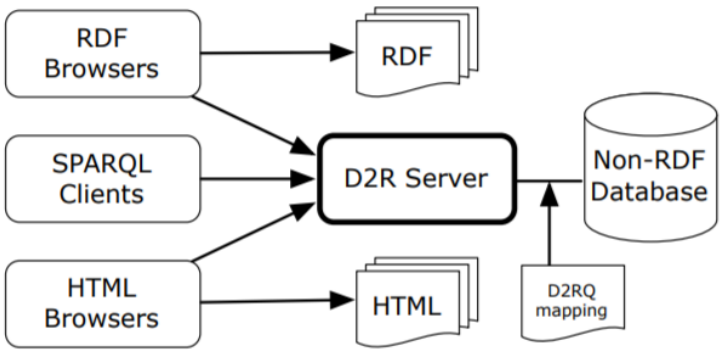


Figure 3.10 – Architecture du système de D2R Server [Bizer et Cyganiak, 2006].

§2 Ontop

Ontop [23] [Giese *et al.*, 2015, Calvanese *et al.*, 2016] repose sur des bases théoriques solides et a été conçu et mis en œuvre en respectant les normes du W3C. Il prend en charge toutes les bases de données relationnelles majeures et met en œuvre de nombreuses techniques d'optimisation pour offrir un bon niveau de performances. Ontop a été adopté dans plusieurs cas d'utilisation académiques et industriels.

L'architecture d'Ontop se compose de quatre couches différentes (Figure 3.11) :

— Les entrées, c'est-à-dire les artefacts spécifiques du domaine tels que l'ontologie, la base de données, les mappings et les requêtes.

— Le noyau du système prenant en charge la traduction, l'optimisation et l'exécution des requêtes.

— Les APIs exposant les interfaces Java standard aux utilisateurs du système.

— Les applications permettant aux utilisateurs d'exécuter des requêtes SPARQL sur les bases de données.

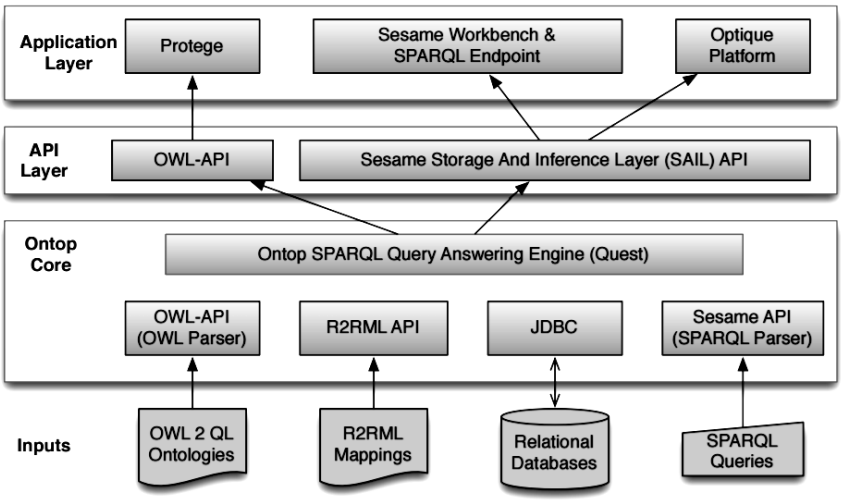


Figure 3.11 – Architecture du système Ontop [Calvanese *et al.*, 2016].

Ontop-spatial [Bereta et Koubarakis, 2016, Bereta *et al.*, 2016] étend Ontop pour supporter la traduction GeoSPARQL-to-SQL à la volée sur les bases de données géospatiales et devient ainsi le premier système de traduction permettant les requêtes spatiales en GeoSPARQL. Il est capable de se connecter à une base de données géospatiale et de créer des graphes virtuels géospatials RDF, en utilisant des ontologies et des correspondances. C'est aussi la première implémentation qui prend en charge l'extension de réécriture des requêtes GeoSPARQL.

23. http://ontop.inf.unibz.it/

3.2.1.2 Traduction de sources non relationnelles

Il existe de nombreux outils pour exposer des sources non relationnelles en données RDF. Par exemple, Teiid [24] ou Exareme [25] sont utilisés dans Ontop ; ou RDF123 [26] [Han *et al.*, 2008], XLWrap [27] [Langegger et Wöß, 2009], CSV2RDF [28] [Ermilov *et al.*, 2013], ou Datalift permettent la publication de données RDF à partir de fichiers tabulaires. Pourtant, [Seliverstov et Rossetti, 2015] ont proposé de tout d'abord charger les fichiers tabulaires dans une base de données relationnelle pour pouvoir ensuite exporter le contenu de la base en RDF. Le plus grand effort de cette étape est de créer un schéma relationnel pour la base. D'une part, cette étape offre la possibilité de répondre aux requêtes sur l'ensemble des données en utilisant un SGBD standard et fournit une base de comparaison pour les différents niveaux d'interrogation sémantique. D'autre part, à travers cette étape, les types de données sont déterminés en examinant la base ; par conséquent, ils seront convertis en de bons formats.

3.2.2 Stockage de données géospatiales

Un triplestore est une base de données spécialement conçue pour le stockage et la gestion de données RDF. Il permet l'interrogation à l'aide du langage de requête SPARQL. Avec l'expansion des données géospatiales et du Web de données [29], les développeurs cherchent à incorporer la capacité de gestion de ces données à leur triplestore. Dans cette partie, parmi les triplestores géospatials existants tels que AllegroGraph [30], Virtuoso Universal Server [31], USeekM [32] ou Oracle Spatial and Graph [33], nous présentons Parliament et Strabon. Ce sont deux triplestores géospatiaux «open-source» qui supportent la majorité des types de géométries et relations topologiques. Ces triplestores donnent aussi naissance à deux langages d'intégration de données géospatiales RDF, respectivement GeoSPARQL et stSPARQL.

3.2.2.1 Parliament

Parliament [34] est un triplestore «open-source» de haute performance développé par BBN Technologies. L'objectif initial du Parliament était de créer un

24. http://teiid.jboss.org/
25. http://madgik.github.io/exareme/
26. http://rdf123.umbc.edu/
27. http://xlwrap.sourceforge.net/
28. https://www.w3.org/TR/csv2rdf/
29. http://linkeddata.org/
30. https://franz.com/agraph/allegrograph/
31. https://virtuoso.openlinksw.com/
32. https://dev.opensahara.com/projects/useekm/
33. https://www.oracle.com/database/spatial/index.html
34. http://parliament.semwebcentral.org/

mécanisme de stockage spécifiquement optimisé pour les besoins du Web sémantique. Depuis sa conception initiale, le triplestore a servi en tant que composante de base à plusieurs projets à BBN pour un certain nombre de clients du gouvernement des États-Unis.

Parliament fonctionne sur une seule machine et intègre un certain nombre de logiciels tiers open-source. Il est compatible avec RDF, RDFS, OWL, SPARQL, et GeoSPARQL. Sa structure de stockage se compose d'une table de ressources, d'une table de déclaration, et d'un dictionnaire de ressources fournissant la mise en correspondance bidirectionnelle une à une entre une ressource et son identifiant. Le moteur de stockage est basé sur les indices des ressources et des déclarations dont l'objectif général est de diviser les requêtes SPARQL avec l'information géospatiale en plusieurs parties. Ainsi, il permet un plan de requête efficace entre les composantes spatiales et non-spatiales.

Parliament permet des requêtes spatiales par le biais de l'utilisation des fonctions d'extension de SPARQL standardisé, appelée GeoSPARQL, comme *Within, Intersects, Overlaps, Crosses*. La syntaxe de GeoSPARQL est légèrement modifiée, par exemple, à la place de *geo:asWKT*, un nom de prédicat peut être utilisé pour attacher une géométrie donnée à une ressource ou un sujet.

À l'exception de la requête des règles de réécriture, le moteur de recherche prend en charge GeoSPARQL et permet la recherche sur les géométries indexées en utilisant un R-tree standard.

3.2.2.2 Strabon

Strabon [35] [Kyzirakos *et al.*, 2012] est un prototype académique pour les données spatio-temporelles en RDF. Strabon offre des sélections et des jointures spatiales ainsi qu'un ensemble de fonctions spatiales. Le langage de requête proposé par Strabon est stSPARQL, ce-ci interagit avec le modèle stRDF, une extension de RDF. Le système est construit en étendant le triplestore Sesame pour la gestion des données RDF spatiales et non spatiales à l'aide de PostGIS. Son objectif est de créer une couche intégrable dans Sesame de façon transparente. Par conséquent, Strabon peut être considéré comme un Sesame Sail qui peut être placé sur le dessus d'une installation de Sesame.

Strabon adhère à l'architecture modulaire de Sésame avec trois composantes principales :

— **Moteur de requête** : Il a pour but d'évaluer les requêtes écrites en langage stSPARQL. La requête est analysée et optimisée. Ensuite, un plan d'exécution est créé et exécuté et les résultats sont renvoyés au client. L'évaluateur et l'optimiseur de Sesame ont été étendus pour traiter les requêtes stSPARQL.

35. http://www.strabon.di.uoa.gr

— **Gestionnaire de stockage** : Toutes les informations spatiales et thématiques sont gérées par PostgreSQL/PostGIS. Les composants de Sesame ont été étendus pour supporter les littéraux géospatiaux exprimés en stRDF.

— **Base de données** : PostGIS ou monetdb est utilisé à la fois pour le stockage des données et pour l'évaluation des requêtes stSPARQL.

Strabon utilise les normes de l'OGC[36], comme WKT et GML, pour la représentation et l'interrogation de données géospatiales. Les types de données *strdf:WKT* et *strdf:GML* sont introduit pour représenter les littéraux géospatiaux. Le type de données *strdf:geometry* est également utilisé pour représenter la sérialisation de la géométrie de manière indépendante de la norme de sérialisation utilisée.

3.2.3 Interrogation de données géospatiales

Beaucoup d'études ont été récemment réalisées sur l'extension de SPARQL pour interroger des données géospatiales en RDF. Ces études ont récemment abouti à la définition de nouveaux langages d'interrogation dont le premier est SPARQL-ST [Perry *et al.*, 2011] visant à interroger des données spatiales et non spatiales qui changent au fil du temps. Néanmoins, cette partie n'aborde que les deux langages les plus connues et utilisés dans la littérature, GeoSPARQL et stSPARQL.

3.2.3.1 GeoSPARQL

GeoSPARQL[37] [Battle et Kolas, 2011] est une nouvelle norme de l'OGC. Son intention est de fournir un moyen standard pour exprimer et interroger des éléments spatiaux en RDF de sorte que les utilisateurs puissent échanger des données facilement, et que les développeurs du triplestore puissent avoir un format standard pour l'indexation. GeoSPARQL définit :

— Une ontologie topologique en RDFS/OWL pour la représentation de données spatiales à l'aide de :
  - Du GML ou du WKT.
  - Des vocabulaires de relations topologiques et des ontologies pour le raisonnement qualitatif.

— Une interface d'interrogation SPARQL utilisant :
  - Un ensemble de fonctions d'extension topologiques SPARQL pour le raisonnement quantitatif.
  - Un ensemble de règles d'inférence en RIF pour la transformation et l'interprétation de requête.

36. Open Geospatial Consortium :http://www.opengeospatial.org/
37. http://www.opengeospatial.org/standards/geosparql

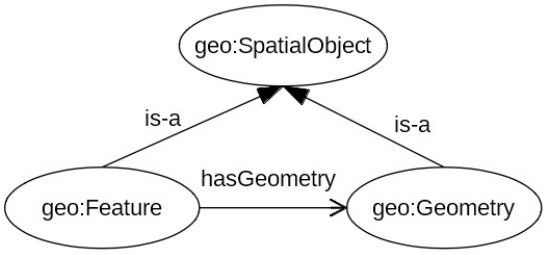


Figure 3.12 – Ontologie de GeoSPARQL [Battle et Kolas, 2011].

Afin d'exécuter des requêtes GeoSPARQL sur un ensemble de données, la partie spatiale de ces données doit s'exprimer par l'ontologie de GeoSPARQL. Celle-si se compose de trois classes principales (Figure 3.12) :

— **geo:Feature** : Elle représente des entités qui peuvent avoir une localisation spatiale, par exemple, un parc, un monument...

— **geo:Geometry** : Elle décrit un emplacement spatial, c'est-à-dire un ensemble de coordonnées.

— **geo:SpatialObject** : C'est une classe racine des deux classes *geo:Feature* et *geo:Geometry*.

La propriété *geo:hasGeometry* met en liaison une entité à son emplacement. En séparant les entités réelles et leurs emplacements, GeoSPARQL permet à plusieurs emplacements d'être liés à une entité à des fins diverses. La ressource de la géométrie a alors un littéral qui est lié à une propriété en utilisant soit *geo:asWKT* ou *geo:asGML*. Pour la comparaison topologique entre les géométries, trois méthodes sont possibles: les fonctions de filtrage, les propriétés entre les géométries et les propriétés entre les entités.

Le code ci-dessous (Code 3.8) est un exemple d'une requête GeoSPARQL qui a pour but de trouver les monuments se situant dans un polygone avec une fonction de filtrage.

Code 3.8 – Recherche des monuments se situant dans un polygone avec une fonction de filtrage

```
1  PREFIX geo: <http://www.opengis.net/def/geosparql/>
2  PREFIX geof: <http://www.opengis.net/def/geosparql/function/>
3  PREFIX sf: <http://www.opengis.net/def/sf/>
4  SELECT ?m
5  WHERE
6  {
7   ?m geo:hasGeometry ?g .
8   ?g geo:asWKT ?gWKT .
9  FILTER (geof:sfWithin(?gWKT,
10 "POLYGON((-77.2 38.8, -77 38.8, -77 39, -77.2 39.9, -77.2 38.8))
      "^^sf:wktLiteral))
```

```
11 }
```

3.2.3.2 stSPARQL

stSPARQL [38] est un langage d'interrogation pour l'interrogation des données géospatiales du triplestore Strabon. Afin de gérer les informations géospatiales, le modèle stRDF, une extension de RDF, est utilisée. Dans ce modèle, les types de données géospatiales *strdf:WKT* et *strdf:GML* sont introduits pour représenter des géométries sérialisés à l'aide des deux normes WKT et GML (Figure 3.13).

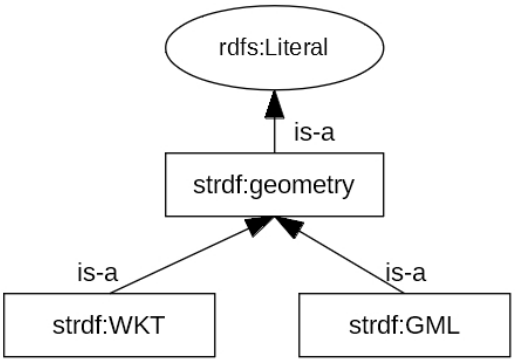


Figure 3.13 – Le modèle stRDF [Kyzirakos *et al.*, 2012].

Le langage de requête stSPARQL étend SPARQL 1.1 avec le modèle stRDF. Dans ce langage, les fonctions SQL peuvent être utilisées dans les requêtes SPARQL grâce aux URIs prédéfinis, par exemple, *strdf:buffer*. De même manière, une fonction booléenne d'extension de SPARQL a été définie pour chaque relation topologique, par exemple *strdf:within*) en trois modèles : OGC Simple Features, 9IM et RCC8. Ainsi, stSPARQL supporte plusieurs familles de relations topologiques et peut exprimer des sélections ou jointure spatiales. Ce langage prend aussi en charge les opérations de mise à jour comme l'insertion, la suppression ou la mise à jour conforme à SPARQL Update 1.1.

Le code ci-dessous (Code 3.9) est un exemple d'une requête stSPARQL qui a pour but de trouver les monuments se situant dans un polygone avec une fonction de filtrage.

Code 3.9 – Recherche des monuments se situant dans un polygone avec une fonction de filtrage

```
1 PREFIX sf: http://www.opengis.net/def/sf/
2 PREFIX ex: <http://example.org/exampleOntology/>
3 SELECT ?m
4 WHERE
```

38. http://www.strabon.di.uoa.gr/stSPARQL

```
5  {
6   ?m geo:hasGeometry ?g .
7   ?g geo:asWKT ?gWKT .
8  FILTER (strdf:within(?gWKT,
9  "POLYGON((-77.2 38.8, -77 38.8, -77 39, -77.2 39.9, -77.2 38.8))
       "^^strdf:WKT))
10 }
```

3.2.3.3 Synthèse

Les deux langages d'interrogation stSPARQL et GeoSPARQL ont été développés indépendamment à peu près au même moment. Comme revendiqué dans [Kyzirakos *et al.*, 2012], stSPARQL et GeoSPARQL ne peuvent être comparés en fonction de leur pouvoir de représentation. Ils partagent la même hypothèse de base que le littéral d'un type de données doit être utilisé pour coder toutes les informations d'une géométrie. Les types de données utilisés dans ces deux langages sont exactement les mêmes même si leur nom sont légèrement différents. Ils permettent la sérialisation de géométries selon les normes de l'OGC. En ce qui concerne les fonctions de manipulation de géométries, stSPARQL et GeoSPARQL ont presque la même puissance expressive car ils comportent le même ensemble de fonctions.

Pourtant, stSPARQL comporte certaines limites par rapport à GeoSPARQL. En effet, il ne supporte pas les relations topologiques binaires à utiliser comme propriétés RDF. En ce qui concerne la représentation de caractéristiques et de géométries, GeoSPARQL impose une ontologie en RDFS qui représente bien la terminologie de l'OGC, alors que, à l'exception des types de données pertinentes, stSPARQL ne propose rien d'autre en termes de vocabulaire pour la modélisation. Pourtant, étant une extension de SPARQL 1.1, stSPARQL dépasse GeoSPARQL en offrant des fonctions d'agrégats géospatiales et de mise à jour.

Conclusion du chapitre

Nous avons présenté dans ce chapitre les différentes technologies utilisées dans le Web sémantique. Elles nous donnent des connaissances de base pour développer des ontologies qui visent à modéliser les données contenues dans les sources hétérogènes et à les intégrer.

Ensuite, nous avons étudié les approches et outils de traduction permettant la mise en correspondance entre les données relationnelles et les données en RDF. La plateforme D2RQ a été choisie pour le peuplement de notre ontologie par sa simplicité et le support de deux types de traduction : la matérialisation et la traduction à la demande (grâce à son extension D2R).

Les dernières parties sont consacrées à la présentation des triplestores géospatiaux «open-source» et les langages permettant l'interrogation de données spatiales stockées dans ces triplestores. D'après les études comparatives, Strabon avec son langage d'interrogation stSPARQL est le meilleur candidat pour la gestion et le stockage de données géospatiales en RDF dans le contexte d'intégration de donnée par l'approche d'entrepôt. Effectivement, il offre la capacité d'effectuer des agrégats géospatiaux et surtout de réaliser des requêtes de mise à jour permettant d'ajouter de nouvelles déclarations à la base de connaissances.

Chapitre 4

Modélisation et raisonnement spatio-temporel

Sommaire

4.1 Le temps 62
4.1.1 Représentation du temps 62
4.1.2 Représentation du changement 62
4.1.3 Raisonnement temporel 64
4.1.3.1 Algèbre de l'intervalle 64
4.1.3.2 Algèbre de l'instant 64
4.1.3.3 Algèbre de l'instant et de l'intervalle 65
4.1.4 Synthèse 66
4.2 Modélisation et raisonnement temporel dans le Web sémantique 66
4.2.1 Ontologies du temps 67
4.2.1.1 OWL-Time 67
4.2.1.2 SWRL Temporal Ontology 68
4.2.2 Représentation des changements 68
4.2.2.1 Réification de RDF 69
4.2.2.2 Réification des relations N-aire 70
4.2.2.3 Versionnement 71
4.2.2.4 4D-fluent 71
4.2.3 Raisonnement temporel 72
4.2.4 Synthèse 72
4.3 L'espace 73
4.3.1 Représentation de l'espace 73
4.3.2 Modèles de représentation 74
4.3.2.1 Modèle «raster» 74
4.3.2.2 Modèle vectoriel 75
4.3.3 Modèles de données spatiales 76
4.3.3.1 OpenGIS Simple Features Specification For SQL 76
4.3.3.2 WKT 77

4.3.3.3 GML 78
4.3.4 Raisonnement spatial 80
4.3.4.1 Modèle RCC 81
4.3.4.2 Modèle 9-IM 82
4.3.5 Synthèse 84
4.4 Modélisation et raisonnement spatial dans le Web sémantique 85
4.4.1 Ontologies de l'espace 85
4.4.1.1 Basic Geo Vocabulary 85
4.4.1.2 GeoRSS 86
4.4.1.3 GeoOWL 88
4.4.1.4 Transformation d'un modèle en ontologie . . 88
4.4.2 Raisonnement spatial 89
4.4.2.1 Architecture hybride pour le raisonnement spatial 89
4.4.2.2 Système de raisonnement PelletSpatial . . . 90
4.4.2.3 Les règles SWRL spatiales 91
4.4.2.4 Raisonnement spatial par triplestores géospatiaux 92
4.4.3 Synthèse 93
4.4.4 Modélisation et raisonnement spatio-temporels dans le Web sémantique 93
4.4.4.1 Le modèle SOWL 95
4.4.4.2 Le modèle Continuum 96
4.4.4.3 Synthèse 97

Introduction du chapitre

Un nombre important d'études ont été menées pour la modélisation et représentation des informations géographiques dans toutes leurs dimensions spatiales et temporelles. Les travaux de Peuquet [Peuquet, 1994] ainsi que ceux de Yuan [Yuan, 1999] mettent en exergue l'importance que revêt une gestion appropriée du temps, non pas comme un simple attribut de l'espace géographique mais comme une dimension à part entière dans un système d'information spatio-temporel. Ces travaux soulignent qu'un tel système doit supporter la représentation simultanée de trois domaines (ou dimensions) : la dimension spatiale, la dimension temporelle et la dimension thématique.

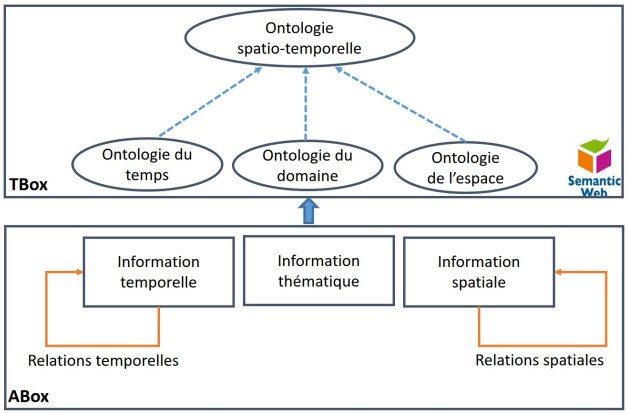


Figure 4.1 – Le temps, l'espace et le spatio-temporel.

Tout comme la modélisation classique, la modélisation de données spatio-temporelles par l'approche ontologique nécessite aussi la prise en compte de ces trois dimensions (Figure 4.1). Pour ces raisons, en vue de construire des ontologies modélisant les données environnementales et leurs relations spatio-temporelles, nous présentons tout d'abord dans ce chapitre les différents aspects de la modélisation et du raisonnement temporel et spatial par les approches classiques. Ce qui donne des éléments de base pour aboutir à la modélisation et au raisonnement spatio-temporels appliqués au Web sémantique décrits dans la suite. Ceux-ci s'appuient sur les ontologies de temps et les ontologies de l'espace.

4.1 Le temps

Le temps est l'objet d'études de plusieurs domaines, tels que la philosophie, l'analyse du langage naturel, les bases de données temporelles, ou encore l'intelligence artificielle. Dans cette section nous abordons les travaux développés en intelligence artificielle qui s'intéressent à la représentation du temps et aux logiques temporelles.

4.1.1 Représentation du temps

Le temps peut être représenté par une unité sans durée, appelé instant, qui peut être assimilé à un point sur une droite, ou par une unité ayant une durée, appelé intervalle, assimilable à un segment de droite [C. Bessière et Schwer, 1997]. Un intervalle est alors défini soit par la date de début et sa durée, soit par les dates de début et de fin. Dès lors, un phénomène ou un événement du monde réel peut être enregistré par un intervalle de validité qui est borné par deux instants.

La norme internationale ISO 8601 [1] a été introduite pour la description du temps afin d'éviter tout risque de confusion dans les communications internationales. Elle est construite sur quatre concepts fondamentaux :

— **Instant** : Un instant définit un unité de temps indivisible. Il est spécifié par sa position sur la droite du temps.

— **Intervalle** : Un intervalle représente la durée entre deux instants donnés, nommés début et fin. Il peut être spécifié par les deux instants de début et de fin ou par l'un d'entre eux et la durée entre les deux.

— **Intervalle répétitif** : Un intervalle répétitif représente une série d'intervalles de temps consécutifs ayant la même durée.

— **Durée** : Une durée représente une quantité de temps.

4.1.2 Représentation du changement

La représentation du temps est influencée par le phénomène à modéliser. De manière conceptuelle, l'objectif fondamental de toutes les bases de données temporelles est d'enregistrer ou de dépeindre les changements au fil du temps. Ceux-ci sont décrits comme un événement ou une succession d'événements qui produisent un changement de l'état, c'est-à-dire, la position et/ou la sémantique de l'entité, par exemple, un changement de culture suite à la récolte ou un changement de la forme d'une parcelle à cause d'une fusion de parcelles.

Afin de pouvoir décrire l'évolution des entités suite aux changements et aussi les événements qui se sont produits, les philosophes ont établi une distinction entre deux paradigmes visant à identifier les entités à travers le temps : l'enduratisme et le perdurantisme.

1. https://www.iso.org/iso-8601-date-and-time-format.html

§1 Endurantisme

L'approche endurantiste, également appelée tridimensionnel ou 3D, est appliquée dans la modélisation Entité-Association ou Orienté Objet. Elle suppose que les entités ont trois dimensions et existent en totalité à chaque moment de leur vie [Hales et Johnson, 2003]. Ainsi, ces entités, appelés *continuants* ou *endurants*, n'ont pas de dimension temporelle. L'approche suppose que l'objet existe en permanence à travers le temps et peut être ainsi toujours identifié à chaque point dans le temps.

Dans ce paradigme, il existe une distinction fondamentale entre les entités et les changements. Dans le cas d'une liaison à une dimension temporelle, les entités doivent être indexées de manière séparée et les références temporelles sont ajoutées sur chaque couche du SIG. Par conséquent, il nécessite moins de capacité de stockage puisque les changements se limitent à un suivi couche par couche. Cependant l'expression de la temporalité et du changement est limitée au cours de sa période d'existence.

Cette approche est largement utilisée dans les domaines tels que la conception des bases de données et le développement des systèmes d'information ou le développement des ontologies [Jarrar *et al.*, 2003].

§2 Perdurantisme

L'approche perdurantiste, aussi appelée quatre dimensions ou 4D, considère que les objets ont quatre dimensions et qu'ils existent donc partiellement qu'à travers le temps. Ainsi, ces objets, appelées *occurents* ou *perdurants*, n'existent que pendant une certaine période de temps au cours de laquelle ils changent de façon continuelle. Chaque période forme une *tranche de temps* (*time slice*) dans leur vie. L'ensemble de ces tranches de temps constitue ainsi la dimension temporelle de l'objet. Cette approche représente donc les différentes propriétés d'une entité dans le temps comme les *fluents* qui ne sont validés que pendant certains intervalles ou à certains instants. Selon ce paradigme, les entités sont généralement considérées comme des *vers de l'espace et du temps*» (*space-time worms*) [Sider, 2001] étant donné qu'ils sont identifiés sur la base de dimensions spatiales et temporelles.

Le principal avantage de l'approche est la simplicité car les entités et les changements sont traités d'une manière similaire. En comparant avec l'endurantisme, le perdurantisme permet des représentations plus riches des phénomènes du monde réel grâce à sa flexibilité et son expressivité [Al-Debei *et al.*, 2012]. Comme ce paradigme fusionne l'espace et le temps au niveau de la primitive spatiale de l'entité, on obtient une expressivité plus fine du changement au sein des objets eux-mêmes. Cependant, il nécessite de grande capacité de stockage puisque chaque entité est individuellement modélisée afin de pouvoir suivre les changements de chacun d'entre eux.

4.1.3 Raisonnement temporel

Les relations entre les objets au fils du temps peuvent être représentées grâce à leurs relations temporelles. Dans la littérature, ces relations sont différemment définies selon l'approche de raisonnement appliquée. Dans cette partie, nous présentons trois formalismes permettant le raisonnement sur les relations temporelles : l'algèbre de l'instant (ou du point), l'algèbre de l'intervalle et l'algèbre de l'instant et de l'intervalle.

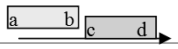
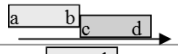
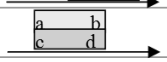
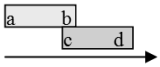
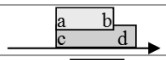
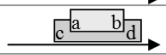
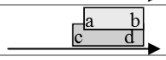
Relation	Contrainte	Schéma	Réciproque
before (avant)	b<c		after (après)
meets (recontre)	b=c		met-by (rencontré par)
equals (égal)	a=c ∧ b=d		
overlaps (chevauche)	d ∧ b		overlapped-by (chevauché par)
starts (débute)	a=c ∧ b<d		started-by (débuté par)
during (pendant)	c ∧ d		contains (englobe)
finishes (termine)	b=d ∧ c<a		finished-by (termine)

Figure 4.2 – Les relations définies dans l'algèbre de l'intervalle.

4.1.3.1 Algèbre de l'intervalle

Allen [Allen, 1983, Allen, 1984, Allen et Ferguson, 1994] propose une algèbre temporelle permettant la définition des relations topologiques entre des intervalles temporels. Comme un intervalle temporel est une période de temps débutant à un certain instant dans le temps et se terminant à un instant ultérieur, il est possible de définir les différentes relations entre les intervalles en se basant sur leurs extrémités. Il existe ainsi 13 relations différentes entre intervalles permettant de représenter toutes les configurations possibles (Figure 4.2).

4.1.3.2 Algèbre de l'instant

Une alternative du raisonnement sur les intervalles de temps est le raisonnement sur les instants (ou les points de temps). L'algèbre de l'instant est introduite par [Vilain *et al.*, 1990] et développée dans la suite par [Ladkin et Reinefeld, 1992]. Cette algèbre est définie de la même manière que l'algèbre de l'intervalle. En effet, les instants sont associés par des relations qui sont composées de relations simples, originellement appelées *precedes*, *same* et *follows* (Figure 4.3).

Relation	Désignation	Schéma	Équivalence
precedes (précède)	<	A precedes B 	before
same (même)	=	A same B 	equals
follows (suit)	>	A follows B 	after

Figure 4.3 – Les relations de base définies dans l'algèbre de point.

L'information entre deux instants peut être représentée comme une disjonction de ces trois relations basiques. De cette façon, l'algèbre de l'instant s'appuie sur sept opérateurs $\{\phi, <, \leq, =, >, \geq, \neq, ?\}$. En outre, elle fournit également l'addition et la multiplication comme opération.

4.1.3.3 Algèbre de l'instant et de l'intervalle

D'une part, l'algèbre de l'instant est inadaptée pour représenter plusieurs phénomènes réels. En effet, les instants ne sont pas suffisants pour exprimer la sémantique du langage naturel et ils sont très inadaptés pour modéliser des événements et des actions naturelles. D'autre part, comme ces deux algèbres ne peuvent être utilisées que pour le raisonnement sur les unités de même type (soit sur les intervalles seuls, soit sur les instants seuls), elles ne sont pas suffisantes non plus pour modéliser des problèmes du monde réel [Krokhin et Jonsson, 2002]. Afin de surmonter ce problème, l'algèbre de l'instant et de l'intervalle a été présentée par [Vilain, 1982] et ensuite développée par [Meiri, 1996, Krokhin et Jonsson, 2002]. Cet algèbre étend les deux algèbres précédentes en ajoutant des relations entre instants et intervalles pour un total de 26 relations. Hors les 13 relations entre les intervalles eux-mêmes et les 3 relations de base entre les instants eux-mêmes, les relations entre les intervalles et les instants sont représentées par la Figure 4.4 :

Relation	Désignation	Contrainte	Schéma	
before (avant)	b	$x<I^-$	x before I	
starts (débute)	s	$x=I^-$	x starts I	
during (pendant)	d	$I^-<x<I^+$	x during I	
finishes (termine)	f	$x=I^+$	x finishes I	
after (après)	a	$x>I^+$	x after I	

Figure 4.4 – Relations entre les intervalles et les instants.

4.1.4 Synthèse

Deux approches visant à décrire l'évolution d'un objet, et aussi les changements, à travers le temps sont étudiées. L'approche perdurantiste est adaptée à la modélisation des données environnementales car elle peut représenter les différentes propriétés d'un objet à travers le temps. En effet, chaque observation ou relevé correspond à une tranche de temps de l'objet étudié. De cette manière, la vie de l'objet se compose d'une ou plusieurs tranches de temps qui sont attachées à des attributs et/ou un emplacement.

Chaque tranche de temps est ainsi formée de trois dimensions : la dimension temporelle, la dimension spatiale et la dimension sémantique. L'emplacement d'un objet est considéré comme un attribut lié à chaque tranche de temps. Lorsque l'objet se déplace, par exemple, le cas des espèces vivantes, ou se déforme, par exemple le cas des territoires, une autre tranche de temps est créée. En conséquence, nous allons étudier les modèles de représentation de données spatiales et le raisonnement sur leurs relations topologiques dans la suite.

Afin de croiser ces informations spatio-temporelles, il est nécessaire d'effectuer le raisonnement sur leurs relations temporelles et spatiales. Pour cela, nous avons étudié les représentations du temps ainsi que les opérateurs pour le raisonnement temporel. L'algèbre de l'intervalle ou l'algèbre de l'instant seules ne sont pas suffisantes pour modéliser les événements et elles ne peuvent être appliquées qu'aux objets de même type, l'algèbre de l'instant et de l'intervalle est le mieux adaptée pour le raisonnement temporel sur les données environnementales. En effet, dans la pratique, une observation ou un relevé peut être identifié soit par un intervalle ou soit par un instant.

Nous venons de présenter différents aspects du temps. Dans les paragraphes suivants, nous allons découvrir comment on peut modéliser la dimension temporelle et raisonner sur les relations temporelles dans le Web sémantique.

4.2 Modélisation et raisonnement temporel dans le Web sémantique

La modélisation de données environnementales nécessite la prise en compte des changements ou de l'évolution des objets. Pour ces raisons, nous allons étudier la modélisation temporelle par l'approche ontologique pour la représentation des objets qui évoluent dans le temps. Nous présentons tout d'abord deux ontologies du temps permettant la représentation des concepts et des relations temporels. Ensuite, les différentes approches visant à incorporer le temps dans le Web sémantique pour décrire l'évolution des entités à travers le temps sont approfondies. Enfin, nous terminons par une étude sur le raisonnement temporel grâce aux technologies du Web sémantique.

4.2.1 Ontologies du temps

Les ontologies du temps visent à représenter des concepts et des relations temporels. Plusieurs ontologies du temps ont fait leur apparition au sein de la communauté. Deux travaux majeurs du domaine sont l'ontologie OWL-Time et le SWRL Temporal Ontology.

4.2.1.1 OWL-Time

Développée au sein du consortium W3C, l'ontologie OWL-Time [2], initialement introduite par [Hobbs et Pan, 2004], repose sur les concepts temporels et les relations temporelles définies dans la théorie d'Allen et bénéficie d'une spécification précise formalisée en OWL. Étant une des premières ontologies pour la représentation temporelle, cette ontologie était tout d'abord destinée à décrire le contenu temporel des pages Web et les propriétés temporelles des services web. Plus tard, elle a été développée en une ontologie du temps de référence pour représenter le temps dans plusieurs domaines, y compris le domaine de la recherche d'information.

L'élément principal de l'ontologie est *TemporalEntity* qui comprend deux sous-classes : *Instant* et *Interval*. Les propriétés *hasBeginning* et *hasEnd* sont utilisées pour définir le début et la fin d'une entité temporelle *Interval*. La classe *Interval* a une sous-classe *ProperInterval* correspondant à un intervalle dont le début et la fin ne sont pas égaux. Quant aux relations temporelles, l'ontologie applique l'algèbre de l'intervalle qui comporte 13 propriétés entre les intervalles. En plus, elle définit la relation *inside* entre un instant et un intervalle.

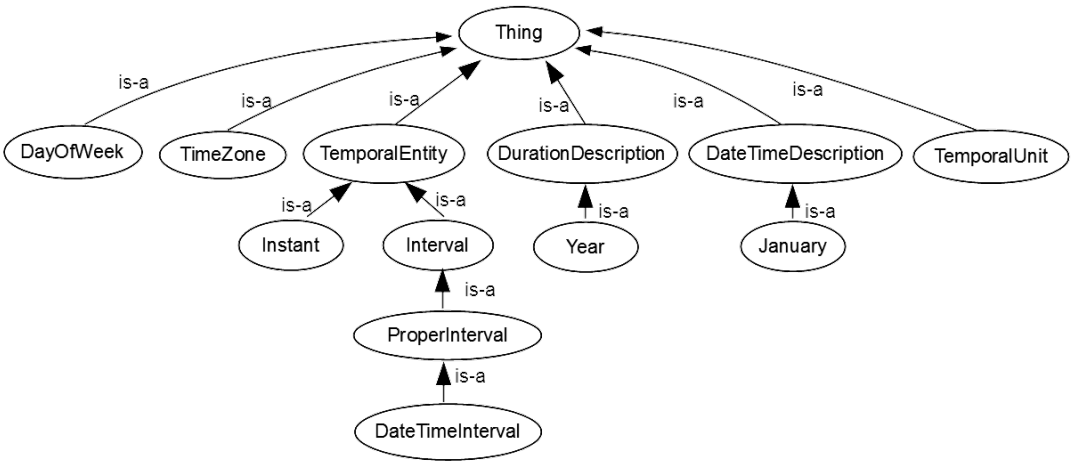


Figure 4.5 – Ontologie OWL-Time.

Une nouvelle version de l'ontologie [3] a été mise en révision. Dans cette ver-

2. https://www.w3.org/2006/time
3. https://www.w3.org/TR/owl-time

sion, de nouvelles classes et propriétés sont introduites. En outre, les relations hiérarchiques entre les concepts sont relâchées en vue d'améliorer la capacité de représentation et de simplifier l'implémentation.

4.2.1.2 SWRL Temporal Ontology

Introduite par [O'Connor et Das, 2010], SWRL Temporal Ontology [4] (Figure 4.6) est utilisée notamment dans les recherches cliniques. Cette ontologie représente le temps de manière similaire à OWL-Time sauf qu'elle ne permet d'établir les relations entre les instants ou les intervalles de temps. La classe de base modélisant l'évolution de l'entité au fil du temps est appelé *Entity*. Elle est associée à la propriété *hasValidTimes* qui détient le temps pendant lequel l'information associée est valide. La valeur de cette propriété est modélisés par une classe, appelée *ValidTime*, qui a deux sous-classes, *ValidInstant* et *ValidInterval*, représentant respectivement les instants et les intervalles de temps.

L'avantage de cette ontologie est qu'elle offre des *built-ins* qui implémentent l'ensemble des 13 opérateurs d'Allen. Ces derniers peuvent être utilisés dans les règles SWRL pour effectuer le raisonnement sur les relations temporelles.

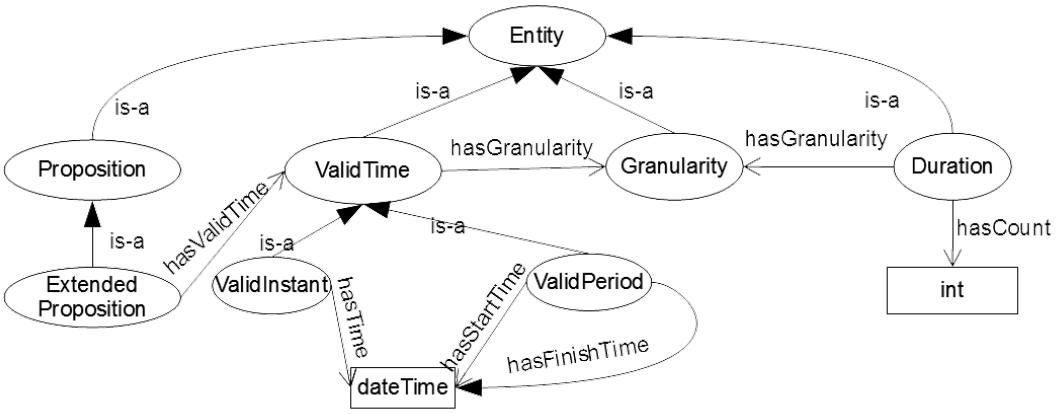


Figure 4.6 – SWRL Temporal Ontology.

4.2.2 Représentation des changements

Les deux ontologies présentées ci-dessus permettent de représenter des concepts et des relations temporels. Toutefois, elles ne s'intéressent qu'à décrire des éléments de données individuels plutôt qu'à construire un modèle temporel pour décrire systématiquement toutes les informations temporelles dans un système. Autrement dit, de telles ontologies ne suffisent pas à modéliser l'évolution des entités. Par ailleurs, comme les deux langages ontologiques du Web sémantique,

4. swrl.stanford.edu/ontologies/built-ins/3.3/temporal.owl

OWL et RDFS, ne permettent que des relations binaires entre les entités, le traitement des informations qui changent au fil du temps devient un problème critique du domaine. Afin de pallier ces problèmes, plusieurs approches ont été proposées. Nous pouvons distinguer les approches qui ne modifient pas les ontologies comme le versionnement, ou les approches qui doivent introduire de nouveaux éléments dans l'ontologie comme les techniques de réification et les modèles des fluents.

Dans cette section, nous présentons les approches visant à modéliser l'évolution des objets spatio-temporels dans ce domaine. Prenons l'exemple de la rotation de culture d'une parcelle dans laquelle existe la relation ternaire entre une parcelle agricole, son exploitation et la période de plantation, exprimée par *hasLandUse(Parcel, LandUse, TimeInterval)*, nous allons examiner comment cette relation est représentée par les approches : Réification de RDF, Réification N-aire, Versionnement et 4D-fluent.

4.2.2.1 Réification de RDF

La réification de RDF est une technique d'usage général permettant la représentation des relations n-aires en utilisant un langage qui ne permet que des relations binaires. Plus précisément, une relation n-aire est représentée comme un objet qui est le sujet de n + 1 déclarations. Les participants de la relation n-aire deviennent l'objet dans les n premières déclarations et le prédicat devient l'objet dans la dernière déclaration. De cette manière, en appliquant la réification de RDF pour la relation ternaire *hasLandUse*, une nouvelle classe nommée *ReifiedRelation* est créée. Un instance de cette classe joue le rôle du sujet des déclarations (Figure 4.7).

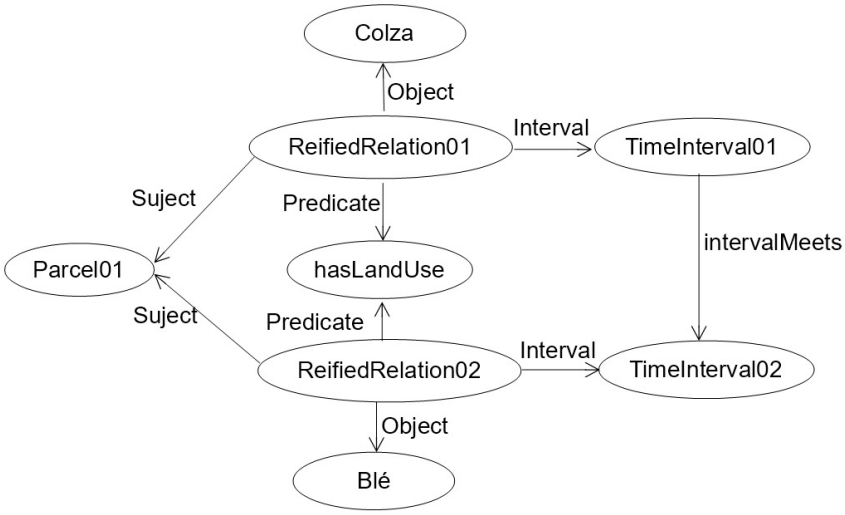


Figure 4.7 – Exemple de la réification de RDF.

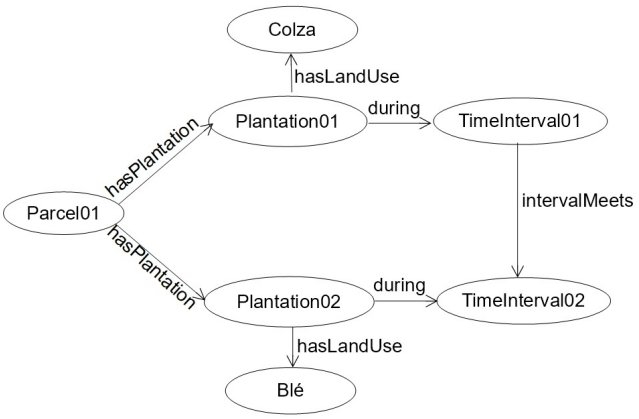


Figure 4.8 – Exemple de la réification des relations N-aire.

Il est visible que cette approche ne traite que des déclarations plutôt que les relations réelles entre les entités. Ainsi, elle souffre principalement de trois inconvénients :

— **Représentation des relations** : L'approche n'est pas sémantiquement naturelle à cause du traitement de ces relations comme déclarations.

— **Redondance d'information** : Une nouvelle classe est créée chaque fois qu'une relation temporelle doit être représentée. Cela augmente la complexité de l'ontologie de manière importante.

— **Capacité de raisonnement** : L'approche ne permet pas le raisonnement sur les relations transitives, symétriques, inverses et fonctionnelles. Par exemple, la relation *isLandUseOf* qui est l'inverse de *hasLandUse* ne peut être représentée puisque cette dernière est réifiée en déclaration.

Comme il est difficile d'utiliser le langage d'interrogation SPARQL sur les déclarations réifiées, l'approche devrait être évitée si possible [Auer *et al.*, 2013].

4.2.2.2 Réification des relations N-aire

Selon le W3C, la réification des relations N-aire [Noy *et al.*, 2006] est la méthode générale pour représenter les prédicats d'arité plus élevée que 2 dans le Web sémantique. En utilisant une forme améliorée de la réification de RDF, l'approche suggère de représenter une relation n-aire comme deux propriétés liées chacune à un nouvel objet plutôt que comme l'objet d'une propriété. Dès lors, chaque instance de cette classe a des relations binaires reliant les entités participantes dans la relation originale. L'information temporelle est donc liée à des instances de la nouvelle classe.

En appliquant cette approche pour la représentation de notre relation ternaire *hasLandUse*, une nouvelle classe, appelée *Plantation*, est créée. Ainsi, nous obtenons de nouvelles déclarations (Figure 4.8).

L'approche comporte des inconvénients, notamment en ce qui concerne la redondance de données. En effet, bien qu'elle ne nécessite qu'un seul objet supplémentaire pour chaque relation temporelle, cette approche souffre aussi de la redondance de données.

4.2.2.3 Versionnement

Dans l'approche de versionnement [Klein et Fensel, 2001], lorsqu'une ontologie est modifiée, une nouvelle version de l'ontologie est créée pour représenter son évolution. La Figure 4.9 décrit deux versions de l'ontologie dont la plus récente est construite suite à un changement de culture.

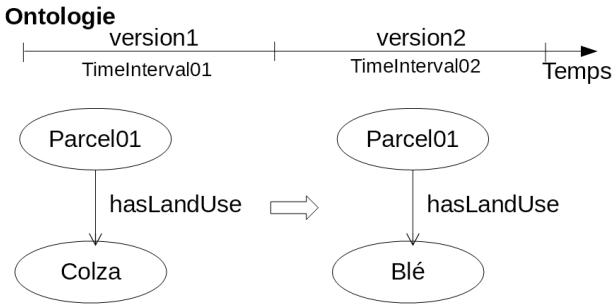


Figure 4.9 – Exemple du versionnement d'ontologie.

La plupart des implémentations adoptent une variété de stratégies d'optimisation afin d'éviter des copies entières de l'ontologie en réponse aux modification de l'ontologie. Cependant, quelles que soient les optimisations adoptées, l'approche possède les inconvénients suivants :

— **Redondance de l'information** : Les changements, même sur un seul attribut exigent la création d'une nouvelle version de l'ontologie.

— **Coût de l'interrogation** : La recherche des événements survenus au cours du temps nécessite des recherches exhaustives dans de multiples versions de l'ontologie.

4.2.2.4 4D-fluent

L'approche 4D-fluent [Welty et Fikes, 2006] est basée sur le perdurantisme. Elle considère que l'existence d'une entité peut être exprimée en plusieurs représentations, chacune correspondant à un intervalle de temps défini, appelée*tranche de temps*. Lorsque la propriété de l'objet change, une nouvelle tranche de temps est établie pour représenter la nouvelle propriété. De cette manière, les changements n'affectent que les propriétés de la partie temporelle de l'ontologie en gardant la partie statique inchangée.

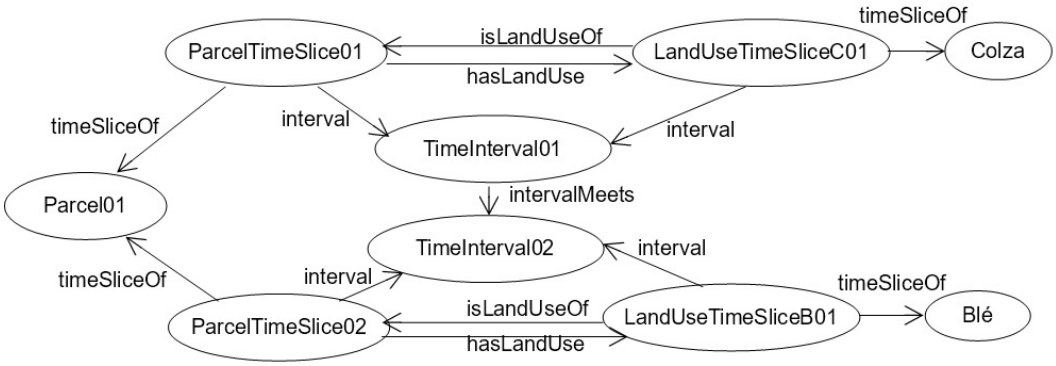


Figure 4.10 – Exemple de l'approche 4D-fluent.

Cette méthode introduit la classe *TimeSlice* qui représente les parties temporelles de l'entité. La classe est liée à *TimeInterval*, une classe du domaine temporel représentant les intervalles de temps, par la propriété *tsTimeInterval*. Chaque *TimeSlice* est connecté à l'entité par la propriété *tsTimeSliceOf*.

En appliquant cette approche pour notre relation ternaire, deux nouvelles classes *ParcelTimeSlice* et *LandUseTimeSlice* sont créées. Ainsi, de nouvelles déclarations sont ajoutées comme illustrées par Figure 4.10.

Cette approche souffre aussi de la prolifération d'objets car elle introduit deux objets supplémentaires pour chaque relation temporelle.

4.2.3 Raisonnement temporel

À notre connaissance, il n'existe pas encore de moteurs d'inférence qui puissent déduire des relations temporelles qualitatives entre les instants et les intervalles décrits en OWL. Comme décrit dans la section 4.1.3, il est possible de raisonner sur ces relations à travers des calculs mathématiques si on dispose de données quantitatives ou à l'aide d'algèbre d'intervalles si on dispose d'un ensemble de relations qualitatives.

Pour cela, ces relations peuvent être exprimées par un ensemble de règles. Dès lors, il est possible d'utiliser un raisonneur comme Pellet si elles sont exprimées en SWRL ou ou de représenter ces règles sous forme des requêtes SPARQL comme présentées dans 3.1.3.2.

4.2.4 Synthèse

Dans cette partie, deux ontologies OWL Time et SWRL Temporal Ontology ont été étudiées pour la représentation des concepts et des relations temporels. Étant une recommandation du W3C, la première est utilisée dans de nombreuses recherches. De plus, elle définit les relations temporelles entre les intervalles définies dans l'algèbre d'Allen et des relations entre les instants et les intervalles.

Bien que ces dernières soient nécessaires pour le croisement de données dans notre contexte, elles sont absentes dans la deuxième ontologie. Pour ces raisons, OWL Time est le meilleur candidat pour représenter la dimension temporelle de données environnementales.

Ensuite, plusieurs approches visant à représenter les changements dans le Web sémantique sont détaillés, parmi lesquelles l'approche N-aire et l'approche 4D-fluent, suivant le perdurandisme qui sont les plus utilisées. En effet, elles sont appliquées dans les ontologies spatio-temporelles présentées à la fin du chapitre. Le point commun de ces approches c'est l'introduction d'une nouvelle classe décrivant des tranches de temps dans la vie de l'objet, cela entraîne inévitablement la redondance des données.

En ce qui concerne le raisonnement temporel dans le Web sémantique, il est nécessaire d'utiliser des règles SWRL ou des requêtes SPARQL pour représenter des relations temporelles. Comme discuté dans le chapitre précédent, selon nous, l'utilisation des requêtes SPARQL comme langage de règles est l'approche la plus adaptée dans le contexte de l'intégration sémantique de données par l'approche entrepôt. De cette manière, il est possible d'ajouter directement les relations temporelles entre les objets à la base de connaissances à l'aide de requêtes basiques, sans besoin d'un raisonneur (dans notre cas, c'est Pellet avec son *built-in* temporel).

4.3 L'espace

Cette section introduit les concepts et modèles principaux permettant la représentation de l'espace géographique. Ensuite les modèles pour le raisonnement sur les relations topologiques sont décrits.

4.3.1 Représentation de l'espace

L'espace peut être vu comme un ensemble dont les éléments sont appelés les points. Les ensembles finis de points, c'est-à-dire les sous-ensembles de l'espace, peuvent être des points, des lignes ou des régions. Dans la pratique, les applications spatio-temporelles actuelles modélisent l'espace comme un sous-ensemble de R^3. Z^3, Z^2, Z, R^2 et R sont les sous-ensembles les plus utilisés.

Les objets dans le monde réel ont une position dans l'espace. Dans des environnements d'application spécifiques, la position des objets dans l'espace est importante et ces objets sont appelés objets spatiaux. Par exemple, une parcelle dans un système cadastral. Un système de coordonnées est utilisé afin de représenter leurs positions et les retrouver dans l'espace. Ce système permet de définir des positions sur la surface de la Terre grâce à un couple de coordonnées géographiques. Principalement, deux types de systèmes de coordonnées sont utilisés : les systèmes géographiques et les systèmes projetés.

— **Système de coordonnées projetées** : Un système de coordonnées projetées se définit sur une surface plane à deux dimensions. Dans ce système, les positions sont identifiées par des coordonnées x et y sur une grille dont l'origine est située au centre. Les coordonnées x et y la situent par rapport à la position centrale. L'une précise sa position horizontale et l'autre sa position verticale. La projection Lambert [5] utilisée officiellement pour les cartes de France Métropolitaine est un exemple de ce système.

— **Système de coordonnées géographiques** : Un système de coordonnées géographiques utilise une surface sphérique à trois dimensions pour définir des positions sur la Terre. Dans ce système, une position est référencée d'après ses valeurs de longitude et de latitude. Ces valeurs correspondent aux angles mesurés depuis le centre de la Terre vers un point de surface. Les angles sont souvent mesurés en degrés. Le système géographique le plus répandu est le système WGS84 [6]. Il est utilisé notamment dans le système de positionnement par GPS [7].

Toutes données géographiques doivent être considérées avec leur système de coordonnées associé car ce système impacte directement la mise en correspondance spatiale de ces données ainsi que les calculs géométriques tels que la distance ou la superficie. Dans le contexte d'intégration de données spatio-temporelles hétérogènes, il est nécessaire de prendre en compte ces systèmes car la différence dans le système de coordonnées utilisés dans les données sources posera des problèmes d'hétérogénéité sémantique.

4.3.2 Modèles de représentation

Dans la littérature, parmi les modèles permettant la représentation et le raisonnement sur l'espace géographique, les modèles de représentation les plus connus sont le modèle «raster» (ou matriciel) et vectoriel.

4.3.2.1 Modèle «raster»

Dans ce modèle, une surface est représentée par un ensemble de cellules (ou pixels) organisées en grille où chaque cellule est définie par sa position. A chaque cellule est attribuée une valeur qui peut correspondre à une mesure, par exemple, la pollution ou l'altitude, à une catégorie, par exemple, type de végétation, ou à l'identifiant d'un objet, par exemple, le numéro d'une route. La résolution dépend de la taille de la cellule, plus la cellule est grande, moins l'information est précise, plus la grille est petite, plus la base de données est grande parce qu'il y a plus de détails.

5. http://geodesie.ign.fr/index.php ?page=rgf93
6. Word Geodetic System 1984 : http://spatialreference.org/ref/epsg/wgs-84/
7. Global Positioning System-Système mondial de positionnement

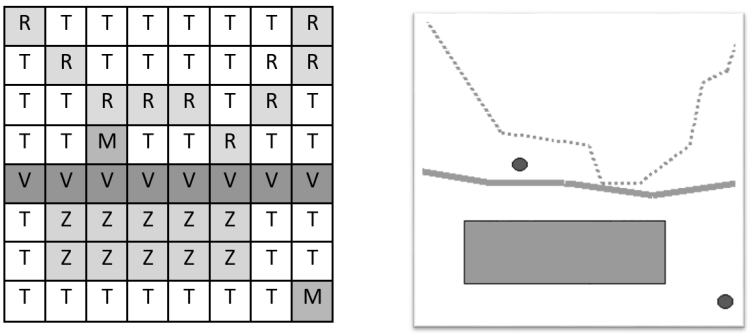


Figure 4.11 – Modèle «raster» et modèle vectoriel.

Un objet du monde réel n'est pas explicitement décrit, seule la cellule de la grille existe dans le système. Il faut chercher toutes les cellules portant le code correspondant à l'objet pour le retrouver.

Ci-dessous sont les avantages qu'offre ce modèle :

— Une structure de données simple ; une matrice de cellules représentant des coordonnées et parfois reliée à une table attributaire.

— Un format puissant pour l'analyse statistique et spatiale sophistiquée.

— Possibilité de représenter des surfaces continues et d'effectuer des analyses de surface.

— Possibilité de stocker de manière uniforme les objets spatiaux.

— Possibilité d'effectuer des opérations de superposition rapides avec des données complexes.

4.3.2.2 Modèle vectoriel

Le modèle décrit les objets spatiaux à l'aide de formes géométriques exprimant leur contour qui se traduit numériquement par des paires de coordonnées (x,y) ou des triplets (x,y,z). Le mode vectoriel correspond à une vue discrète du monde, constitué d'entités distinctes, contrairement au mode «raster» qui correspond à un modèle continu (Figure 4.11).

Le modèle offre des avantages suivants :

— Un stockage plus efficace en ne stockant que les données pertinentes et non l'espace géographique entier.

— Facilité dans la représentation des formes complexes ou linéaires.

— Précisions spatiales car il n'est pas imposé par les dimensions des cellules comme le modèle «raster».

4.3.3 Modèles de données spatiales

La compréhension et l'interprétation de toute donnée spatiale dépendent toujours du modèle de données spatiales dans lequel les données sont exprimées. Il permet la représentation des caractéristiques géographiques et est utilisé comme une base de modélisation pour des SIG ou encore comme un format d'échange d'informations géographiques sur internet. Dans cette section, nous présentons les modèles de données spatiales standards proposés par l'OGC. Ces derniers donne des connaissances de base sur les concepts et relations spatiaux et sont à l'origine des ontologies spatiales.

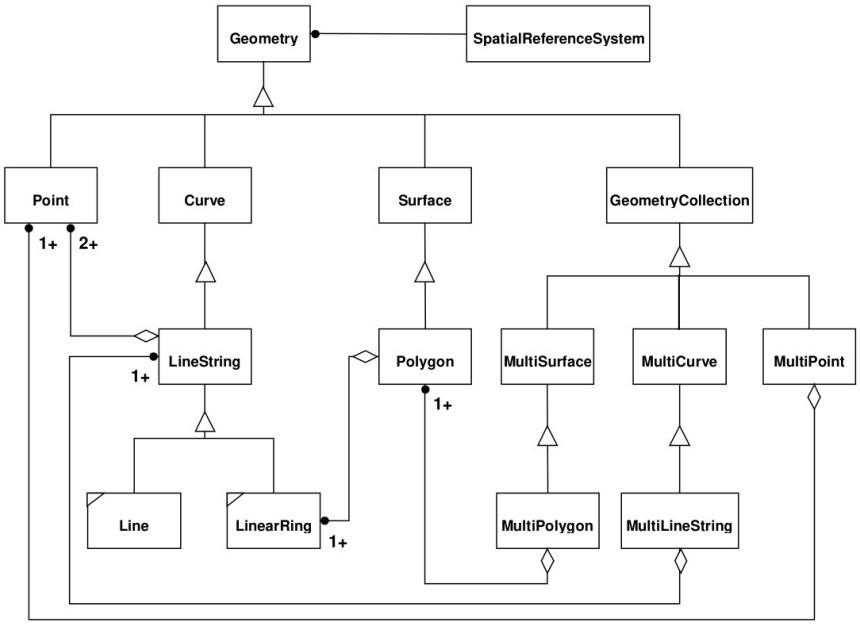


Figure 4.12 – Diagramme de classe de l'OpenGIS SFS for SQL [Beddoe *et al.*, 1999].

4.3.3.1 OpenGIS Simple Features Specification For SQL

La spécification «OpenGIS Simple Features Specification For SQL» [Beddoe *et al.*, 1999] définit un modèle de données SQL standard pour prendre en charge le stockage, la manipulation et la mise à jour des collections d'éléments avec une composante spatiale simple. Ce modèle constitue la brique de base pour des modèles de données spatiaux et sont spécifiés par des attributs spatiaux et non spatiaux.

La Figure 4.12 représente le diagramme de classe du modèle. La classe racine abstraite de la hiérarchie *Geometry* a les sous-classes concrètes : *Point*, *Curve*, *Surface*, *GeometryCollection*. Chaque classe est associée à un système de référence spatial *SpatialReferenceSystem*. Il existe des classes de collections

d'éléments comme *MultiPoint* pour les objets de dimension 0, *MultiLineString* pour les objets de dimension 1 et *MultiPolygon* pour les objets de dimension 2. Les classes *MultiCurve* et *MultiSurface* sont présentées comme des super-classes d'interface pour manipuler des collections de courbes et de surfaces.

Dans ce modèle, on propose un ensemble d'opérateurs topologiques basé sur le modèle de DE-9IM (voir 4.3.4.2) qui permettent de déterminer la relation spatiale entre deux objets spatiaux.

4.3.3.2 WKT

WKT [8], décrite dans la spécification «Simple Feature Access - Part 1: Common Architecture» [9], est une norme de l'OGC largement utilisée pour représenter les géométries. Elle est utilisée pour représenter les géométries dans des systèmes de coordonnées de référence et permet des transformations entre ces systèmes.

La Figure 4.13 illustre la hiérarchie des classes de géométries simples proposées dans WKT. La classe *Geometry* comporte des sous-classes *Point*, *Curve*, *Surface* et *GeometryCollection*. Cette dernière est spécialisée en classes de géométries de dimension 0, 1 et 2 nommées *MultiPoint*, *MultiLineString* et *MultiPolygon* respectivement. Chaque géométrie est liée à un système de référence de coordonnées spécifique et éventuellement à un système de référence de mesure.

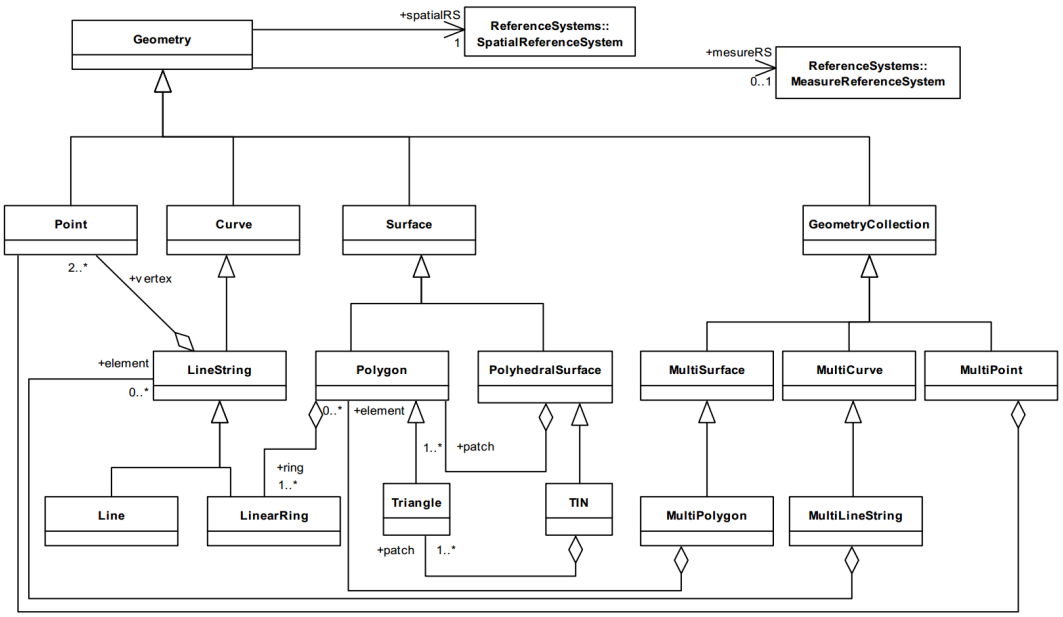


Figure 4.13 – Les classes des géométries introduites dans WKT.

Les coordonnées peuvent être en deux dimensions (x, y) ou en trois dimensions

8. Well-Known Text : http://www.opengeospatial.org/standards/wkt-crs
9. http://www.opengeospatial.org/standards/sfa

(x, y, z). Quelle que soit la dimension choisie, une mesure additionnelle m peut être utilisée (x, y, m) ou (x, y, z, m). Par exemple, le point M: POINT(48.85, 2.35, 15) est utilisé pour représenter la température de 15 degré à Paris dont la latitude est de 48.85 degré et la longitude est de 2.35 degré.

Les géométries WKT sont utilisées dans les spécifications de l'OGC et sont présentes dans des applications qui mettent en œuvre ces spécifications. Par exemple, PostGIS ou Oracle Spatial contiennent des fonctions qui permettent de convertir les géométries vers et à partir d'un format WKT, ce qui les rend lisibles par l'homme.

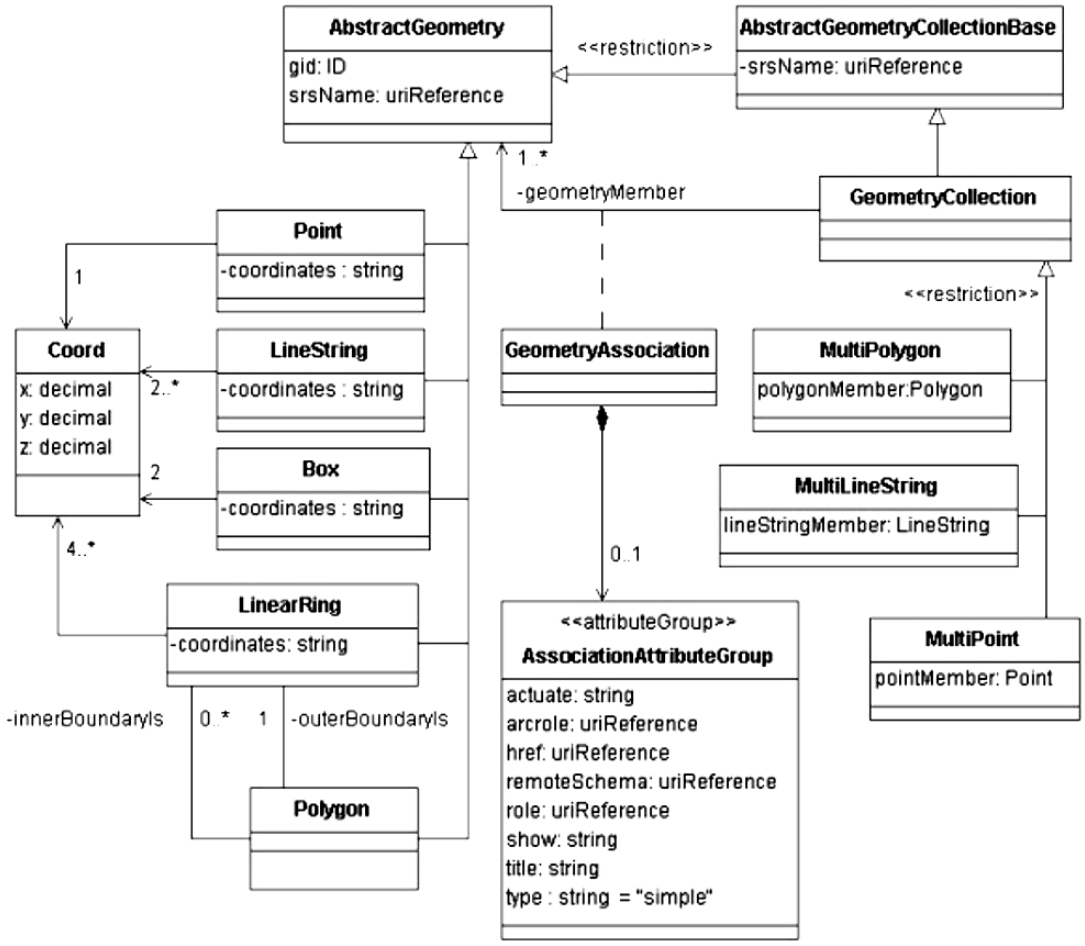


Figure 4.14 – Représentation UML du schéma pour GML.

4.3.3.3 GML

GML [10] est la norme de codage la plus commune basé sur XML pour la représentation des données géospatiales. Développée par l'OGC, elle a pour but

10. Geography Markup Language :http://www.opengeospatial.org/standards/gml

de garantir l'interopérabilité de données dans le domaine de l'information géographique et de la géomatique. Elle fournit des schémas XML pour définir une variété de concepts géographiques, tels que les caractéristiques géographiques, la géométrie, les systèmes de coordonnées de référence, la topologie, le temps et les unités de mesure.

GML fournit des éléments géométriques pour les géométries : *Point*, *LineString*, *LinearRing*, *Polygon*, et les collections de ces éléments, respectivement, *MultiPoint*, *MultiLineString*, *MultiPolygon* et *MultiGeometry* (Figure 4.14). Chacun de ces éléments est décrit dans un schéma GML qui spécifie les normes de notation et de nommage. Par exemple, la définition d'une point devra respecter le schéma (Code 4.1) ci-dessous.

Code 4.1 – Définition d'un point par GML

```
<complexType name="PointType">
 <complexContent>
  <extension base="gml:AbstractGeometricPrimitiveType">
   <sequence>
    <choice>
     <element ref="gml:pos"/>
     <element ref="gml:coordinates"/>
   </choice>
  </sequence>
 </extension>
</complexContent>
</complexType>
```

Les coordonnées de ces géométries sont encodées au moyen des éléments *coordinates*, *posList* et *pos*. Les éléments géométriques possèdent, outre un identifiant unique éventuel, un attribut *srsName* contenant le nom du système de référence utilisé. Le langage GML ne spécifie pas de système de référence par défaut, ce qui implique de le spécifier au moyen de cet attribut dont la valeur doit être une URI vers la définition du système de référence.

Ce langage réalise une distinction entre les caractéristiques géographiques et les objets géographiques. Les objets géographiques peuvent être des points, des lignes, des polygones ou d'autres types de données plus complexes, alors que les caractéristiques géographiques réfèrent aux entités du monde réel telles que des routes, des forêts, des lacs, etc... Les objets géographiques définissent des localisations ou des régions correspondant aux caractéristiques géographiques.

A ses débuts, le standard GML ne permettait que la représentation de primitives géographiques simples comme les lignes, les points, ou les polygones. À partir de la version 3.0, il autorise la prise en compte de primitives plus complexes comme les courbes, les surfaces ou les grilles raster, des éléments temporels et la topologie.

Il existe de nombreux travaux qui proposent des ontologies de l'espace basés

sur ce modèle comme GeoRSS ou encore GeoOWL (voir 4.4.1).

4.3.4 Raisonnement spatial

Une fois la représentation de l'espace établie, il est souhaitable de raisonner sur les positions relatives des entités qui le constituent. Il existe trois grandes catégories de relations spatiales [Egenhofer, 1989] : les relations métriques, les relations topologiques et les relations d'ordre. Dans ce travail, nous nous intéressons particulièrement aux relations topologiques.

Du point de vue de l'espace géographique, la topologie est l'ensemble des relations perçues permettant de situer les objets les uns par rapport aux autres. Les relations topologiques font l'objet d'une abondante littérature scientifique qui utilise de nombreux modèles mathématiques. Les modèles dominant sont le modèle 9-IM (9-Intersection Model) et le modèle RCC (Region Connection Calculus).

Les informations topologiques sont utiles pour détecter et corriger les erreurs de numérisation, par exemple deux lignes sur une couche vectorielle de routes qui ne se croisent pas parfaitement à une intersection. Elle est aussi nécessaire pour effectuer certains types d'analyse spatiale comme l'analyse de réseau.

Comme la dénomination des relations topologiques varie quelque peu selon les modèles de la littérature, nous adaptons la dénomination de l'OGC qui définit 6 prédicats topologiques: *Disjoint, Touches, Contains, Within, Equal, Overlaps*. Ces relations avec les relations correspondantes dans le modèles RCC-8 que nous allons présenter dans les paragraphes suivants sont illustrées par la Figure 4.15.

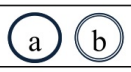
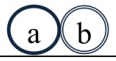
OGC relation	RCC8 relation	Schema	
disjoint	disconnected [DC]	DC(a,b)	
touches	externally connected [EC]	EC(a,b)	
overlaps	partially overlapping [PO]	PO (a,b)	
equal	equal [EQ]	EQ(a,b)	
within	tangential proper part [TPP]	TPP(a,b)	
within	non-tangential proper part [nTPP]	nTPP(a,b)	
contains	tangential proper part inverse [TPPi]	TPPi(a,b)	
contains	non-tangential proper part inverse [nTPPi]	nTPPi(a,b)	

Figure 4.15 – Les relations topologiques définies par l'OGC.

4.3.4.1 Modèle RCC

Le modèle est fondé sur l'approche par régions et s'inspirent des travaux de Clarke [Clarke, 1981]. Il décrit en logique du premier ordre des relations spatiales entre des entités dont les primitives sont des régions. Ainsi, il utilise une relation primitive C(x,y) spécifiant un lien de connexion entre deux régions x et y. Cette relation est réflexive et symétrique et peut être interprétée comme le fait que la frontière délimitant la région x partage au moins un point avec la frontière de la région y. Ainsi, la primitive C(x,y) peut être utilisée pour définir huit relations élémentaires connu sous le nom de RCC-8 [Randell *et al.*, 1992].

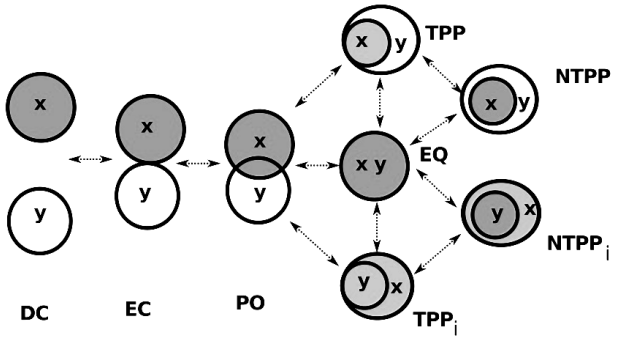


Figure 4.16 – Les relations de l'algèbre RCC-8 et leurs transitions.

Le modèle RCC-8 ne s'applique que sur des polygones et possède son propre vocabulaire pour définir les relations. Dans ce modèle, les relations topologiques sont définies comme suit (Figure 4.16) :

- **DisConnected (DC)** : x est déconnecté de y.
- **Externally Connected (EC)** : x est connecté à y par sa frontière.
- **Equal (EQ)** : x et y sont identiques.
- **Partial Overlaps (PO)** : x et y ont une partie commune.
- **Tangential Proper Part (TPP)** : x est une partie tangentielle de y.
- **Non Tangential Proper Part (NTPP)** : x est une partie non tangentielle de y.
- **Tangential Proper Part inverse (TPPi)** : y est une partie tangentielle de x.
- **Non Tangential Proper Part inverse (NTPPi)** : y est une partie non tangentielle de x.

En plus de RCC-8 et de ces huit relations élémentaires, il existe aussi des algèbres simplifiées ou étendues comprenant une, deux, trois, cinq, quinze et vingt-trois relations, notées respectivement: RCC-1, RCC-2, RCC-3, RCC-5, RCC-15 et RCC-23. Elles sont toutes construites à partir de huit relations de base définies

dans le RCC-8. Par exemple, le RCC-5 renonce à la distinction entre la frontière et l'intérieur d'une région et considère, en conséquence, égales les relations DC et EC ainsi que les relations TPP et NTPP.

En général, tous les modèles RCC ont en commun le fait que les relations qu'ils modélisent sont axiomatisées et définies en utilisant la logique du premier ordre qui leur fournit une sémantique formelle. Cependant, les procédures de décision basées sur cette formalisation ne sont pas très efficaces [Renz, 2002]. Il est à remarquer que les huit relations dans RCC-8 sont identiques à celles définies par le modèle des 9-IM, avec une correspondance un par un entre les relations [Cui *et al.*, 1993].

4.3.4.2 Modèle 9-IM

Un autre modèle pour représenter les relations spatiales est le 9-IM [Egenhofer et Franzosa, 1991]. Le modèle permet de caractériser différentes manières suivant lesquelles deux objets peuvent entrer en relation topologique. Ce modèle est basé sur la théorie des ensembles de points, et repose sur l'usage de trois primitives appliquées à cet ensemble. Supposons qu'un ensemble de points A placé dans un espace X, il existe les primitives suivantes : l'intérieur (A^o), l'extérieur (A^-) et la frontière (∂A) comme illustrés par la Figure 4.17.

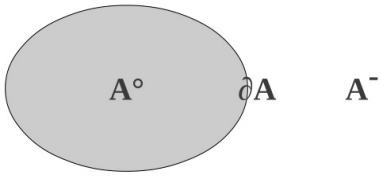


Figure 4.17 – Décomposition d'une région en trois ensemble de points : l'intérieur (A^o), l'extérieur (A^-) et la frontière (∂A).

Ainsi, la relation topologique, notée R, existante entre les deux objets A, B se base sur la comparaison entre les trois ensembles de points de A (A^o), A^-) et ∂A) et B (B^o), B^-) et ∂B). Ceci résulte en neuf comparaisons représentées sous forme d'une matrice 3 x 3 (Figure 4.18) :

$$R(A,B) = \begin{pmatrix} A^o \bigcap B^o & A^o \bigcap \partial B & A^o \bigcap B^- \\ \partial A \bigcap B^o & \partial A \bigcap \partial B & \partial A \bigcap B^- \\ A^- \bigcap B^o & A^- \bigcap \partial B & A^- \bigcap B^- \end{pmatrix}$$

Figure 4.18 – Matrice de comparaison du modèle des 9-IM.

Les valeurs de la matrice sont le vide, noté $\emptyset$ ou bien le non vide, noté $\neg\emptyset$, suivant que les ensembles de points sont intersectés ou non. Comme chaque cellule

de la matrice peut prendre une de ces deux valeurs, il existe alors 2^9 configurations possibles et donc autant de relations spatiales potentielles. Néanmoins, dans le cas de régions spatiales simples, comme tous les régions sont considérées sans trous, cet ensemble peut être ramené à huit possibilités correspondantes aux relations introduit dans RCC-8. La Figure 4.19 décrit ces huit relations sous la forme d'une matrice.

DC(A,B)

$$R_1(A,B) = \begin{pmatrix} \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}$$

EC(A,B)

$$R_2(A,B) = \begin{pmatrix} \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}$$

EQ(A,B)

$$R_3(A,B) = \begin{pmatrix} \neg\emptyset & \emptyset & \emptyset \\ \emptyset & \neg\emptyset & \emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix}$$

PO(A,B)

$$R_4(A,B) = \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}$$

TPP(A,B)

$$R_5(A,B) = \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix}$$

TPPi(A,B)

$$R_6(A,B) = \begin{pmatrix} \neg\emptyset & \emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}$$

NTPP(A,B)

$$R_7(A,B) = \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix}$$

NTTPi(A,B)

$$R_8(A,B) = \begin{pmatrix} \neg\emptyset & \emptyset & \emptyset \\ \neg\emptyset & \emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix}$$

Figure 4.19 – Représentation des relations topologiques en utilisant la méthode des 9-IM et leur équivalence dans RCC8.

De nombreux travaux ont étendu le champ d'application de la méthode des 9-intersections, le modèle étendu DE-9IM (Dimensionally Extended 9-Intersection Model) [Clementini *et al.*, 1993] est particulièrement important car il vise à prendre en compte des entités de différentes dimensions. La dimension d'une intersection est calculée par une fonction *dim* qui produit quatre valeurs possibles : 0 pour un point, 1 pour une ligne, 2 pour une région et -1 si l'intersection n'existe pas [Strobl, 2008].

Ainsi, le modèle DE-9IM peut s'appliquer sur plusieurs types de données géométriques. Deux nouvelles relations sont également formées en plus :

— **Intersects** : L'intersection de deux géométries, quel que soit leur type, n'est pas un ensemble vide. Le prédicat est vrai si l'intérieur des géométries est

intersecté. Cela est représenté par quatre configurations suivantes (Figure 4.20) :

$$R_1(A,B) = \begin{pmatrix} T & * & * \\ * & * & * \\ * & * & * \end{pmatrix} \qquad R_2(A,B) = \begin{pmatrix} * & * & * \\ T & * & * \\ * & * & * \end{pmatrix}$$

$$R_3(A,B) = \begin{pmatrix} * & T & * \\ * & * & * \\ * & * & * \end{pmatrix} \qquad R_4(A,B) = \begin{pmatrix} * & * & * \\ * & T & * \\ * & * & * \end{pmatrix}$$

Figure 4.20 – Représentation de la relation *intersects* en utilisant le modèle DE-9IM (T: dim = 0, 1, ou 2 ; *: dim = -1, 0, 1, ou 2).

— **Crosses** : Ce prédicat ne s'applique qu'entre deux lignes ou un polygone et une ligne. L'intersection de ces géométries se situe à leur intérieur. Cela est représenté par trois configurations (Figure 4.21) :

$$R_1(A,B) = \begin{pmatrix} T & * & * \\ * & * & * \\ T & * & * \end{pmatrix} \qquad R_2(A,B) = \begin{pmatrix} T & * & T \\ * & * & * \\ * & * & * \end{pmatrix}$$

$$R_2(A,B) = \begin{pmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

Figure 4.21 – Représentation de la relation *crosses* en utilisant le modèle DE-9IM (T : dim = 0, 1, ou 2 ; * : dim = -1, 0, 1, ou 2).

4.3.5 Synthèse

Nous avons présenté les différents modèles de représentation des données spatiales en discutant de leurs avantages et de leurs inconvénients. Ensuite, les modèles de données ainsi que les systèmes de référence sont aussi détaillés. Tout cela a pour but de donner une meilleure compréhension du caractère spatial des données environnementales. En effet, dans la pratique, la représentation de données spatiales dépend du système de gestion qui applique un modèle de données de son choix et aussi du système de collecte qui utilise son propre système géodésique.

Les modèles de données spatiales et les modèles mathématiques permettant le raisonnement sur les relations topologiques ont aussi été abordés. Nous nous sommes intéressés aux modèles de données spatiales standards proposés par l'OGC qui est à l'origine de la majorité des ontologies de l'espace. En outre, il est aussi possible de transformer ces modèles en ontologie selon les besoins.

À propos du raisonnement sur les relations topologique, deux modèles différents ont été présentés.

4.4 Modélisation et raisonnement spatial dans le Web sémantique

Dans cette partie, nous étudions tout d'abord la modélisation de données spatiales par l'approche ontologique. Les approches utilisant le Web sémantique pour raisonner sur les relations topologiques sont décrites ensuite.

4.4.1 Ontologies de l'espace

De nombreux travaux ont proposé des ontologies de l'espace pour définir les concepts et les relations spatiaux. Ces ontologies sont majoritairement basées sur le modèle GML et utilisent le langage de formalisation ontologique OWL. Dans cette partie, nous nous intéressons à la partie déclarative et à la représentation des données spatiales par des ontologies. Nous présentons les ontologies de l'espace les plus utilisées : Basic Geo Vocabulary, GeoRSS et GeoOWL ; et la méthode de transformation d'un modèle spatial en une ontologie spatiale, à savoir qu'il y a aussi d'autres ontologie moins connues comme NeoGeo Geometry Ontology [11], GeoNames Ontology [12] ou Biological Spatial Ontology [13].

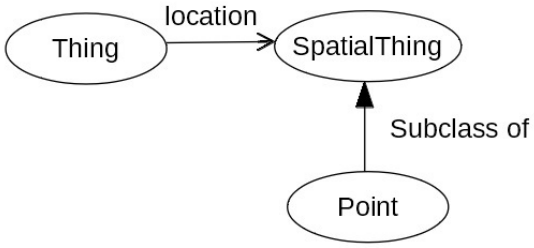


Figure 4.22 – Le vocabulaire Basic Geo.

4.4.1.1 Basic Geo Vocabulary

Un des premiers travaux dans ce domaine est le Basic Geo Vocabulary [14], un langage crée par le W3C. Il s'agit d'un vocabulaire fournissant un espace de nom pour représenter la localisation et les autres informations d'un objet basé sur le standard du WGS84. Ce vocabulaire a commencé l'exploration des possibilités

11. http://geovocab.org/
12. http://www.geonames.org/ontology/documentation.html
13. http://www.obofoundry.org/ontology/bspo.html
14. https://www.w3.org/2003/01/geo/

pour la représentation des données cartographiques ou de localisation par RDF mais il ne cherchait pas à développer des aspects géospatiaux. Par conséquent, un vocabulaire RDF minimaliste est déclaré pour décrire un point (*pos:Point*) avec sa longitude *(pos:long)*, sa latitude (*pos:lat*) et son altitude (*pos:alt*) (Figure 4.22).

4.4.1.2 GeoRSS

GeoRSS [15] est un standard permettant d'incorporer de l'information géographique dans des flux web de sorte que des applications puissent interroger, partager et fusionner des flux géoréférencés. Dans ce standard, des points, des lignes ou des zones d'intérêt sont associées à une caractéristique décrite dans un flux Web. Les différents flux supportés par GeoRSS sont RSS 1.0, RSS 2.0 et Atom.

Les encodages actuels de GeoRSS sont le GeoRSS Simple et GeoRSS GML. Le dernier est un langage de structuration qui supporte un plus grand nombre de fonctions que l'autre.

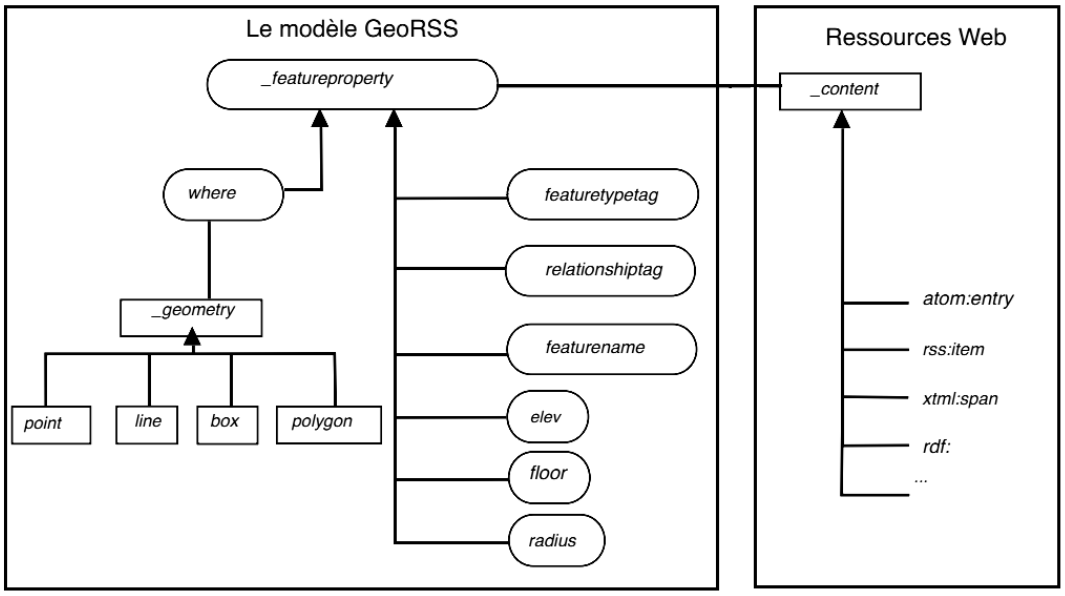


Figure 4.23 – Le modèle GeoRSS-Simple.

— **GeoRSS-Simple** : Le modèle est un format très léger et peut être facilement ajouté à des flux existants pour prendre en charge les géométries de base et couvre les cas typiques d'utilisation pour l'encodage de localisation. Dans ce format, la relation *georss:where* utilise un ensemble de types spatiaux équivalents à ceux de la spécification GML tels que : *georss:Point*, *georss:Line*, *georss:Polygon*, *georss:Circle* et *georss:Box* (Figure 4.23).

15. http://www.georss.org/

— **GeoRSS-GML** : Le modèle est un profil d'application formel de GML et bénéficie du support de l'OGC. Il supporte toutes les géométries existant dans GML et utilise le WGS84 comme système de référence par défaut. Dans ce format, la relation *geo:where* utilise un ensemble limite de types spatiaux de la spécification GML tels que : *gml:Point*, *gml:LineString*, *gml:Polygon* et *gml:Envelop* (Figure 4.24).

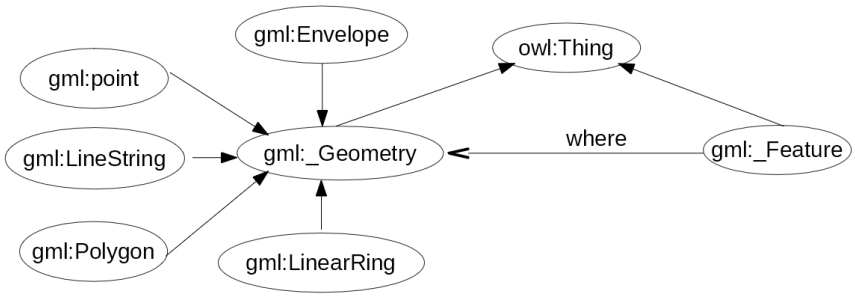


Figure 4.24 – Le modèle GeoRSS-GML.

Afin d'être le plus générique possible, GeoRSS se décline sous différentes formes : Atom, GML, RDF et OWL. Au centre de l'ontologie, on trouve la classe *gml:_Feature* qui regroupe tous les types d'éléments spatiaux, et peut être reliée à une description géométrique (*gml:_Geometry*) à travers l'utilisation de la propriété *where*. L'ontologie définit les géométries de base telles que *gml:Point*, *gml:Polygon*, *gml:LineString*, *gml:LinearRing*, *gml:Envelop* comme des spécialisations de la classe abstraite *gml:_Geometry*. Une telle présentation du GeoRSS au format OWL est décrite par la Figure 4.25.

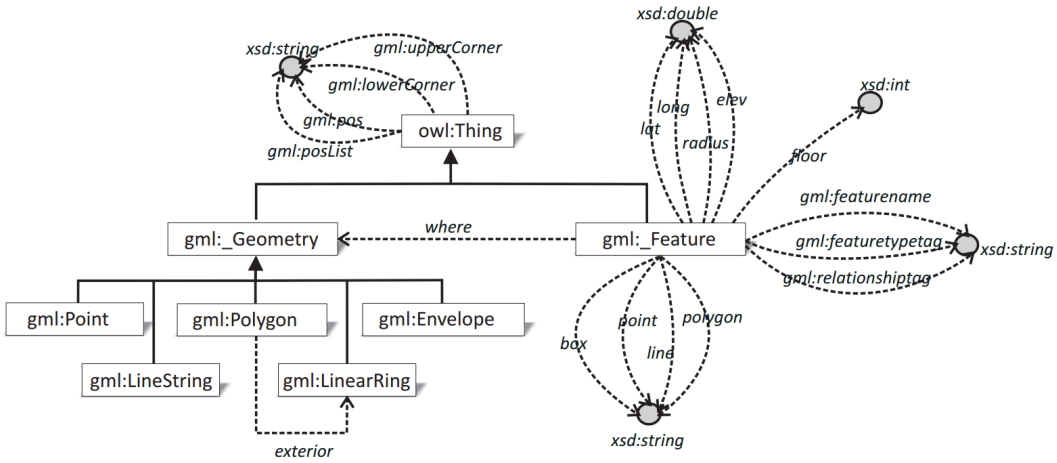


Figure 4.25 – Représentation du GeoRSS au format OWL [Miron, 2009].

GeoRSS offre une représentation spatiale simplifiée et de ce fait bénéficie d'une plus grande adoption par la communauté. Le W3C était largement in-

fluencé par cet encodage pour enrichir son vocabulaire Basic Geo Vocabulary. Ainsi, des travaux ont été effectués pour donner naissance à l'ontologie GeoOWL qui est devenu une recommandation pour la représentation des entités géographiques et géo-localisées.

4.4.1.3 GeoOWL

L'ontologie géospatiales GeoOWL a été développée pour étendre le Basic Geo Vocabulary qui ne supporte que la représentation d'un point. Elle est formalisée en OWL et basée sur le modèle de données GeoRSS. Ainsi, elle constitue une extension compatible de GeoRSS pour une utilisation dans des contextes RDF plus généraux.

L'ontologie se compose d'une propriété principale *_featureproperty* pour représenter les caractéristiques géographiques. Cette propriété a plusieurs sous-propriétés dont *geo:where* qui prend la classe abstraite *_geometry* pour valeur. Les sous-classes de *_geometry* comprennent les géométries définies par GML. Les propriétés de ces classes sont un sous-ensemble des propriétés correspondantes définies dans le modèle GML. Grâce à ces caractéristiques, l'ontologie est compatible avec GeoRSS GML. Enfin, les propriétés *geo:lat* et *geo:long* présentées dans Basic Geo Vocabulary sont retenues comme sous-propriétés de *geo:where*. À noter que les relations spatiales entre éléments géographiques, telles que *equals, disjoints* et *intersects* ne sont pas supportées par cette ontologie.

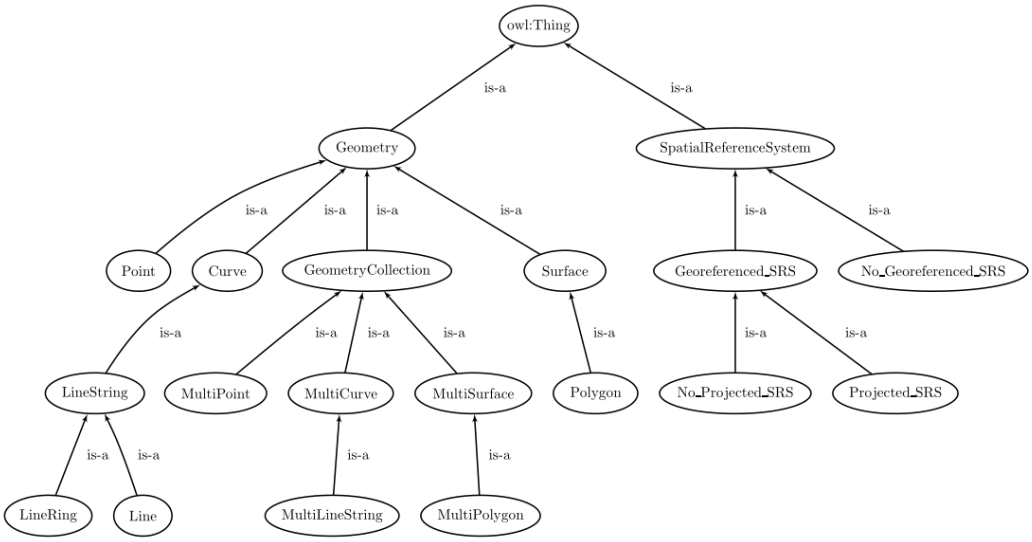


Figure 4.26 – Ontologie owlOGCSpatial [Mefteh *et al.*, 2012].

4.4.1.4 Transformation d'un modèle en ontologie

Il est possible d'utiliser un modèle de données spatiales pour créer une ontologie de l'espace. Une telle approche a été appliquée pour développer l'onto-

logie owlOGCSpatial (Figure 4.26). En effet, les auteurs ont utilisé la technique de transformation de modèle pour définir la partie déclarative de l'ontologie à partir du diagramme de classes UML de la spécification de l'OGC. De cette façon, les objets spatiaux appartiennent aux sous-classes de la classe *Geometry*. Ainsi, les coordonnées de ces objets sont représentées par la propriété *wkt* et les relations spatiales entre eux se traduisent par l'ensemble des propriété correspondantes, c'est-à-dire, *equals, disjoint, intersects, overlaps, contains, crosses, within* et *touches*. L'ontologie a montré des succès dans la modélisation et le raisonnement sur la dimension spatiale pour les données de trajectoires.

4.4.2 Raisonnement spatial

Afin de réaliser le raisonnement spatial dans le Web sémantique, plusieurs travaux ont été effectués en s'appuyant sur les approches : ajout du raisonnement spatial dans le Web sémantique, traduction des relations spatiales en axiomes de classes OWL, utilisation des règles SWRL spatiales et utilisation des langages de requête géospatial.

4.4.2.1 Architecture hybride pour le raisonnement spatial

[Grütter et Bauer-Messmer, 2007] a proposé une architecture hybride afin de pouvoir inclure le raisonnement spatial dans le Web sémantique (Figure 4.27). Dans ce système, la combinaison entre le modèle RCC et OWL ne se fait pas au niveau du formalisme mais directement dans le système de représentation. D'une part, les noms des relations RCC-1, RCC-2, RCC-3, RCC-5 et RCC-8 sont ajoutés dans la TBox sous forme d'une hiérarchie de propriétés. Celle-ci permet une communication entre les agents utilisant des relations différentes. D'autre part, ces noms seront référés pour représenter la relation entre deux régions en assertions qui sont ensuite ajoutées dans l'ABox. Le raisonneur consulte d'abord les assertions présentes dans l'ABox pour déterminer les relations entre les régions. Il se réfère par la suite aux tables de composition du RCCBox pour assurer la consistance de l'ABox.

Les résultats peuvent être stockés dans l'ABox afin d'optimiser le processus d'inférence. Bien que cette approche évite de calculer de nouveau le résultat des requêtes identiques, elle est déconseillée pour les raisons suivantes :

— L'assertion de toutes ces relations entraîne l'augmentation de la taille de la base de connaissances, nuisant ainsi aux performances du système.

— De plus, le stockage des inférences déjà réalisées soulève également la question de la consistance des résultats dans le temps. En effet, des entités peuvent être fusionnées ou détruites et leurs relations doivent être alors calculées de nouveau afin d'éviter l'inconsistance de la base de connaissances.

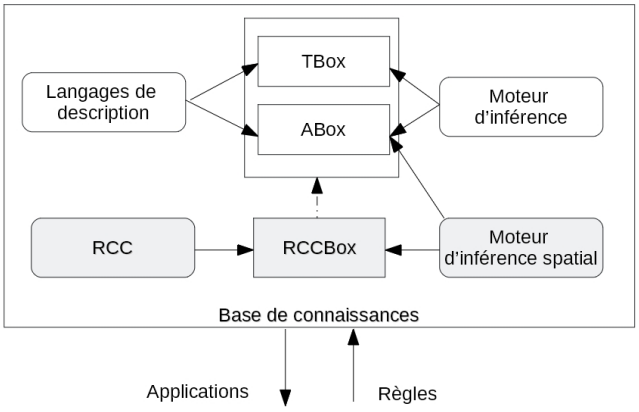


Figure 4.27 – Système hybride pour le raisonnement spatial basé sur le RCC.

— Le système hybride n'est capable de raisonner qu'à partir des relations déjà spécifiées dans l'ABox. Autrement dit, il n'est pas possible d'inférer des relations topologiques entre des objets en ne se basant que sur leur géométrie.

4.4.2.2 Système de raisonnement PelletSpatial

[Stocker et Sirin, 2009] ont proposé d'intégrer une extension spatiale au raisonneur Pellet, appelée PelletSpatial. Les principaux services de cette extension sont :

— Inférence des relations spatiales.

— Vérification de la consistance d'un ensemble de relations spatiales exprimées en formalisme RCC-8.

— Réponse aux requêtes spatiales exprimées en SPARQL combinant le RCC et le RDF/OWL.

Le premier raisonneur de PelletSpatial utilise la traduction des relations de RCC-8 en axiomes OWL-DL comme proposée par [Katz et Grau, 2005]. Pour cela, chaque relation de RCC-8 est traduite sous la forme d'une propriété et chaque région correspond à un concept valide en logique de description.

Bien que fonctionnelle, cette approche souffre de mauvaises performances en raison du processus de traduction d'une logique vers une autre. En effet, celui-ci s'avère coûteux en temps de calcul même pour un nombre réduit de régions [Stocker et Sirin, 2009]. En plus de ce raisonneur spatial, PelletSpatial implémente également un langage de requête basé sur SPARQL. Il est ainsi possible d'interroger la base de connaissances en utilisant des critères de sélection spatiaux et non spatiaux.

4.4.2.3 Les règles SWRL spatiales

Il est aussi possible d'utiliser des règles SWRL spatiales pour raisonner sur les relations spatiales. Si les règles classiques peuvent être directement interprétées, les règles spatiales nécessitent quant à elles un processus de traduction. En effet, comme le langage SWRL ne propose aucun type ni fonction spatiale, il nécessite alors une étape d'interprétation au préalable. Celle-ci permet de reconnaître la fonction spatiale ainsi que les arguments passés en paramètre avant de l'envoyer au module de raisonnement spatial.

[Karmacharya *et al.*, 2010] ont introduit des *built-ins* spatiaux qui sont des extensions de ceux existants dans le SWRL. Ils permettent d'exécuter des règles déductives à l'aide d'un moteur de traduction de règles. Ainsi, les règles SWRL spatiales sont traduites en règles SWRL normales. Dans le système proposé, les atomes spatiaux sont d'abord calculés grâce à une base de données spatiale. Le résultat de ces derniers est ensuite interprété en axiomes avant d'être ajouté dans l'ontologie. Enfin, le résultat d'une règle peut être calculé avec un raisonneur comme Racer, Jess ou Pellet en remplaçant les atomes par les axiomes correspondants (Figure 4.28). Cette approche a été appliquée pour l'analyse de données spatiales dans le domaine d'archéologie industrielle.

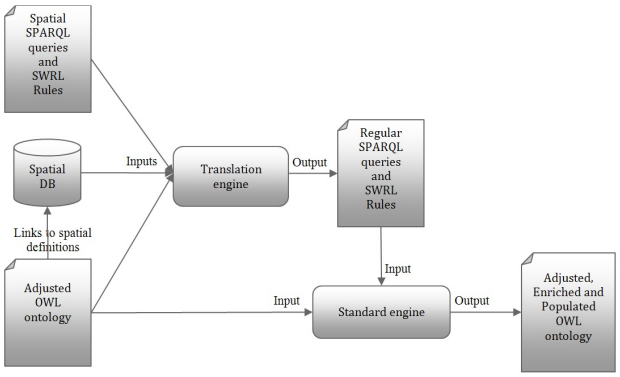


Figure 4.28 – Traduction des règles SWRL spatiales [Karmacharya *et al.*, 2010].

Bien que fonctionnelle, le système possède plusieurs inconvénients :

— Il nécessite une base de données spatiale pour le stockage de données et la gestion des relations spatiales. Dans le cas des atomes de traitement spatial comme *spatialswrlb:Buffer*, le système doit calculer de nouvelles géométries et les stocker dans la base de donnés.

— Il souffre de la prolifération d'individus et d'axiomes. En effet, il est nécessaire d'ajouter de nouveaux individus et axiomes à l'ontologie en vue de maintenir le lien avec la base de données et le lien entre les individus. Par exemple, dans le cas de l'atome *spatialswrlb:Buffer*, un nouveau individu

de type *feat:sp_Buffer* et deux axiomes sont ajoutés pour chaque couple (géométrie, taille de buffer).

— La gestion des données intermédiaires et la traduction des règles ont aussi un impact important aux performances du système.

Au lieu d'une base de données spatiale, le raisonneur PelletSpatial est utilisé dans [Vandecasteele et Napoli, 2012], ou Pellet avec Java Topology Suite dans [Vandecasteele, 2012] pour exprimer les relations spatiales en règles SWRL spatiales (Figure 4.29). Ces méthodes ont été appliquées dans l'analyse de comportements de navires. Comme souligné par l'auteur, l'augmentation du nombre des objets entraînait une diminution des performances du système.

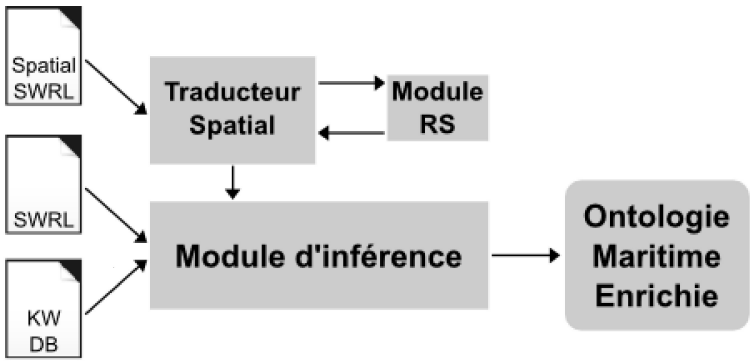


Figure 4.29 – Traduction des règles SWRL spatiales [Vandecasteele et Napoli, 2012].

4.4.2.4 Raisonnement spatial par triplestores géospatiaux

Dans la section 3.2.2, nous avons présenté différents triplestores géospatiaux qui ont pour but de permettre non seulement le stockage et la gestion des données spatiales mais aussi le raisonnement sur leurs relations topologiques. La capacité de raisonnement est accessible à l'aide d'un langage de requête géospatial comme GeoSPARQL ou stSPARQL (voir 3.2.3). Les performances du raisonnement spatial de ces triplestores sont incomparables à celles de ces approches précédentes parce que :

— Certains triplestores, comme Strabon, sont construit au-dessus d'un SGBD spatial pour pouvoir profiter des optimisations de performances proposées par celui-ci.

— Il ne nécessite pas un raisonneur, ni l'étape de préparation ou de traduction au préalable.

— Il ne souffre pas de la prolifération d'individus ou d'axiomes.

L'approche utilisée par [Mefteh *et al.*, 2012, Wannous, 2014] pour modéliser et raisonner sur les données de trajectoires. Les auteurs utilisent Oracle Spatial

and Graph, un triplestore géospatial, pour la gestion et le raisonnement sur les trajectoires. [Chentout et Vaisman, 2013] a aussi appliqué cette approche pour interroger les données spatio-temporelles de la ville de Brussels qui sont gérées par Strabon. Ou encore [Pokharel *et al.*, 2014] ont utilisé Virtuoso, un triplestore avec le support spatial, pour la gestion et l'interrogation des données spatio-temporelles dans le domaine agricole de Népale.

4.4.3 Synthèse

Dans cette section, nous avons décrit les ontologies visant à représenter des données spatiales mais aucune d'entre elles ne fournit un vocabulaire pour décrire les relations topologiques. Or ces relations sont nécessaires pour le croisement des sources de données spatio-temporelles. Par conséquent, afin de représenter ces relations, les solutions suivantes se proposent :

— Utilisation d'une autre ontologie incluant des relations spatiales comme NeoGeo.

— Utilisation d'une des ontologies présentées en y ajoutant des relations topologiques.

— Application de la transformation de modèle pour créer une nouvelle ontologie de l'espace.

En ce qui concerne le raisonnement spatial, bien que fonctionnelles, les différentes solutions proposées varient grandement en terme de performances et de capacité de raisonnement. Si l'intégration de concepts spatiaux directement au sein du formalisme de représentation semble être l'alternative la plus pertinente, elle souffre néanmoins d'importantes limites notamment en termes de calcul. De ce fait, l'approche consistant à modifier le système de représentation des connaissances apparaît comme la solution la plus adaptée. Néanmoins, ce système hybride n'est capable de raisonner qu'à partir de relations déjà spécifiées dans l'ABox. Autrement dit, cela signifie qu'il n'est pas possible d'inférer des relations topologiques entre des objets en ne se basant que sur leur position. À notre avis, les trois premières sont plutôt des preuves de concept. Elles souffrent de mauvaises performances, donc ne conviennent qu'aux petits ensembles de données. Au contraire, la dernière approche donne de bonnes performances mais elle nécessite une matérialisation de données dans un triplestore spatial. Dès lors, elle est la plus adaptée à l'approche d'intégration par l'entrepôt.

4.4.4 Modélisation et raisonnement spatio-temporels dans le Web sémantique

Le temps et l'espace se représentent différemment selon des visions personnelles. En effet, la représentation de celles-ci dépend de schémas cognitifs individuels. La question est alors de savoir comment traduire ces représentations sous

une forme exploitable et partagée pour favoriser l'interopérabilité entre systèmes, autrement dit l'intégration de données spatio-temporelles hétérogènes.

Les chapitres précédents ont montré que les ontologies permettent d'effectuer cette traduction. En effet, dans ce contexte, elles sont utilisées comme un format d'échange, plus particulièrement elles jouent le rôle d'un schéma global pour l'intégration sémantique des sources hétérogènes. De plus, elles facilitent l'analyse de ces données en prenant en compte le contexte et la connaissance liées au domaine de l'application ainsi qu'en fournissant des outils pour inférer de nouvelles connaissances.

La modélisation spatio-temporelle par une approche ontologique concerne les ontologies du domaine, les ontologie du temps et les ontologies de l'espace pour la représentation des informations sémantiques, des informations temporelles et des informations spatiales respectivement (Figure 4.30). En plus de ces trois composantes, la modélisation spatio-temporelle dans le Web sémantique nécessite un mécanisme pour incorporer le temps et raisonner sur les relations spatio-temporelles.

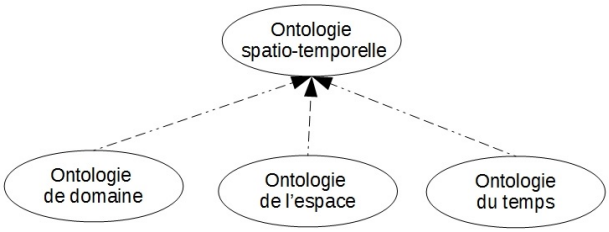


Figure 4.30 – Les composantes d'une ontologie spatio-temporelle.

À noter que le raisonnement spatio-temporel peut être considéré comme l'étape logique suivante au raisonnement spatial et au raisonnement temporel [Wolter et Zakharyaschev, 2000]. La plupart des formalismes de modélisation spatio-temporelle proposés sont basés sur une combinaison de modèles de raisonnement spatial et temporel [Wolter et Zakharyaschev, 2000, Muller, 2002]. Ainsi, les relations spatio-temporelles sont déduites par la combinaison entre les relations spatiales et les relations temporelles. En effet, de nombreux travaux ont appliqué cette approche dans l'inférence des relations spatio-temporelles, comme dans [Claramunt et Jiang, 2001, Ren *et al.*, 2009]. En conséquence, de la même manière, le raisonnement spatio-temporel dans le Web sémantique peut se réaliser en combinant les mécanismes de raisonnement temporel et spatial décrits précédemment.

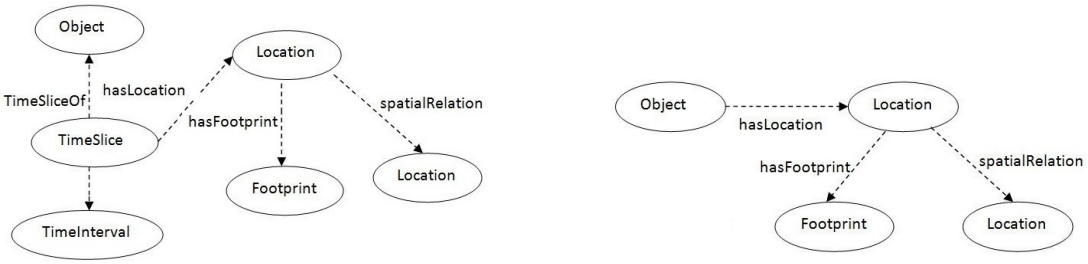
Dans cette section, nous présentons brièvement deux ontologies qui visent à représenter et à raisonner sur les informations spatio-temporelles. Ces modèles ont un point en commun, c'est qu'ils appliquent l'approche 4D pour la modélisation spatio-temporelle. En effet, le concept *TimeSlice* joue le rôle central en étant lié à une dimension spatiale, une dimension temporelle et éventuellement une di-

mension sémantique. À noter qu'il existe aussi de nombreuses recherches sur la modélisation spatio-temporelle par l'approche ontologie appliquée aux données de trajectoire comme [Vandecasteele et Napoli, 2012, Mefteh *et al.*, 2012, Wannous, 2014, Noël *et al.*, 2015, Soltan Mohammadi *et al.*, 2017].

4.4.4.1 Le modèle SOWL

SOWL [Batsakis et Petrakis, 2011] est une ontologie pour représenter et raisonner sur les informations spatio-temporelles dans OWL. En s'appuyant sur des normes bien établies du Web sémantique, telles que l'OWL 2.0 et le SWRL, SOWL permet une représentation des informations statiques et dynamiques. Les deux relations directionnelles et topologiques du RCC-8 sont intégrées dans SOWL. La représentation commune à la fois des informations temporelles et spatiales qualitatives en plus des informations quantitatives est une caractéristique distinctive du modèle.

La représentation du temps dans SOWL est basée sur l'approche 4D-fluents améliorée par les relations temporelles d'Allen qui sont définies comme des propriétés entre les intervalles. Une autre implémentation basée sur l'approche N-aire a également été mise en œuvre. Les relations spatiales topologiques et directionnelles sont utilisées pour définir la relation entre les étendues spatiales des objets.



(a) Le cas d'un objet mobile. (b) Le cas d'un objet statique.

Figure 4.31 – Le modèle SOWL [Batsakis et Petrakis, 2011].

Le modèle fait une distinction entre un objet mobile et un objet statique. Dans le cas d'un objet mobile, sa position est une propriété du *timeslice* validant un intervalle de temps spécifique (Figure 4.31a), alors que pour un objet statique, elle est une propriété de l'objet lui-même (Figure 4.31b). Même si la position d'un objet est statique, certaines de ses propriétés peuvent changer, dès lors, l'objet est considéré comme mobile. Dans le cas des trajectoires auquel la position de l'objet change de manière continuelle, le raisonnement n'est pas encore géré.

Le raisonnement spatio-temporel appliqué dans le modèle est basé sur un ensemble de règles en SWRL exprimant les relations temporelles et les relations spatiales topologiques et directionnelles. Cependant, cette approche ne peut garantir la décision et n'est donc pas compatible avec les spécifications du W3C.

4.4.4.2 Le modèle Continuum

[Harbelot, 2015] a introduit le modèle Continuum pour l'analyse des phénomènes dynamiques géospatiaux. Le modèle est une ontologie spatio-temporelle qui reprend les fondamentaux des ontologies de fluents ainsi que des modèles de représentation spatiale. En outre, les connaissances et le contexte liés à l'environnement géospatial sont stockés au sein d'une ontologie de domaine. La Figure 4.32 illustre les grandes composantes du modèle.

Le modèle utilise des entités dynamiques évoluant dans le temps, nommés *timeslices* pour relier ces composantes. Chaque *timeslice* se définit selon quatre composantes que sont : l'identité, l'espace, le temps, et la sémantique intrinsèque de l'entité (Figure 4.33). L'identité est la composante la plus importante du modèle.

Afin de représenter les associations spatiales dans le temps, la relation *filiation* est introduite. Elle permet de connecter deux *timeslices* consécutifs dans le temps. Cette propriété est essentielle pour établir un lien spatio-temporel entre deux entités. Dans ce modèle, la relation de filiation est spécialisée à travers différentes couches de connaissances. Ainsi chaque spécialisation de cette relation offre une connaissance plus approfondie de l'évolution d'une entité.

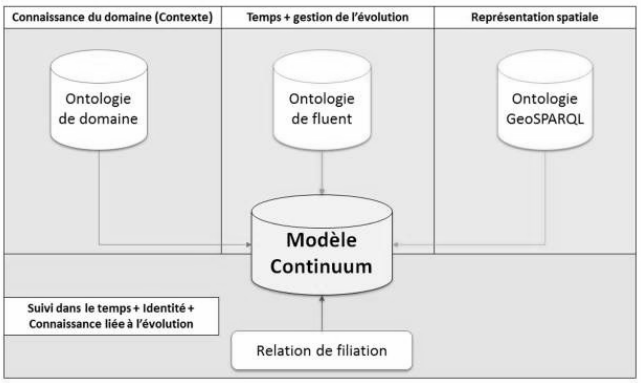


Figure 4.32 – Le modèle Continuum [Harbelot, 2015].

Pour le raisonnement temporel, l'auteur a adopté l'algèbre d'Allen représentée sous formes de règles SWRL. Quant au raisonnement spatial, en théorie, selon la présentation, les fonctions d'extension de GeoSPARQL devraient être exploitées. Néanmoins, le triplestore Stardog [16] a été adopté en raison de ses capacités de raisonnement. Plus précisément, selon l'auteur, Stardog supporte des inférences en OWL et par les règles SWRL et offre ainsi des options uniques pour traiter des contraintes sous l'hypothèse du monde fermé ou du monde ouvert. Par conséquent, l'analyse spatiale se réalise à l'aide des outils externes. Ainsi toutes

16. http://stardog.com/

informations liées à la dimension spatiale sont traitées et identifiées avant d'être converties en RDF et finalement chargées dans le triplestore.

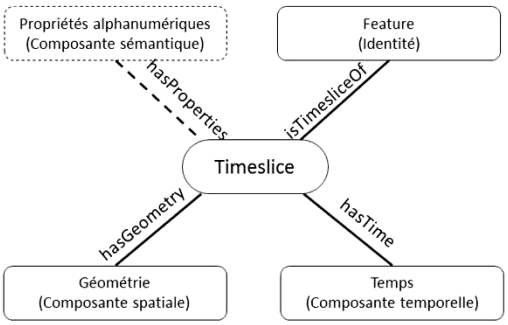


Figure 4.33 – Quatre composantes du *Timeslice* [Harbelot, 2015].

4.4.4.3 Synthèse

Nous avons présenté deux modèles permettant la représentation de données spatio-temporelles dans le Web sémantique. Le point commun entre ces modèles est du au fait qu'ils appliquent les approches perdurantistes, soit 4D-fluent, soit N-aire, pour modéliser les changements. En effet, le concept central de ces modèles est le *timeslice* qui est attaché à l'objet et se compose d'une dimension temporelle, d'une dimension spatiale, et éventuellement d'une dimension sémantique. Chacune de ces dimensions peut être modélisée par une ontologie temporelle, une ontologie spatiale et une ontologie du domaine respectivement.

Conclusion du chapitre

Ce chapitre nous a permis de présenter les notions et les modèles que nous jugeons les plus importants pour la gestion du temps et de l'espace en nous concentrant sur la modélisation spatio-temporelle dans le Web sémantique. En général, un modèle spatio-temporel nécessite une ontologie du temps, une ontologie de l'espace et une ontologie du domaine pour représenter les informations spatio-temporelles et les caractéristiques des objets en études. De plus, en vue d'effectuer le croisement des données hétérogènes, il est nécessaire de considérer aussi leurs relations spatio-temporelles. Ces dernières sont obtenues par le raisonnement spatio-temporel qui est considéré comme une association du raisonnement temporel et du raisonnement spatial.

Pour ces raisons, le chapitre commence par les études sur les représentations du temps ainsi que les opérateurs pour le raisonnement temporel. Plus particulièrement, nous avons souligné l'importance de l'algèbre de l'instant et de l'intervalle envers ce type de raisonnement dans le domaine environnemental. Ensuite, les ontologies du temps ainsi que les approches permettant la représentation des changements ont été étudiées.

Dans la partie suivante, nous avons présenté les modèles de représentation de donnés spatiales, les systèmes géodésiques et le raisonnement sur les relations topologiques. Cela forme les connaissances de base sur la représentation de données spatiales pratiquement utilisées dans les SIG en général et les SIG dans le domaine environnemental en particulier, comme celui de la zone atelier de Plaine et Val de Sèvre.

A partir des éléments étudiés, d'après nous, l'ontologie OWL-Time et les approches perdurantistes comme N-aire et 4D-fluent sont les meilleurs candidats pour la modélisation de la dimension temporelle et pour associer cette dimension aux objets. Quant à la dimension spatiale, l'ontologie owlOGCSpatial, développée par notre équipe, est la plus adaptée à notre contexte car elle permet la représentation des concepts et des relations spatiales en respectant la norme de l'OGC. Pourtant, lorsqu'un langage de requête comme GeoSPARQL ou stSPARQL est utilisé, il est nécessaire d'intégrer son ontologie, c'est-à-dire l'ontologie de GeoSPARQL ou le modèle stRDF, à l'ontologie spatiale choisie.

Au niveau du raisonnement temporel, l'utilisation de requête SPARQL comme langage de règles est l'approche la plus adaptée à notre contexte. Ainsi, chaque relation temporelle peut être exprimée par une requête dont le résultat sera ajouté directement à la base de connaissances. Pour le raisonnement spatial, quant à lui, afin d'obtenir de meilleures performances, il est souhaitable d'utiliser un triplestore géospatial au lieu des approches traditionnelles. Néanmoins, cette approche oblige la matérialisation des sources de données.

Chapitre 5

Fouille de donnés sémantiques

Sommaire

5.1 Extraction de connaissances à partir de données . . . 101
5.1.1 Présentation . . . 101
5.1.2 Algorithmes de fouille de données . . . 103
5.1.2.1 Les règles d'association . . . 103
5.1.2.2 Arbre de décision . . . 104
5.1.2.3 Partitionnement . . . 106
5.2 Fouille de donnés sémantiques . . . 107
5.2.1 Approches statistiques . . . 108
5.2.2 Rôles de l'ontologie dans l'extraction de connaissances 109
5.2.3 Synthèse . . . 110

Introduction du chapitre

Les technologies du Web sémantique ont été choisies pour développer un outil d'intégration sémantique et d'exploitation de données spatio-temporelles hétérogènes. Afin de proposer un outil efficace, nous souhaitons aller plus loin en y intégrant un ensemble de processus d'extraction de connaissances. Pour cela, nous cherchons une approche permettant la découverte de connaissances à partir de données sémantiques de caractère spatio-temporelle, ou autrement dit une base de connaissances spatio-temporelle. Une solution possible est de se référer au processus de l'extraction de connaissances à partir de données (ECD) (Figure 5.1).

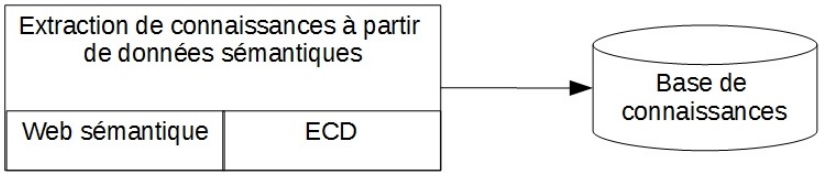


Figure 5.1 – Extraction de connaissances à partir de données sémantiques.

La première partie du chapitre présente brièvement les notions et les étapes de l'ECD en se concentrant sur les techniques de fouille de données. Un algorithme typique de chaque technique est étudié.

Dans la deuxième partie, nous présentons la fouille de données du Web sémantique et les différentes approches visant à appliquer les techniques de fouille traditionnelles aux données sémantiques.

La dernière partie est consacrée à la discussion du rôle de l'ontologie (et du Web sémantique) dans chaque étape de l'extraction des connaissances à partir d'une base de connaissances.

5.1 Extraction de connaissances à partir de données

Avec la croissance exponentielle de la collecte et du stockage des données, le problème de l'accès aux connaissances renfermées dans ces données dépasse largement les capacités d'analyse humaines. Les utilisateurs ont besoin de nouveaux outils pour analyser leurs données et en extraire des connaissances auparavant inconnues sans forcément avoir besoin de connaissances en informatique et/ou en statistiques. Les techniques d'extraction de connaissances ont vu le jour dans les années 1990 afin de répondre à ces besoins. Dans cette partie, nous présentons les notions et les processus utilisés pour extraire des connaissances à partir de données. Trois techniques de fouilles sont détaillées dans la suite avec un algorithme de démonstration.

5.1.1 Présentation

L'extraction de connaissances dans les bases de données (ECD, ou KDD en anglais) désigne le processus d'extraction des informations implicites, précédemment inconnues et potentiellement utiles à partir des données [Frawley *et al.*, 1992]. Les informations extraites sont potentiellement utiles pour l'aide à la décision, la gestion des informations, l'optimisation des requêtes ou le contrôle de processus, etc.

Elle combine les différentes techniques issues de diverses disciplines dont l'analyse de données, l'intelligence artificielle et l'apprentissage automatique pour extraire des savoirs ou des connaissances à partir de grandes quantités de données. Plus concrètement, [Collard, 2003] a résumé les objectifs fondamentaux de la discipline :

— Fouiller, creuser et extraire ce qui est caché.

— Prendre en compte le volume de données.

— Transformer des données brutes en connaissances expertes.

— Fournir des connaissances précieuses, nouvelles, valides et utiles à un utilisateur expert.

L'ECD est un processus semi-automatique et itératif, constitué de plusieurs étapes allant de la préparation des données jusqu'à la présentation des résultats, en passant par la phase de fouille de données [Fayyad *et al.*, 1996]. Les différentes étapes de ce processus sont décrites dans la Figure 5.2.

— **Sélection** : La première étape est de développer une compréhension du domaine d'application, de capturer des connaissances pertinentes, et d'identifier l'objectif de la fouille du point de vue de l'utilisateur final. Sur la base de cette compréhension, les données cibles utilisées dans le processus peuvent être choisies, il s'agit de sélectionner des échantillons de données appropriés et un sous-ensemble pertinent de variables.

— **Prétraitement** : Dans cette étape, les données sélectionnées sont traitées de manière à permettre une analyse ultérieure. Les mesures typiques prises dans cette étape incluent la gestion des valeurs manquantes, l'identification (et éventuellement la correction) du bruit et des erreurs dans les données, l'élimination des doublons, la fusion et la résolution des conflits des données provenant de différentes sources.

— **Transformation** : La troisième étape produit une projection des données dans une structure que les algorithmes de fouille peuvent exploiter. Dans la plupart des cas, cela implique de transformer les données en une forme propositionnelle où chaque instance est représentée par un vecteur de caractéristiques. Pour améliorer les performances des algorithmes de fouille ultérieurs, des méthodes de réduction de dimensionnalité peuvent également être appliquées en vue de réduire le nombre effectif de variables considérées.

— **Fouille de données** : L'étape a lieu une fois que les données sont présentes dans un format utile et que l'objectif initial du processus est adapté à une méthode de fouille particulière, telle que la classification, la régression ou le regroupement. Elle comprend la détermination des modèles et des paramètres appropriés et l'adaptation de méthode de fouille aux critères globaux du processus. L'étape résulte des modèles (de fouille) dans une forme de représentation particulière ou un ensemble de représentations, telles que des partitions, des arborescences, ou des ensembles de règles.

— **Évaluation et interprétation** : Dans cette dernière étape, les modèles obtenus sont examinés en fonction de leur validité. Par ailleurs, l'utilisateur évalue l'utilité des connaissances extraites par l'algorithme. Cette étape peut également impliquer la visualisation des modèles.

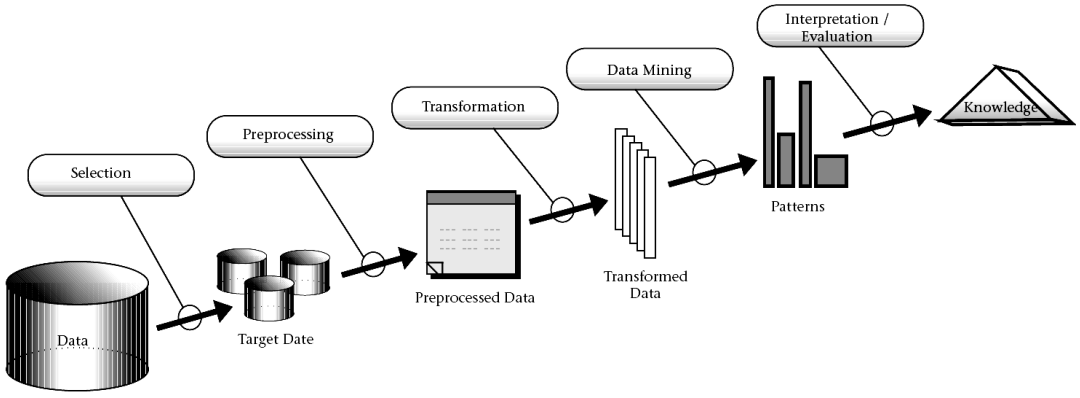


Figure 5.2 – Processus de découverte de connaissances [Fayyad *et al.*, 1996].

Nous pouvons distinguer deux catégories de techniques de fouille : l'apprentissage supervisé et l'apprentissage non supervisé. La deuxième se différencie de

la première par le fait qu'elle ne dispose pas de connaissances sur les classes à identifier. De plus, aucune des données disponibles ne sont étiquetées, nous ne pouvons donc construire de modèle d'apprentissage. Par conséquent, ces deux types d'apprentissage ont des objectifs généralement différents. L'apprentissage supervisé est utilisé pour tâches prédictives alors que l'apprentissage non supervisé est plutôt utilisé à des fins exploratoires.

La fouille de données exploratoire essaie d'identifier de l'information explicite qu'un utilisateur ou un décideur peut directement interpréter et transposer. Les techniques les plus connues dans cette catégorie sont les règles d'association et le partitionnement. Leur objectif est de détecter des régularités dans les données pour en identifier par la suite des groupes homogènes. La prédiction est une autre catégorie de techniques de fouille qui consiste à trouver une fonction capable d'associer un ensemble de paramètres d'entrée avec une ou plusieurs variables de sortie.

Dans la suite, nous présentons brièvement ces trois techniques accompagnées chacune d'un algorithme typique que nous avons utilisé dans nos travaux.

5.1.2 Algorithmes de fouille de données

Dans cette partie, nous présentons trois types d'algorithmes de fouille de données que nous jugeons pertinents pour notre contexte : les règles d'association, les arbres de décision et le partitionnement. En effet, il est important que l'algorithme permette des analyses et que le résultat soit simple, facilement compréhensible et interprétable par les utilisateurs finaux.

5.1.2.1 Les règles d'association

L'extraction de règles d'association a été introduite par [Agrawal *et al.*, 1993]. Elle a été développée à l'origine pour l'analyse des bases de données de transactions de ventes, et avait pour objectif de découvrir des relations significatives entre les produits vendus. Cette méthode a été ensuite étendue et appliquée dans plusieurs domaines dans le but de découvrir des relations entre les variables stockées dans les bases de données.

Une règle d'association est définie syntaxiquement comme une règle logique de la forme «si X alors Y» ($X \Rightarrow Y$) où les propositions X et Y sont des conjonctions d'expressions simples de comparaison sur les attributs. X est appelé antécédent de la règle ou partie gauche et Y est appelé conséquence ou partie droite. Par exemple, une règle découverte à partir des données de ventes dans un supermarché pourrait indiquer qu'un client achetant des couches serait susceptible d'acheter de la bière ($\{\text{Couches}\} \Rightarrow \{\text{Bière}\}$).

Afin de sélectionner des règles intéressantes à partir de l'ensemble des règles possibles, des contraintes sur diverses mesures de signification et d'intérêt sont

utilisées. Les contraintes les plus connues sont les seuils minimums de support et de confiance.

— **Le support** : Le support est un indicateur de fiabilité de la règle qui est égal à la fréquence d'apparition d'un ensemble d'*items* dans les transactions par rapport au nombre total de transactions de la base de données.

$$supp(X) = \frac{\mid \{t \in T; X \subseteq t\} \mid}{\mid T \mid}$$

— **La confiance** : La confiance est un indicateur de pertinence de l'inférence par une règle. Elle indique la fréquence d'apparition de la règle.

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$$

Dans le processus d'extraction de règles d'association, seuls les ensembles d'*items* ayant un bon support et les règles ayant une bonne confiance sont gardées. Le support et la confiance doivent être calculés pour toutes les règles possibles puis comparés aux seuils définis a priori par les utilisateurs. Les règles ayant les valeurs de support et de confiance au-delà de ces seuils sont conservées car elles possèdent des relations dites fortes.

Apriori [Agrawal et Srikant, 1994] est l'algorithme fondateur de l'extraction de règles d'association. Cet algorithme commence par une phase de recherche des ensembles d'*items* (*itemsets*) fréquents (ou candidats) en balayant la base de données. Une structure en treillis permet de générer des *itemsets* par niveau (*itemsets* de longueurs 1, 2..., k, k étant la longueur maximale des *itemsets* de la base de données). Un *itemset* est considéré comme fréquent si le support calculé est supérieur ou égal au support minimal. Dans la deuxième phase, Apriori sauvegarde les règles dont la confiance dépasse la confiance prédéfinie.

5.1.2.2 Arbre de décision

Un arbre de décision est un classifieur supervisé qui représente des résultats de classement sous la forme d'une arborescence. Le but des arbres de décision est de permettre la prédiction, c'est-à dire de prédire la classe d'un nouvel exemple à partir de ses attributs. Il est devenu un des outils populaires pour générer des règles de classification et plus généralement des règles de prédictions. On parle ainsi des arbres de classification lorsque la variable à prédire est catégorielle et que ses valeurs représentent donc des classes.

L'arbre est en général construit en séparant l'ensemble des données en sous-ensembles homogènes possibles par des tests en fonction de la variable à prédire. Ce processus est répété sur chaque sous-ensemble obtenu de manière récursive. Il s'agit donc d'un partitionnement récursif qui est achevé à un nœud soit lorsque tous les sous-ensembles ont (presque) la même valeur de la caractéristique-cible, ou lorsque la séparation ne peut plus améliorer la prédiction.

Un arbre de décision est composé d'un nœud racine, d'un ensemble de nœuds internes et d'un ensemble de feuilles. Chaque feuille de l'arbre dénote une classe. Chaque nœud interne de l'arbre, appelé nœud de décision, est étiqueté par un test qui peut être appliqué à toute description d'un individu. Généralement, chaque nœud correspond à un test sur un attribut unique. Ainsi, chaque branche issu d'un nœud est une réponse possible au test et correspond à une valeur de l'attribut ou à un intervalle de valeurs.

La Figure 5.3 représente un arbre de décision qui essaie de diagnostiquer des maladies, chacune se trouve dans les feuilles de l'arbre. Celles-ci représentent les quatre classes d'affectation dont trois maladies : {*Rhume, Mal de gorge, Refroidissement*}. L'attribut *Douleur*, situé sur le premier niveau, représente la racine de l'arbre et prend une de deux valeurs {*Gorge, Non*}. Pour chaque nouvel exemple dont on ne connaît pas l'étiquette, un parcours de l'arbre est effectué du nœud racine vers une feuille, la classe de la feuille atteinte est alors affectée à l'exemple au final.

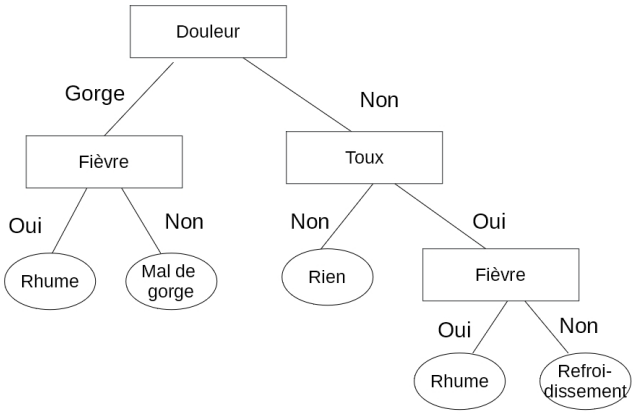


Figure 5.3 – Un arbre de décision diagnostiquant les maladies.

Dans toutes les méthodes, l'arbre est construit de la racine vers les feuilles de manière gloutonne et récursive :

— Décider si un nœud est terminal, c'est-à-dire s'il doit être étiqueté comme une feuille ou porter un test.

— Si un nœud n'est pas terminal, sélectionner un test à lui associer.

— Si un nœud est terminal, affecter-lui une classe.

En pratique, il n'est pas toujours souhaitable de construire un arbre dont les feuilles correspondent à des sous-ensembles parfaitement homogènes du point de vue de la variable-cible. En effet, les performances d'un arbre de décision reposent principalement sur la détermination de sa taille. Les arbres ont tendance à produire un classifieur trop complexe, collant exagérément aux données ; c'est le phénomène de sur-apprentissage [Breiman *et al.*, 1984].

Il est possible de construire un ensemble de règles à partir de l'arbre de décision en générant une règle pour chaque feuille. Pour cela, on fait des conjonctions de tous les tests rencontrés dans le chemin depuis la racine jusqu'à cette feuille [Tufféry, 2012]. Ces règles, de type «Si X alors Y» (X ⇒ Y) , fournissent un modèle prédictif facilement interprétable et compréhensible sans besoins de connaissances préalables du modèle de prédiction. Par exemple, l'arbre ci-dessus peut produire la règle suivante :

$$Si\ (Douleur = Gorge\ et\ Fièvre = Non)\ alors\ Maladies = Mal\ de\ gorge$$

Le post-traitement des règles extraites de l'arbre a été intégré dans l'algorithme C4.5, appelé «C4.5 rules» [Quinlan, 1993], afin de rendre plus compact et robuste le modèle de prédiction.

Les arbres de décision sont parmi les méthodes les plus utilisées en raison de la simplicité, des règles explicites fournies par le classement, de la faible indépendance avec l'échantillon de données et de la facilité de compréhension et d'interprétation.

5.1.2.3 Partitionnement

Les techniques de partitionnement de données (ou «clustering») sont des méthodes statistiques largement utilisées dans la fouille de données. Elles visent à partitionner un ensemble d'objets hétérogènes en un certain nombre de sous-ensembles plus homogènes, appelés partitions (ou «clusters»). Cela signifie que les partitions doivent contenir des données qui partagent un haut degré de similarité en vue de maximiser la similarité à l'intérieur de chaque partition et de minimiser la similarité entre les partitions. En général, chaque partition doit contenir au moins un objet, et donc les partitions vides ne sont pas tolérées. En outre, chaque objet doit appartenir à une seule partition.

La Figure 5.4 montre le résultat du partitionnement par l'algorithme K-moyennes sur l'ensemble de données des fleurs d'iris [Fisher, 1936] en fonction de la longueur et la largeur de leurs sépales et de leurs pétales en fixant le nombre de partitions à trois.

Un critère généralement utilisé pour juger la qualité du partitionnement est la proximité des objets. En effet, lors d'un bon partitionnement, les objets d'une même partition doivent être très proches les uns des autres et très éloignés des autres partitions. Cette notion de proximité ou d'éloignement induit forcément un calcul de distance géométrique entre ces objets.

Lorsqu'un algorithme de partitionnent est appliqué, il est nécessaire de déterminer le nombre de partition au préalable. Or, en général l'utilisateur ne possède pas assez de connaissances sur les données et ne peut donc connaître le nombre idéal de partitions. Une des premières idées est de calculer un critère pour plusieurs partitions candidates et de ne retenir que celle qui optimise ce critère.

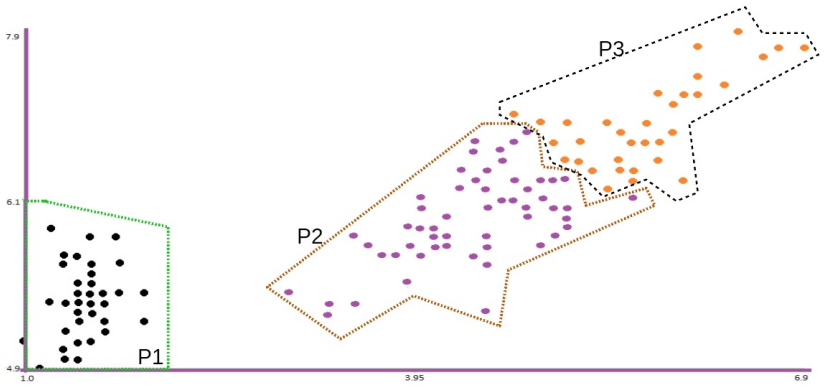


Figure 5.4 – Un partitionnement appliqué aux données de fleurs d'iris.

Le K-moyennes [Macqueen, 1967], une variante de la méthode des centres mobiles [Forgy, 1965], est un des algorithmes les plus simples. L'idée principale est de choisir arbitrairement un ensemble de centres de partition et de rechercher itérativement le partitionnement optimal. L'algorithme sélectionne d'abord de manière aléatoire le centre de k partition autour desquels sont regroupés les objets les plus proches de ces centres. Il calcule ensuite le nouveau centre de chaque partition puisqu'il peut se changer après l'affectation des objets. Cette opération est répétée jusqu'à ce que la dispersion des membres de chaque partition soit minimale.

À chaque itération, les partitions deviennent plus compactes, conduisant à la convergence de l'algorithme. Cependant l'optimum local produit par la méthode pose le problème de l'initialisation. Une solution est de lancer plusieurs fois l'algorithme en prenant les moyennes aléatoirement à chaque fois, puis de comparer leur mesure de distorsion. Au final, le partitionnement qui possède la distorsion minimale est conservé.

5.2 Fouille de donnés sémantiques

Au cours de la dernière décennie, de nombreuses approches ont été proposées pour appliquer la fouille aux données sémantiques. Nous pouvons distinguer trois approches :

— **Approches basées sur la logique de description** : Ces approches visent à adapter les algorithmes existants au nouveau format de représentation des données. En effet, il est possible de modifier les algorithmes de fouille traditionnels pour traiter des données sémantiques en utilisant la logique de description comme un format de représentation de connaissances [Jozefowska *et al.*, 2010].

— **Fouille des graphes** : Si l'on considère les données RDF comme un graphe où les ressources sont reliées par des prédicats comme des arêtes, un autre domaine de recherche connexe consiste à extraire des sous-graphes [Kuramochi et Karypis, 2001] ou des sous-arbres fréquents [Chi *et al.*, 2004].

— **Approches statistiques** : Ces approches visent à transformer les données RDF en structure de données classique utilisée par ces algorithmes.

Comme nous souhaitons implémenter un fonctionnement simple et efficace pour faciliter l'analyse des données environnementales, il est préférable de réutiliser les algorithmes existants et approuvés dans le domaine. Dès lors, les approches statistiques conviennent à nos besoins car elles ne consistent pas à introduire de nouveaux algorithme comme le cas de la fouille des graphes ou à modifier les algorithmes existant comme le cas de des approches basées sur la logique de description.

5.2.1 Approches statistiques

Dans ces approches, grâce à ses connaissances d'experts, l'utilisateur définit des cibles de la fouille et doit formuler des requêtes SPARQL pour sélectionner les données désirées. Celles-ci sont converties dans un format adapté aux algorithmes de fouilles. Dans la plupart des cas, les connaissances d'experts sont requises pour construire ces données. L'utilisateur doit formuler des requêtes SPARQL pour générer des attributs qui sont habituellement obtenus par des agrégats binaires ou numériques. La génération entièrement automatique n'est pas possible [Ristoski et Paulheim, 2016].

Un des premiers travaux est SPARQL-ML [Kiefer *et al.*, 2008]. Il s'agit d'une extension du langage SPARQL avec une déclaration spécialisée afin de pouvoir apprendre un modèle par des méthodes de classification et de régression.

Une approche similaire a été utilisée dans le plugin Semweb [M. A. Khan et Dengel, 2010] de RapidMiner [1], qui prétraite les données RDF de manière à pouvoir être traitées ultérieurement par un outil de fouille, RapidMiner dans ce cas. Là encore, l'utilisateur doit spécifier une requête SPARQL pour sélectionner les données d'intérêt, qui sont ensuite converties en vecteurs de caractéristiques.

LiDDM [Narasimha *et al.*, 2011] est un système pour l'exploration de données provenant de données liées. L'outil permet d'effectuer des requêtes SPARQL vers de multiples sources. Les données obtenues peuvent être ensuite utilisées par des techniques d'apprentissage automatique. Le traitement de données se compose de l'intégration, du filtrage et de la segmentation de données. Il se réalise de manière manuelle par l'utilisateur.

[Fleischhacker *et al.*, 2012] ont proposé une approche de fouille des données RDF pour divers types d'axiomes de propriété grâce aux règles d'association. L'approche offre des moyens efficaces pour enrichir les bases de connaissances.

1. http://www.rapidminer.com/

[Nebot et Berlanga, 2012] ont présenté une nouvelle méthode pour extraire des règles d'association à partir de données sémantiques. À partir des connaissances du schéma, c'est-à-dire de la Tbox, codées dans l'ontologie, les transactions appropriées sont dérivées pour alimenter ensuite les algorithmes de fouille des règles d'association traditionnels. Ce processus est guidé par l'analyste qui spécifie des requêtes sémantiques. Le système est décrit par la Figure 5.5.

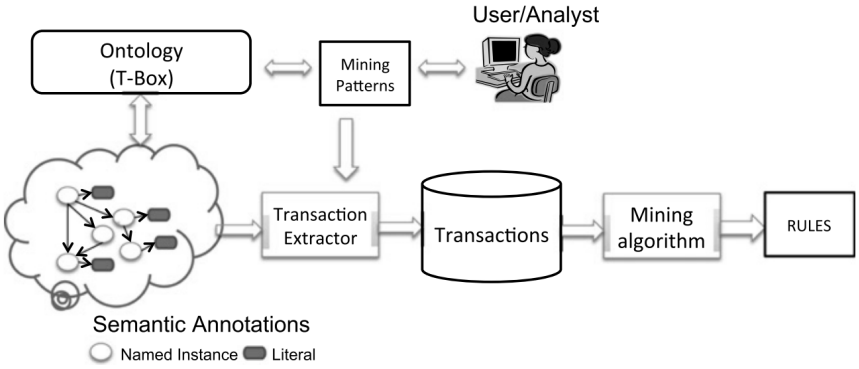


Figure 5.5 – Système de fouille de données sémantiques [Nebot et Berlanga, 2012].

[Ristoski *et al.*, 2015] ont développé une extension pour RapidMiner permettant d'appliquer toutes les étapes du processus de l'extraction de connaissances aux données ouvertes et liées, telles que la liaison, la combinaison de données provenant de plusieurs sources, le prétraitement et le nettoyage, la transformation, l'analyse de données et l'interprétation des résultats d'exploration de données. Il s'agit d'un outil non-supervisé, l'utilisateur n'a pas besoin de connaître SPARQL ou RDF au préalable.

5.2.2 Rôles de l'ontologie dans l'extraction de connaissances

Les ontologies peuvent jouer différents rôles dans le domaine de l'extraction de connaissances. [Nigro *et al.*, 2007] divisent les ontologies utilisées en trois catégories :

— **Ontologies pour le processus de fouille de données** : Ces ontologies définissent les connaissances nécessaires pour effectuer les processus de fouille de données, telles que les étapes et les paramètres possibles. Elles sont souvent utilisées pour aider l'utilisateur à créer des processus de fouille appropriés, par exemple en assurant qu'un algorithme choisi est capable de gérer les données.

— **Ontologies de métadonnées** : Ces ontologies décrivent les connaissances sur les données, telles que les informations sur la provenance ou les processus utilisés pour construire certains ensembles de données.

— **Ontologies de domaine** : Ces ontologies représentent les connaissances d'arrière-plan sur le domaine d'application, c'est-à-dire le domaine sur lequel l'extraction de données est exécuté. Elles sont utilisées pour décrire les sources de données ou les variables choisies pour la fouille.

Les ontologies de domaine correspondent à notre contexte dans lequel nous souhaitons appliquer des techniques de fouille aux données modélisées par ce type d'ontologie. Par conséquent, nous ne détaillerons que leurs rôles dans le processus de l'extraction de connaissances.

Sélection

Afin de mieux comprendre le domaine d'application et les méthodes de fouille qui conviennent aux données, il est nécessaire de comprendre les données. Tout d'abord, il est souhaitable de comprendre quel est le domaine d'application, quelles connaissances sont capturées et quelles sont les connaissances supplémentaires qui pourraient être extraites à partir de ces données. Ensuite, on peut identifier l'objectif de la fouille plus facilement et sélectionner un échantillon de données approprié. Dans de nombreux cas, l'utilisateur doit posséder des connaissances spécifiques du domaine afin de mieux comprendre les données. Pour cela, il est possible d'utiliser des technologies du Web sémantique pour une meilleure représentation et exploration de données en exploitant des ontologies spécifiques du domaine.

Prétraitement

Les ontologies aident à prétraiter les données, principalement pour augmenter leur qualité. En effet, tout d'abord, les valeurs aberrantes ou erronées peuvent être identifiées grâce aux contraintes définies par les ontologies. Ensuite, les valeurs manquantes peuvent être déduites et/ou remplies à l'aide d'autres sources telles que les données liées, tout en se basant sur les concepts et les relations modélisés dans les ontologies.

Évaluation et interprétation

Les modèles ontologiques peuvent aider à l'interprétation des modèles trouvés, en particulier pour les tâches descriptives. Celles-ci englobent généralement des partitions trouvées ou des modèles de règles décrivant un ensemble de données. Les informations provenant des données liées et/ou des ontologies peuvent aider à analyser davantage ces résultats, par exemple en expliquant les caractéristiques typiques des instances d'une partition. Elles peuvent donc expliquer le partitionnement choisi par les algorithmes de fouille. En outre, les règles peuvent être perfectionnées et/ou généralisées, ce qui améliore leur interprétation.

5.2.3 Synthèse

La meilleure façon d'unifier l'apprentissage automatique et le Web sémantique est de se concentrer sur RDF [Bloem et De Vries, 2014]. Ces auteurs ont

proposé un pipeline pour passer de RDF à un modèle d'apprentissage automatique (Figure 5.6). D'après les auteurs, il existe deux méthodes dans lesquelles l'apprentissage automatique peut être appliqué aux données RDF : l'apprentissage par graphe et l'apprentissage basé sur des caractéristiques (approches statistiques).

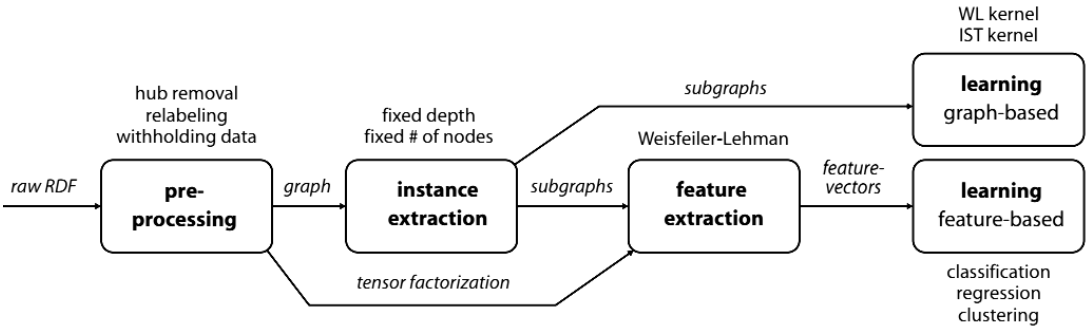


Figure 5.6 – Un pipeline pour passer de RDF à un modèle d'apprentissage automatique [Bloem et De Vries, 2014].

On présume que la première méthode est plus attrayante et à long terme plus opportuniste, car elle exploite l'apprentissage basé sur les graphes à partir de données RDF, comme celles-ci couvrent un graphe de ressources avec arêtes représentant les prédicats. Cependant, l'apprentissage par graphe a ses propres inconvénients. Il est important de noter que, comme deux nœuds différents dans un graphe RDF n'ont pas la même URI, la fouille de graphes serait limitée à l'exploration de concepts et non à la fouille de données [Abedjan et Naumann, 2013]. Par conséquent, la deuxième méthode devient la plus simple à implémenter. Les données RDF sont transformées en forme tabulaire qui sert ensuite à la génération de caractéristiques auxquelles des algorithmes peuvent être directement appliqués. Il s'agit aussi d'une méthodologie pratiquée pour développer de nombreuses extensions chez RapidMiner comme dans [M. A. Khan et Dengel, 2010, Potoniec et Lawrynowicz, 2011, Nolle *et al.*, , Ristoski *et al.*, 2015] en vue d'appliquer des méthodes de fouille de données traditionnelles aux données RDF provenant d'une source locale ou des données liées.

Conclusion du chapitre

Ce chapitre est consacré à la présentation du processus d'extraction de connaissances à partir de données sémantiques. Tout d'abord, l'ECD et plus spécialement la fouille de données a été étudiée. Pour chaque technique de fouille, un algorithme typique a été présenté. Ensuite, nous avons présenté les approches de fouille de données sémantiques. Parmi les approches étudiées, l'approche statistique transformant les données RDF en forme tabulaire est la plus adaptée à notre contexte. En effet, la transformation est le point clé pour adapter la fouille de données classique aux données sémantiques. Cela permet de réutiliser des algorithmes de fouille existants et approuvés par la communauté. Ces algorithmes et/ou leurs variantes sont incorporés dans la plupart des outils. Nous pouvons considérer des outils *open-sources* et réutilisables comme Weka [a], KEEL [b] ou SPMF [c]. Un outil de ce type sera utilisé pour le développement de notre prototype. À noter qu'il est souhaitable de bien comprendre les paramètres de ces algorithmes pour obtenir de bonnes analyses.

Enfin, nous avons décrit les rôles des ontologies du domaine dans l'extraction de connaissances à partir de données sémantiques. À partir de ces études, nous introduisons une approche visant à mieux incorporer et exploiter les technologies du Web sémantique dans l'extraction de connaissances à partir d'une base de connaissances. L'approche, détaillée dans 7.4, propose de former une boucle commençant par l'enrichissement de la base jusqu'à la visualisation et validation du résultat.

a. http://www.cs.waikato.ac.nz/ml/weka/
b. http://sci2s.ugr.es/keel/
c. http://www.philippe-fournier-viger.com/spmf/

Deuxième partie

Proposition

Chapitre 6

Une ontologie pour l'environnement

Sommaire

6.1 Contexte d'application 117
6.1.1 Zone atelier Plaine et Val de Sèvre 117
6.1.2 Corpus de données 118
6.1.2.1 La base d'assolement 118
6.1.2.2 Les données de biodiversité 121
6.1.3 Expression de besoins d'analyse 122
6.2 Une ontologie pour l'environnement 123
6.2.1 Représentation de l'information spatio-temporelle . . . 124
6.2.2 Réutilisation des ontologies du temps et de l'espace . . 125
6.2.2.1 Réutilisation de l'ontologie de temps 126
6.2.2.2 Réutilisation de l'ontologie de l'espace 128
6.2.3 Conception et développement des ontologies locales . . 128
6.2.3.1 Une ontologie pour l'assolement 128
6.2.3.2 Une ontologie pour les observations des oiseaux 129
6.2.3.3 Une ontologie pour les observations des nids 129
6.2.3.4 Une ontologie pour les observations des campagnols 130
6.2.4 Une ontologie pour l'environnement 130
6.2.5 Études de cas 134

Introduction du chapitre

L'observation des pratiques agricoles sur le long terme étant une nécessité pour la recherche sur des questions environnementales, le CNRS de Chizé a mis en place un observatoire des assolements et des biodiversités sur la zone atelier Plaine et Val de Sèvre, en partie financé par le CNRS. Ce chapitre présente tout d'abord la zone atelier et les jeux de données qui constituent notre contexte d'application. La première partie est consacrée à la présentation de la zone atelier, les besoins d'analyse et d'exploitation des données des experts du domaine.

Nous utilisons ce cas d'étude pour justifier nos approches qui ont pour but de résoudre le problème d'intégration et d'exploitation de données spatio-temporelles hétérogènes. Dès lors, une ontologie jouant le rôle d'un schéma global est développée pour cette intégration. Afin de prendre en compte l'aspect spatio-temporel et le représenter à l'aide des ontologies, nous nous appuyons sur l'approche perdurantiste en réutilisant des ontologies du temps et des ontologies de l'espace. Enfin, nous décrivons le développement de ces ontologies en vue d'aboutir à une ontologie pour l'environnement.

6.1 Contexte d'application

Dans les zones rurales avec une prédominance d'activités agricoles, les études sur les questions environnementales telles que la préservation de la biodiversité, l'érosion des sols par ruissellement, ou la pollution de l'eau peuvent bénéficier de l'analyse sur le long terme de la mosaïque des cultures résultant des pratiques agricoles. En effet, les paysages agricoles sont principalement le résultat des décisions des agriculteurs portant sur le choix des cultures et de leur répartition à l'échelle de l'exploitation. L'agencement, la forme et la nature des cultures structurent l'organisation spatiale du paysage qui a une incidence sur les processus écologiques à différentes échelles [Lazrak *et al.*, 2010, Schaller *et al.*, 2012]. Cette organisation spatiale du paysage évolue dans le temps parce que les agriculteurs modifient l'assolement de leurs parcelles et éventuellement les limites de ces parcelles chaque année.

La nécessité de collecter des observations sur une longue durée en vue d'effectuer des recherches sur les relations entre l'environnement et l'anthroposystème a entraîné la mise en place de zones ateliers par le CNRS. La spécificité des zones ateliers réside à la fois dans la taille de l'objet d'étude, qui est de dimension régionale, et dans la façon de mener les recherches, avec une expérimentation des hypothèses et alternatives en grandeur nature. Leur problématique est celle des interactions entre un milieu et les sociétés qui l'occupent et l'exploitent.

Dans cette partie, nous présentons la zone atelier Plaine et Val de Sèvre, les jeux de données hétérogènes et les besoins d'intégration et d'exploitation de ces sources des experts du domaine. Ces éléments constituent notre contexte d'application.

6.1.1 Zone atelier Plaine et Val de Sèvre

Appartenant au réseau des zones ateliers national, la zone atelier Plaine et Val de Sèvre couvre 450 km^2 au sud de la ville de Niort, dans le département des Deux-Sèvres, France (Figure 6.1 [1]). Il s'agit essentiellement d'une plaine de cultures céréalières intensives : céréales, maïs, tournesol, pois et colza où les activités d'élevage sont encore présentes mais en forte baisse. Les parcelles agricoles sont encore de taille modeste (4-8 ha) et 15% d'entre elles sont occupées par des prairies (artificielles, permanentes ou temporaires). Ainsi, il s'agit d'un dispositif totalement unique en France, de par ses problématiques et l'ampleur (dans l'espace et dans le temps) des données collectées, dont en particulier l'occupation des sols (19 000 parcelles pendant 20 ans).

La problématique de recherche menée sur la zone atelier peut se décomposer en 3 axes principaux [Bretagnolle *et al.*, 2012] :

— **Fonction observatoire** : Le plus grand défi lié à la mise en place de

1. http://za-geminat.cnrs.fr/wp-content/uploads/2015/08/Carte_ZAPVS.jpg

réseaux de sites de recherches à long terme est de fournir des réponses à la société sur les impacts des changements globaux sur la structure et le fonctionnement des écosystèmes, les modifications de l'environnement à différentes échelles spatiales, la dégradation des ressources et la perte de la biodiversité et des services associés.

— **Recherche sur la biodiversité** : On a mis en œuvre un programme de recherche autour de la dynamique spatiale et temporelle de la biodiversité en paysage hétérogène et perturbé ; et du rôle de la biodiversité dans l'expression et le maintien de certains services écosystémiques (production, pollinisation, régulation des cycles biogéochimiques etc.).

— **Recherche des compromis** : Enfin, un dernier front de recherche concerne la question des compromis à trouver entre les différents services écosystémiques liés à l'agriculture, à la biodiversité et à sa conservation.

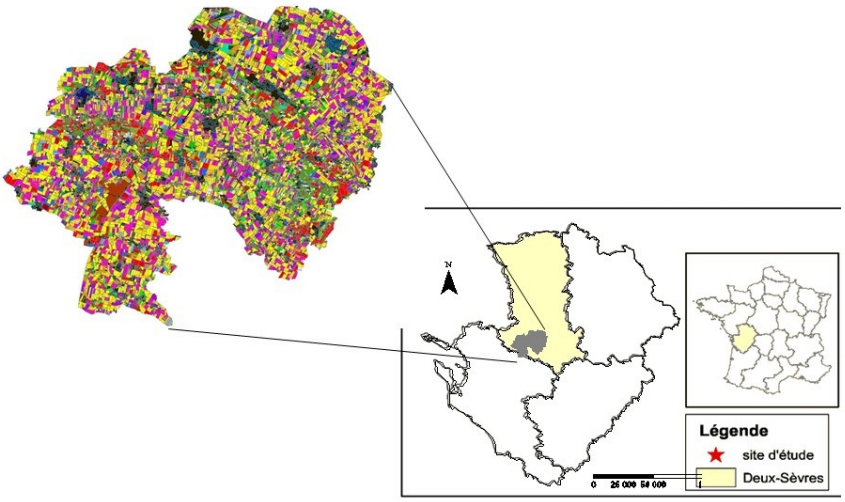


Figure 6.1 – La zone atelier Plaine et Val de Sèvre.

6.1.2 Corpus de données

Depuis plus de vingt ans, plusieurs bases de données ont été recueillies par l'équipe Agripop (CNRS Chizé). Ces données peuvent être catégorisées en deux groupes : la base d'assolement et les bases de biodiversité (plus d'informations peuvent être trouvées sur le site de la zone atelier [2]).

6.1.2.1 La base d'assolement

L'organisation spatiale du paysage évolue dans le temps parce que les agriculteurs modifient l'assolement de leurs parcelles chaque année, mais également

2. http://www.za.plainevalsevre.cnrs.fr/

recomposent parfois les parcelles entre elles changeant ainsi les formes des parcelles. Depuis 1994, les occupations du sol sont donc relevées annuellement sur le terrain et numérisées sur les 19 000 parcelles agricoles. Ces données sont centralisées dans une base de données nommée «Assolement».

Dans cette base de données, une parcelle agricole est considérée comme une unité de gestion, un polygone entouré par des entités ayant différentes cultures dans les années successives, généralement quatre. Une parcelle est délimitée par des limites physiques telles qu'une route, une rivière, un chemin de champ ou une limite d'un seul champ. Elle ne contient qu'un seul type de culture. Elle diffère de la parcelle cadastrale, mais également des blocs stockés dans le RPG (Registre Parcellaire Graphique) qui sont mis à jour tous les deux ans et distribués par l'IGN, disponibles ici [3]. Le RPG contient des informations pertinentes pour le suivi des propriétaires des parcelles, mais les limites des blocs ne correspondent pas aux parcelles observées sur le terrain. En fait, les agriculteurs souvent divisent leur bloc en plusieurs parcelles, mais seulement la surface de chaque bloc est déclarée avec la nature de la culture en pourcentage.

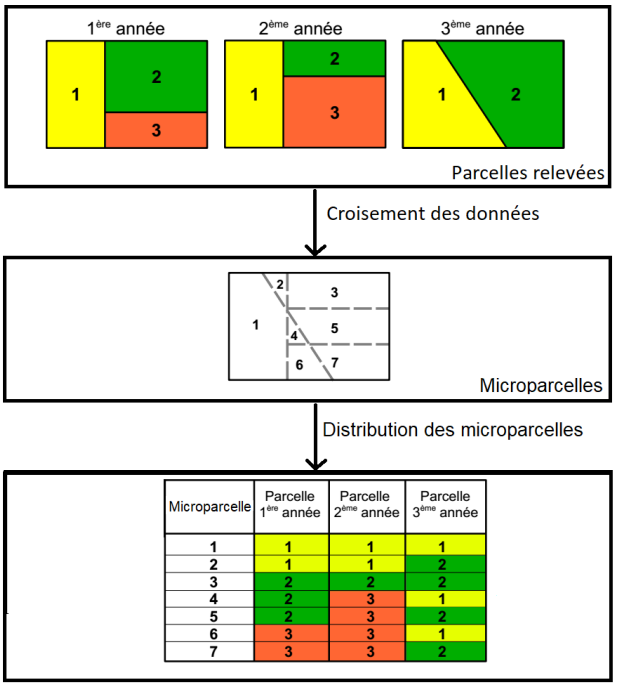


Figure 6.2 – Relations Parcelle-Microparcelle dans le modèle des composites spatio-temporels.

Le modèle de données de la base se fonde sur le modèle des composites spatio-temporels proposé par [Langran et Chrisman, 1998]. L'idée consiste à ne pas stocker la géométrie de chaque parcelle pour chaque année, mais à utiliser dans le

3. http://www.geoportail.gouv.fr/donnee/251/registre-parcellaire-graphique-rpg-2012

modèle de petites géométries, appelées ici *microparcelles*, obtenues par l'intersection de toutes les parcelles au cours de la période d'observation. La géométrie de toutes les parcelles peut être reconstruite à la volée pour chaque année en utilisant une composition des microparcelles constituant la parcelle. Par exemple, la figure 6.2 représente comment construire des microparcelles et les distribuer aux parcelles d'origine. L'intersection des parcelles d'origine produit 7 microparcelles. Les microparcelles 1, 2, 4 et 6 constituent la première parcelle alors que l'union des 3, 5 et 7 forme la deuxième de la troisième année.

La figure 6.3 montre le modèle utilisé pour la gestion des données de l'assolement. La classe d'association *Contains* permet de mémoriser les changements de la parcelle en faisant le lien entre les parcelles (*Parcel*) et les micro-parcelles (*Microparcel*). Les classes *Farmer* et *Association* permettent d'identifier les exploitants (au sens aussi de propriétaire) des parcelles à des dates différentes. La classe *Association* regroupe un à plusieurs exploitants (*Farmer*). Les parcelles sont régies par différents types de contrats, correspondant à un engagement des exploitants. La rotation des cultures est décrite par la relation datée *Characterizes* entre la parcelle et des types de culture (*Culture*), décrites par un code *OCS_code* et un nom *name*. La nomenclature utilisée pour décrire les cultures est plus fine que celle du RPG, et sa stabilité est garantie par l'équipe de recherche.

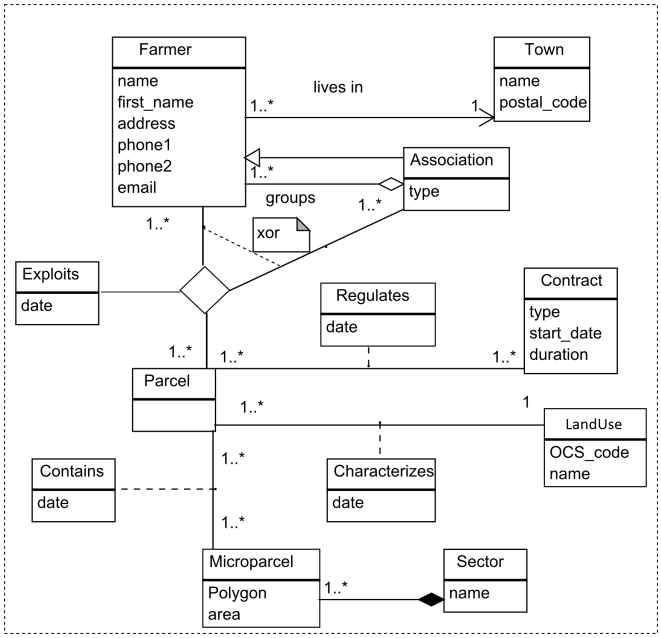


Figure 6.3 – Modèle des composites spatio-temporels pour l'évolution de parcelles.

Le modèle a été appliqué depuis l'année 1994. Par conséquent, il n'est plus d'actualité et possède certains inconvénients. D'après le gestionnaire du système, si ce modèle comporte des avantages substantiels pour l'analyse des données, il

ne peut être manipulé sans un minimum de développements par les utilisateurs, qui devraient sinon à la main reconstituer la couche fusionnée lors des mises à jour des formes de parcelles chaque année. Il n'existe pas d'outils sur étagère capable de réaliser ces opérations. Le modèle utilisé a été révisé en 2017 mais le changement n'est pas encore mis en pratique.

6.1.2.2 Les données de biodiversité

Parallèlement, des données d'avifaune sont collectées sur le terrain depuis plusieurs années et rassemblées dans une autre base de données «Oiseaux» qui est structurée suivant un schéma relationnel spatial, implémenté dans PostgreSQL avec PostGIS. Ces données, ponctuelles et datées, proviennent des différents chercheurs qui rapportent leurs observations concernant 600 espèces grâce à une interface Web. Pour les oiseaux, la base constitue une collection d'observations décrivant le comportement des espèces observées ainsi que leurs nids, et le contexte des observations (hauteur de végétation, date, heure, localisation, etc.).

Il existe par ailleurs d'autres données structurées sur différentes espèces, souvent dans des tableurs, ou bien des bases de données MS Access. Il serait souhaitable de pouvoir interroger et croiser aussi ces sources avec la connaissance de l'assolement et de la faune avicole. C'est le cas des données relatives à l'observation des carabes, petits coléoptères auxiliaires des champs très sensibles à la qualité des milieux, qui bénéficient d'un suivi depuis 9 ans dans une base MS Access. Il existe également un ensemble de 10 000 observations de trois espèces de micromammifères dans une base MS Access qui sont de bons indicateurs des pratiques agricoles : campagnol des champs, mulot sylvestre et musaraigne musette. Ces bases MS Access sont dans les deux cas plutôt une collection de fichiers tabulaires sans contraintes relationnelles, support à l'origine de formulaires de saisie qui ne sont plus systématiquement utilisés.

Par rapport à ces données, les modalités d'acquisition des données ont également suivi des évolutions technologiques, au rythme de l'évolution des usages de l'informatique. Partant de simples formulaires papier remplis sur le terrain puis saisis sur des fichiers Excel au retour du terrain, la zone atelier a progressivement mis en place des améliorations, pour aboutir à l'architecture actuelle du système d'information (Figure 6.4). Seulement deux des bases de données, Assolement et Oiseaux, bénéficient d'une base de données bien structurée sous PostgreSQL et d'outils adéquats pour la saisie terrain et l'administration des informations. Les autres bases de données (Micromammifères, Carabes par exemple) mériteraient tout autant cet effort d'intégration, cependant cela représente une lourde charge de travail.

La problématique de l'acquisition de données sur le terrain dans des systèmes informatiques soulève toujours des questions de soutenabilité des développements, car les protocoles sont en constante évolution et les personnels dédiés sont, eux, très peu pérennes. Elle soulève également des questions d'investisse-

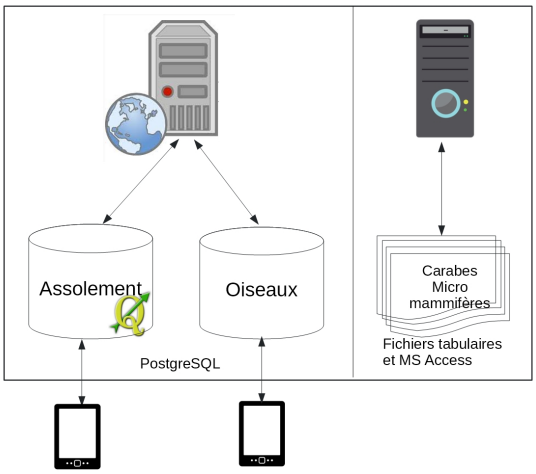


Figure 6.4 – Architecture du système de collecte de données.

ments matériels et technologiques, de vécu du changement pour les personnels en charge de l'acquisition des données qui sont loin d'être résolues. C'est ainsi qu'en définitive, la plupart des données de biodiversité collectées se retrouvent dans des formats très hétérogènes, et qu'un besoin très important d'harmonisation émerge pour leur analyse.

6.1.3 Expression de besoins d'analyse

Afin de faciliter les analyses transversales entre évolution du paysage et variations de la biodiversité, il est souhaitable de croiser les informations de la base de données d'Assolement avec celles des bases de données des biodiversités. En effet, un nombre conséquent d'analyses peuvent-être menées à partir de ces données :

Vérification de données

Dans un premier temps, l'analyse peut être utilisée pour vérifier l'ensemble des données collectées. Lors de la rotation des cultures, les experts peuvent décrire un certain nombre de règles de succession afin d'éliminer ou de corriger les valeurs douteuses. Par exemple, les successions de cultures peu probables comme «Tournesol-Tournesol» ou «Tournesol-Colza», ainsi que la disparition de bois ou l'urbanisation des parcelles dans la zone atelier peuvent être détectées et examinées.

Découverte de pratiques agricoles

Ensuite, les événements territoriaux tels que la fusion, l'intégration, la scission, l'extraction, la réallocation et la rectification (Figure 6.5) peuvent être détectés. L'analyse de ces événements permet de découvrir la corrélation entre l'organisation du paysage et la pratique agricole.

Par ailleurs, il est possible de découvrir des relations entre les cultures plantées dans les parcelles voisines. Dans la pratique en effet, l'agriculteur

alloue une culture à une parcelle en tenant compte de son voisinage. En d'autres termes, les cultures sont organisées dans l'espace et dépendent de leur voisinage proche.

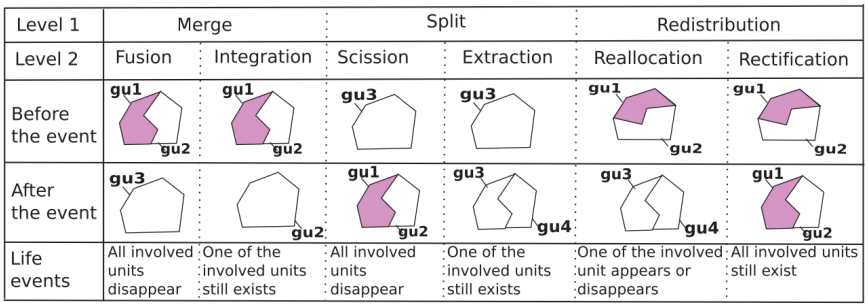


Figure 6.5 – Les événements territoriaux [Plumejeaud *et al.*, 2011].

Croisement de données

On souhaite aussi rechercher les relations entre données provenant de différentes sources.Premièrement, on peut réaliser des analyses entre les observations des espèces et le type de culture des parcelles afin de découvrir des préférences animales pour un certain type de culture, ou de configuration spatiale des cultures, des parcelles, voire même une configuration spatio-temporelle des cultures. En effet, le rythme et le type de rotation pourrait très bien être appréciés des oiseaux comme favorables ou non à la nidification.

Deuxièmement, on peut examiner les relations entre différentes observations d'espèces pour mieux comprendre leurs rôles dans la chaîne alimentaire, par exemple, l'abondance des campagnols attrapés en fonction des nids de busards aux alentours.

Ces analyses nécessitent des requêtes croisant les données de différentes bases de données et la capacité de raisonnement sur les relations spatio-temporelles entre les relevés et les observations (Figure 6.6). Ces besoins d'analyse correspondent bien à notre étude qui vise à proposer une approche ontologique pour l'intégration et l'exploitation de données hétérogènes.

6.2 Une ontologie pour l'environnement

À partir de cas d'étude, nous souhaitons construire une ontologie spatio-temporelle pour l'intégration des sources données présentés. Pour cela, comme il n'existe aucune ontologie a priori pour ces sources, on construit d'abord une ontologie pour chacune d'entre elles. Ensuite l'alignement ou la fusion d'ontologies peut être éventuellement appliqué pour trouver des correspondances entre

ces ontologies. Les ontologies locales avec leur correspondances constituent des éléments nécessaire pour former l'ontologie globale à la fin.

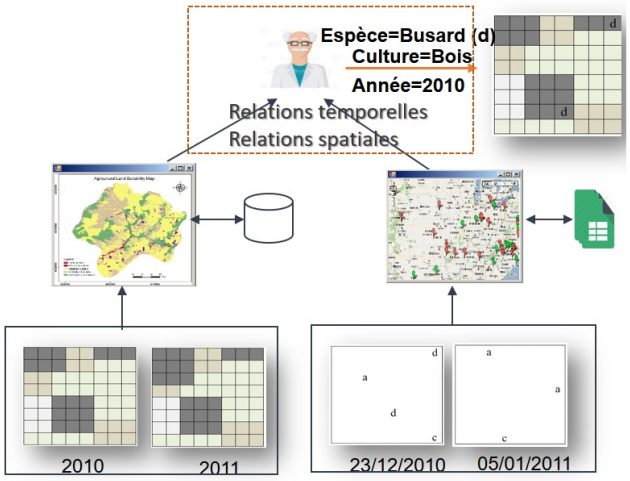


Figure 6.6 – Intégration de deux sources de données avec un raisonnement spatio-temporel.

6.2.1 Représentation de l'information spatio-temporelle

Afin de prendre en compte l'aspect spatio-temporel et le représenter par une ontologie, nous nous appuyons sur l'approche perdurantiste en réutilisant les parties spatiales et temporelles des ontologies. En effet, nous considérons que les objets spatio-temporels possèdent une ou plusieurs tranches de temps (*timeslices*) correspondant aux différents états à travers leur vie. Ainsi, la classe *Timeslice* est introduite. Elle est liée à la classe des objets spatio-temporelles et se compose de trois composantes (Figure 6.7) :

— **Composante sémantique** : Cette composante représente l'ensemble de propriétés décrivant les caractéristiques de l'entité durant la période de validité du *timeslice*.

— **Composante temporelle** : Cette composante représente une unité de temps durant lequel le *timeslice* est valide.

— **Composante spatiale** : Cette composante représente une description géométrique de l'entité.

Les composantes temporelle et spatiale doivent impérativement être décrites pour chacune des *timeslices*. La composante sémantique, en revanche, est facultative dans la description du *timeslice*.

Dans les paragraphes suivants, nous présentons la conception de notre ontologie spatio-temporelle pour l'environnement à partir des ontologies locales grâce à

la réutilisation des ontologies du temps et des ontologies de l'espace des travaux antérieurs.

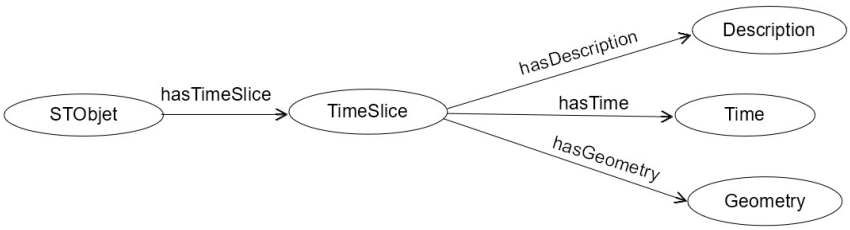


Figure 6.7 – Représentation d'un *timeslice*.

6.2.2 Réutilisation des ontologies du temps et de l'espace

En vue d'aboutir à une modélisation spatio-temporelle complète et donc un raisonnement fiable, nous devons considérer, pour chaque type de concept spatial et temporel, la possibilité de le référencer et de réutiliser des modèles ontologiques existants, de préférence ceux qui sont testés.

La réutilisation des ontologies existantes permet de faciliter la modélisation et d'améliorer le raisonnement. L'objectif de la réutilisation est l'élaboration d'ontologies de meilleure qualité, tout en réduisant les coûts de développement. En effet, la construction d'une nouvelle ontologie est un travail compliqué qui consomme du temps. Toutefois, le processus de réutilisation doit être bien étudié pour choisir l'opération adéquate entre la fusion et l'intégration des ontologies existantes.

D'une manière générale, la réutilisation des ontologies peut être partielle ou totale. La première correspond à l'intégration des ontologies et la deuxième à la fusion des ontologies [Pinto et Martins, 2000]. Dans le cas de la réutilisation partielle, la construction se fait en assemblant, étendant, spécialisant et adaptant d'autres ontologies qui font partie de l'ontologie résultante. Dans le cas de la réutilisation totale, la construction se fait en fusionnant les différentes ontologies portant sur le même sujet ou un sujet semblable en une seule qui unifie tous. Dans ce travail, nous adoptons l'approche de la réutilisation partielle car d'après les expériences de [Pinto et Martins, 2000], pourvu qu'un ensemble de petites ontologies modulaires, hautement réutilisables ne soient disponibles, de grandes ontologies à des fins spécifiques peuvent être plus facilement assemblées. D'autre part, l'approche de la réutilisation totale entraîne des conséquences négatives sur le processus d'inférence [Mefteh, 2013].

Vu le besoin de considérer les concepts et les opérateurs spatiaux et temporels dans notre travail, nous avons proposé la solution de la réutilisation des ontologies standards. Avant de pouvoir importer et réutiliser des ontologies existantes, nous suivons une démarche en trois étapes :

1. Énumérer les concepts et les relations nécessaires pour notre cas d'étude qui peuvent être assimilés à des concepts et des relations temporels et spatiaux.
2. Identifier une ontologie dans le domaine du temps et une de l'espace permettant la réutilisation. Celles-ci doivent contenir les concepts et les relations énumérés dans l'étape précédente.
3. Intégrer ou fusionner l'ontologie choisie à l'ontologie d'origine.

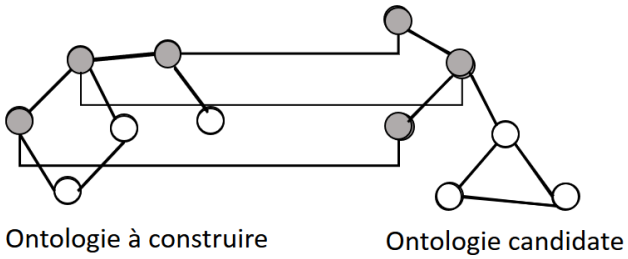


Figure 6.8 – Réutilisation de l'ontologie.

Dans le contexte de l'intégration sémantique de données, il est recommandé d'étudier l'ensemble des sources à intégrer plutôt qu'une seule source. Ainsi, on obtient une vue globale sur tous les concepts et les relations nécessaires pour bien choisir les ontologies à réutiliser en une seule fois. Par exemple, si on ne considère que la base des observations des oiseaux, il suffit de réutiliser le Basic Geo Vocabulary car cette source ne contient que des points d'observation comme géométrie. Pourtant, si les autres bases sont aussi considérées en même temps, on peut identifier d'autres concepts tels que polygone, multi-polygone ou éventuellement ligne ou multi-ligne, eux, qui n'existent pas dans ce vocabulaire.

Afin d'intégrer l'ontologie choisie à l'ontologie d'origine, on peut s'appuyer sur la mise en correspondance des concepts. Pour cela, le constructeur *owl:equivalentClass* est utilisé pour indiquer que les deux classes reliées ont les mêmes instances. Pourtant, afin de simplifier la modélisation, il est possible de référer directement aux concepts temporels et spatiaux des ontologies réutilisées. De cette façon, il n'est plus nécessaire de nommer des concepts dans l'ontologie d'origine. Par exemple, la Figure 6.1 montre comment on peut intégrer l'ontologie OWL-Time en mettant en correspondance deux concepts de temps *Observation_Date* et *Instant*, ou en mettant une référence directe au concept *Instant* pour représenter la date de l'observation.

6.2.2.1 Réutilisation de l'ontologie de temps

L'application du principe de définition des besoins pour la réutilisation d'une ontologie du temps fait ressortir les deux concepts temporels : instant et intervalle. L'identification des relations temporelles conduit à la considération de l'ensemble des relations de l'algèbre temporelle d'Allen [Allen, 1983] entre les

intervalles de temps. Les 13 relations d'Allen permettent de répondre à des questions sur la proximité temporelle de deux phénomènes, à condition d'employer pour les intervalles la même granularité. En plus, nous devons exprimer et modéliser la relation *inside* entre un instant et un intervalle qui est indispensable pour le croisement de nos bases de données. En effet, une observation d'oiseau est datée par un instant, qui se situe dans l'intervalle d'existence d'une culture sur une parcelle.

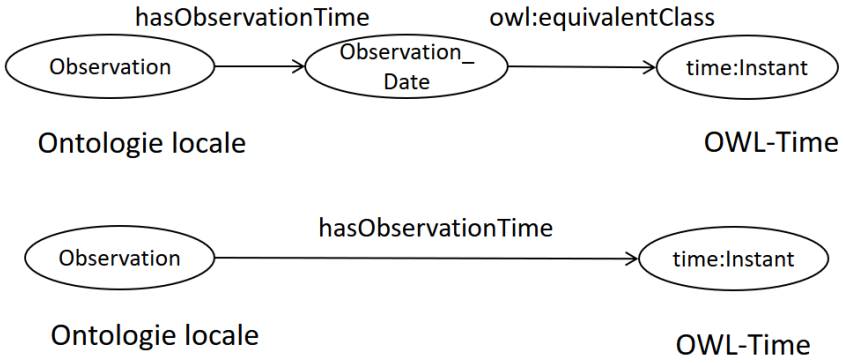


Tableau 6.1 – Exemple d'une intégration de l'ontologie OWL-Time.

À partir des ontologies temporelles étudiées dans la section 4.2.1, étant une recommandation du W3C, l'ontologie OWL-Time est choisi pour la réutilisation parce qu'elle permet la représentation de ces deux concepts temporels, *Instant* pour les instants et *ProperInterval* pour les intervalle, et leurs relations.

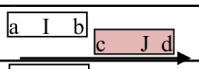
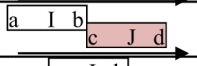
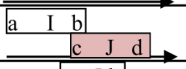
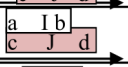
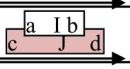
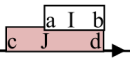
Relation	Constraint	Schema	Inverse
before [b]	b<c		after [a]
meets [m]	b=c		met-by [mi]
equals [eq]	a=c ∧ b=d		
overlaps [o]	d ∧ b		overlapped-by [oi]
starts [s]	a=c ∧ b<d		started-by [si]
during [d]	c ∧ d		contains [di]
finishes [f]	b=d ∧ c<a		finished-by [fi]
inside [i]	c<x<d		

Tableau 6.2 – Relations temporelles entre les entités spatio-temporelles.

Les relations temporelles identifiées entre les entités spatio-temporelles sont résumées par le Tableau 6.2. Soient I=[a, b], J=[c, d] deux intervalles et P=x un

instant, 14 relations sont prises en considération. Les 13 premières proviennent de l'algèbre d'Allen pour représenter des relations temporelles entre les intervalles, alors que la dernière représente une relation entre un instant et un intervalle qui est nécessaire pour le croisement des bases de données. À noter qu'ils existent d'autres relations entre ces derniers mais elles sont peu importantes pour notre modélisation. Les 14 relations décrites sont incluses dans l'ontologie OWL-Time.

6.2.2.2 Réutilisation de l'ontologie de l'espace

L'application du principe de définition des besoins pour la réutilisation d'une ontologie de l'espace fait ressortir les concepts spatiaux : point, ligne, polygone ainsi que les concepts multi-points, multi-lignes et multi-polygones. L'identification des relations spatiales conduit à la considération de l'ensemble des relations spatiales : *Equals, Disjoint, Intersects, Overlaps, Contains, Crosses, Within* et *Touches*. De plus, nous devons prendre en compte les différents systèmes de référence utilisés dans les sources hétérogènes. Par exemple, dans la zone atelier, les deux systèmes, WGS84 et Lambert II sont souvent utilisés.

Cette analyse nous amène à considérer l'ontologie owlOGCSpatial [Mefteh *et al.*, 2012] développée par notre équipe de recherche à l'aide de la transformation de modèle comme présenté dans 4.4.1.4.

6.2.3 Conception et développement des ontologies locales

Dans cette section, nous présentons le développement des ontologies locales. Celles-ci s'appuient sur le modèle de base proposé dans 6.2.1 et réutilisent les concepts et relations temporels et spatiaux définis dans OWL-Time et owlOGC-Spatial. D'une part, elles ont pour but de modéliser les sources de données de la zone atelier. D'autre part, elles aident à construire une ontologie finale pour l'intégration de données.

6.2.3.1 Une ontologie pour l'assolement

La Figure 6.9 présente une ontologie pour les relevées de l'assolement. Dans cette ontologie, les parcelles (*gem:Parcel*) sont des objets spatio-temporels qui ont plusieurs tranches de temps (*gem:TimeSlice*) définissant la culture associée (*gem:LandUse*), un emplacement occupé (*os:Polygon* ou *os:MultiPolygon*) et correspondant à un intervalle de temps défini (*time:Interval*).

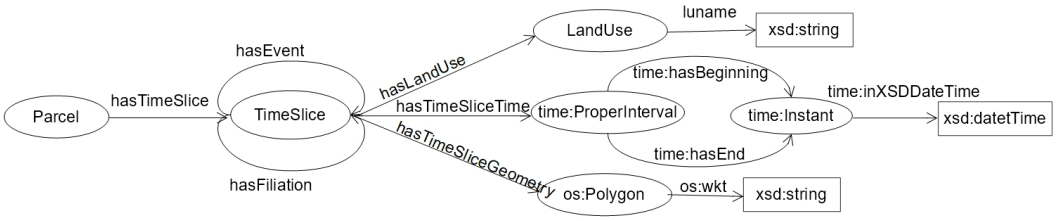


Figure 6.9 – Une ontologie pour l'assolement.

Afin de représenter ces associations spatiales dans le temps, nous utilisons la relation *filiation* qui permet de connecter deux *timeslices* consécutifs d'un même objet. Un changement sur la composante spatiale ou sur la composante sémantique génère un nouveau *timeslice*. D'une part, ce *timeslice* est obligatoirement lié par une relation de filiation avec le *timeslice* d'origine. D'autre part, l'intervalle d'existence du *timeslice* parent est contiguë à l'intervalle du *timeslice* enfant.

Pour améliorer les performances du système et simplifier la représentation, la géométrie des parcelles est calculée grâce aux microparcelles qui leur appartiennent. Ceci signifie que finalement le modèle à composites spatio-temporels n'est plus nécessaire. Dès lors, par rapport à la gestion de l'alimentation de la base de données, cela peut représenter une simplification conséquente du système.

6.2.3.2 Une ontologie pour les observations des oiseaux

La Figure 6.10 présente une ontologie pour les observations des oiseaux. Dans cette ontologie, les oiseaux (*gem:Bird*) sont des objets spatio-temporels. Ils sont observés dans une ou plusieurs observations (*gem:Obsv*) dans lesquelles une description (*gem:ObsvDesc*), l'heure (*time:Instant*) ainsi que le point d'observation (*os:Point*) sont relevés.

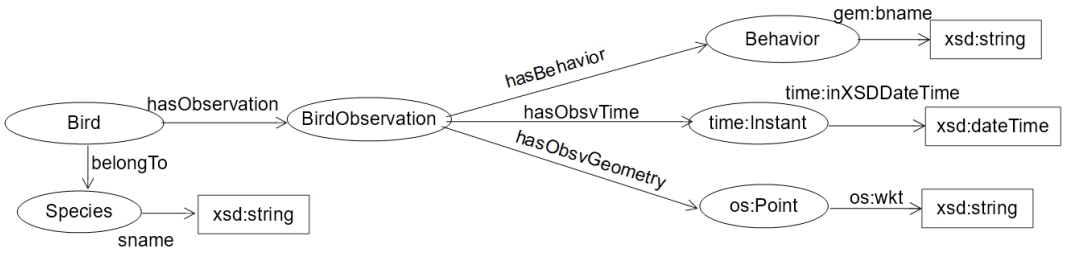


Figure 6.10 – Une ontologie pour les observations des oiseaux.

6.2.3.3 Une ontologie pour les observations des nids

La Figure 6.11 représente une ontologie pour les observations de nids d'oiseau. Dans cette ontologie, les nids (*gem:Nest*) sont des objets spatio-temporels. Ils

sont observés dans une ou plusieurs observations (*gem:NestObsv*) dans lesquelles le contexte (*gem:Context*), la date et l'heure (*time:Instant*) ainsi que le point d'observation (*os:Point*) sont relevés.

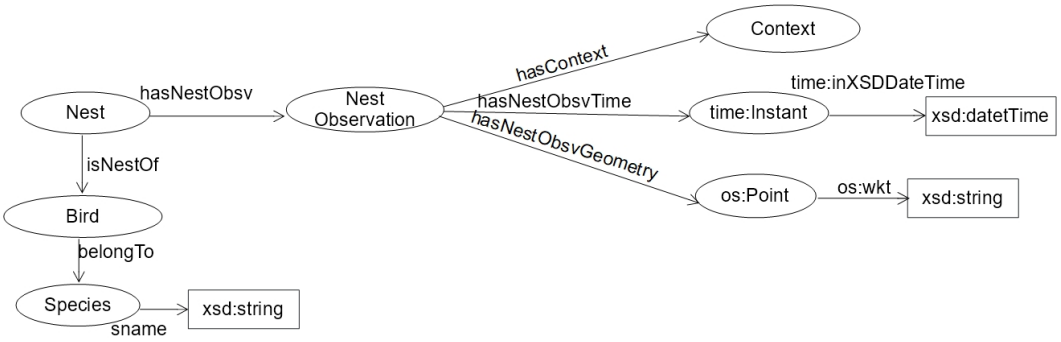


Figure 6.11 – Une ontologie pour les observations des nids.

6.2.3.4 Une ontologie pour les observations des campagnols

La Figure 6.12 représente une ontologie pour les observations des campagnols. Dans cette ontologie, les micromammifères (*gem:Micromammal*) sont des objets spatio-temporels. Ils sont capturés dans un ensemble de pièges (*gem:Trapset*) qui est posé à une date précise et à des coordonnées déterminées (*os:Point*).

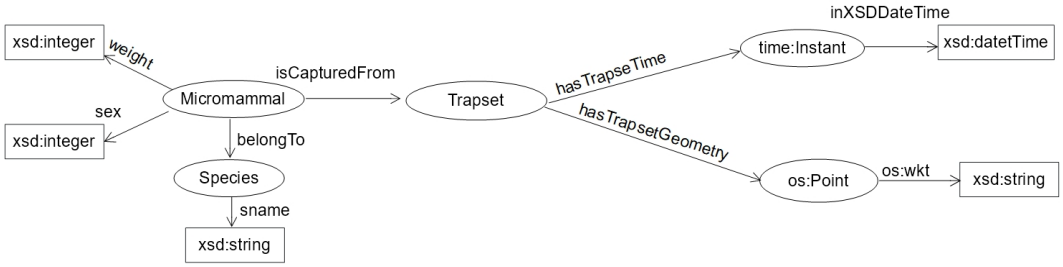


Figure 6.12 – Une ontologie pour les observations des campagnols.

6.2.4 Une ontologie pour l'environnement

À partir des ontologies locales présentées précédemment, une ontologie globale pour l'environnement est construite (Figure 6.16). Les concepts de même sémantique provenant des ontologies locales sont généralisés (ou fusionnés) aux concepts abstraits de cette ontologie. Pour cela, le constructeur *rdfs:subClassOf* reliant une classe spécifique à une classe générale est utilisé. Nous proposons l'approche suivantes :

1. Pour les concepts réutilisés dans chaque ontologie locale, on essaie de les généraliser jusqu'à l'obtention d'un concept existant en commun. Par exemple, *Instant* et *ProperInterval* sont les deux concepts utilisés dans les ontologies locales. Le dernier est généralisé à *Interval* pour enfin obtenir le concept en commun *TemporalEntity*. Par conséquent, ces quatre concepts sont à intégrer dans l'ontologie globale. Dans un context général, une telle généralisation ont été appliquée dans tOWL [Frasincar *et al.*, 2010] pour permettre les différentes représentations de la dimension temporelle.

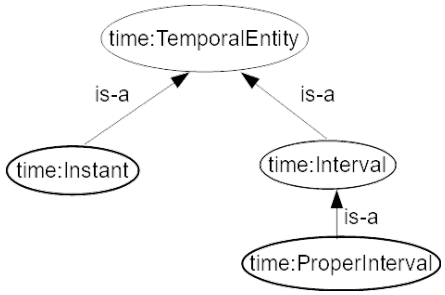


Figure 6.13 – Exemple d'une généralisation des concepts réutilisés.

2. Pour les autres concepts de même sémantique, nommer des concepts abstraits qui les généralisent. Le constructeur *rdfs:subClassOf* est aussi utilisé pour décrire la relation hiérarchique entre ces concepts. Ce processus permet de rassembler les concepts similaires provenant de différentes sources en groupe. Ainsi, ils bénéficient de ses propriétés qui sont aussi en commun entre eux. Par exemple, les concepts *Trapset*, *TimeSlice*, *Observation* et *NestObservation* sont généralisés au concept abstrait *STElement* (élément spatio-temporel).

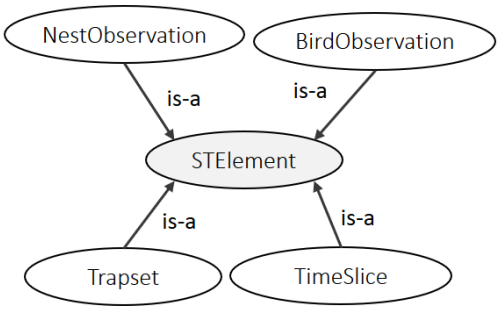


Figure 6.14 – Exemple d'une généralisation des concepts.

3. Nommer une relation qui lie les nouveaux concepts. Ainsi, les relations existantes entre leurs sous-concepts sont des sous-propriétés de cette relation. Par exemple, la relation *hasTime* a été introduite pour relier le concept *STElement* et sa dimension temporelle *TemporalEntity*.

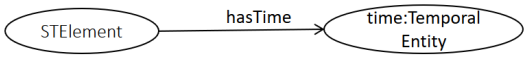


Figure 6.15 – Exemple d'une généralisation de propriété.

En appliquant ces démarches, les concepts suivants sont introduits :

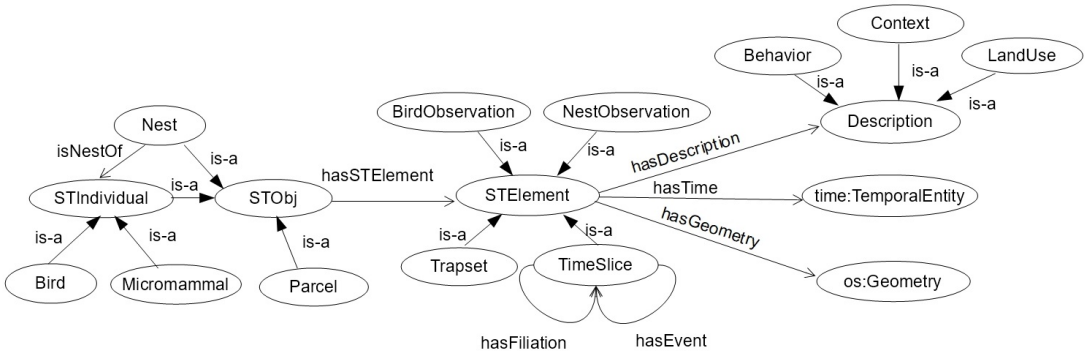


Figure 6.16 – Une ontologie pour l'environnement.

Objet spatio-temporel

Les objets spatio-temporels (*gem:STObj*) dans nos études sont les parcelles, les nids et les individus (appelés ici les individus spatio-temporels, *gem:STIndividual*) appartenant aux différents taxons observés (insectes, oiseaux ou micro-mammifères). Ils peuvent être identifiés avec un numéro de suivi ou rester anonymes.

Élément spatio-temporel

Les objets spatio-temporels ont un ou plusieurs éléments spatio-temporels (*gem:STElement*) qui correspondent à leurs caractéristiques et occupations spatiales à travers leur vie. La classe *gem:STElement* se compose de quatre sous-classes :

- *gem:Observation* pour les observations des oiseaux.
- *gem:NestObservation* pour les observations des nids.
- *gem:Trapset* pour les relevées des pièges des micromammifères.
- *gem:TimeSlice* pour les relevés de l'assolement.

Chaque élément spatio-temporel a trois composantes : spatiale (*os:Geometry*), temporelle (*time:TemporalEntity*) et sémantique (*gem:Description*).

Composante temporelle

Dans cette composante, les deux concepts *ProperInterval* et *Instant* sont généralisés au concept *TemporalEntity* de l'ontologie OWL-Time.

Composante spatiale

Dans cette composante, les concepts *os:Polygon*, *os:Multi-Polygon* et *os:Point* sont généralisés au concept abstrait *os:Geometry* de l'ontologie owlOGC-Spatial. Afin de simplifier la représentation et d'améliorer les performances du système, les données spatiales provenant de différents systèmes de coordonnées sont toutes converties en WGS84.

Composante sémantique

Dans cette composante, la classe *gem:Description* est utilisée pour désigner toute la sémantique liée aux éléments spatio-temporels. Elle généralise la description des observations ou la culture de la parcelle relevée.

L'ensemble des concepts utilisés est décrit par le tableau 6.3. À noter que la hiérarchie de concept se représente par le constructeur *rdfs:subClassOf*.

Concept	**Sur-concept**	**Description**
TopConcept	-	TopConcept
Species	TopConcept	Espèces vivants
STObj*	TopConcept	Objet spatio-temporel
Nest	STObj	Nid d'oiseaux
Parcel	STObj	Parcelle agricole
STIndividual*	STObj	Individu spatio-temporel
STElement*	TopConcept	Élément spatio-temporel
Observation	STElement	Observation des oiseaux
NestObservation	STElement	Observation des nids
TimeSlice	STElement	Relevé de l'assolement
Trapset	STElement	Relevé de piège de micromammifère
LandUse	TopConcept	Culture
LandUseType	TopConcept	Type de culture
TemporalEntity*	TopConcept	Entité temporelle
Interval*	TemporalEntity	Intervalle
ProperInterval	Interval	Intervalle de temps
Instant	TemporalEntity	Instant de temps
Geometry*	TopConcept	Géometrie
Surface*	Geometry	Surface
Polygon	Surface	Polygone
Point	Geometry	Point
GeometryCollection*	Geometry	Collection de géométrie
Multi-Surface*	GeometryCollection	Multi-Surface
Multi-Polygon	Multi-Surface	Multi-Polygone

Tableau 6.3 – Les concepts utilisés dans l'ontologie pour l'environnement (*: classe abstraite).

6.2.5 Études de cas

À partir de l'ontologie proposée, quelques études de cas sont examinées afin de montrer la capacité de l'ontologie dans la représentation des phénomènes réels ainsi que des besoins d'analyse des experts. En effet, dans ces exemples, des relations spatio-temporelles entre les objets sont exploitées.

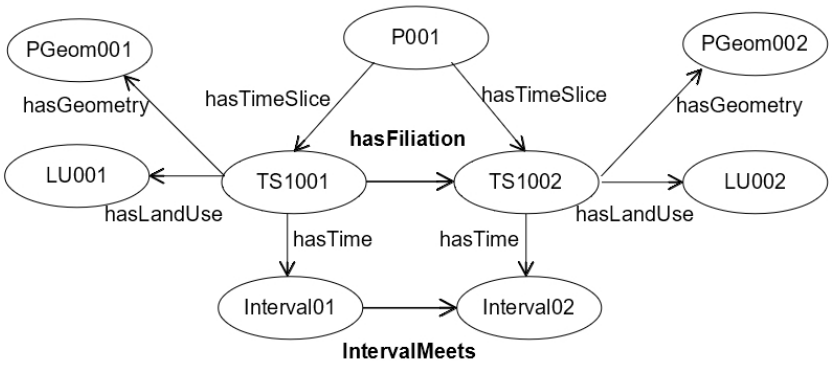


Figure 6.17 – Représentation d'une rotation de culture.

§1 Représentation d'une rotation de culture

La rotation des cultures se produisant dans une parcelle est représentée par leurs *timeslices* adjacents. Dans ce cas, la relation *intervalMeets* entre l'intervalle de temps de chaque *timeslice* est utilisé (Figure 6.17). De manière plus rapide, nous pouvons profiter de la relation *hasFiliation* entre deux *timeslices*.

§2 Représentation d'un événement territorial

Un événement territorial peut être représenté par les relations *hasFiliation* et *hasEvent* entre des *timeslices* de différentes parcelles qui sont déduites par des relations spatio-temporelles entre ces derniers. Par exemple, la Figure 6.18 décrit un événement d'intégration de deux parcelles, dans lequel, une parcelle (P001) est absorbée par une autre (P002). Après l'absorption, comme la parcelle P002 continue à exister, elle garde donc son identité et donne la naissance à une autre *timeslice*, alors que l'autre (P001) disparait.

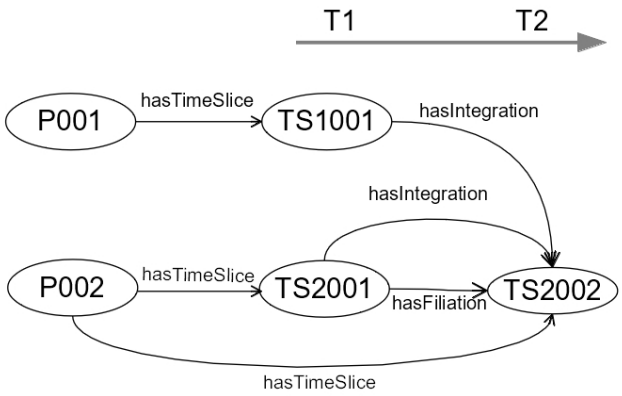


Figure 6.18 – Exemple d'un événement d'intégration de deux parcelles.

§3 Croisement des observations et des relevées d'assolement

Pour croiser les données des observations des espèces vivantes et des relevées d'assolement, la relation spatiale entre les points d'observation et la géométrie des parcelles ; et la relation temporelle entre les relevées sont utilisées. La Figure 6.19 représente une telle liaison dans laquelle, la relation *within* entre le point d'observation P1001 et le polygone d'une parcelle P001 et la relation *inside* entre l'intervalle Interval001 et l'instant Instant001 sont utilisées.

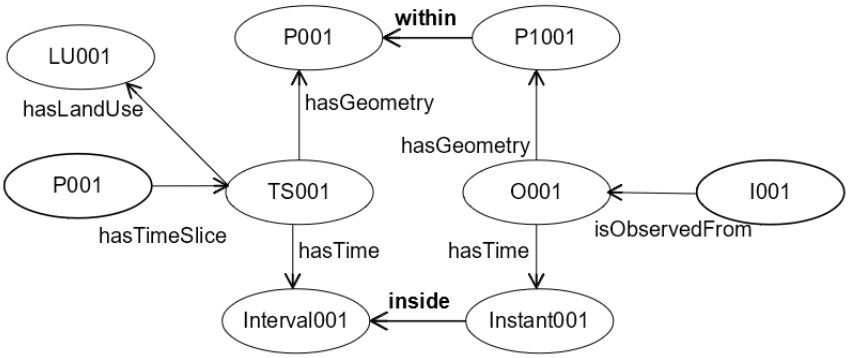


Figure 6.19 – Exemple d'un croisement des observations et des relevées de l'assolement.

§4 Croisement de différentes observations

De la même manière,les relations spatio-temporelle entre les instances de l'*Observation* sont utilisées afin de relier les différentes observations. Par exemple, la Figure 6.20 décrit comment lier l'apparition d'un nid d'oiseaux (N001) situé aux coordonnées P001 qui est aux alentour d'un point d'observation d'un campagnol (I101) de l'année précédente (year101). La relation spatiale se compose d'une

fonction spatiale *buffer* et de la relation *within*, alors que la relation temporelle est déduit par la comparaison des années extraites des instants d'observations.

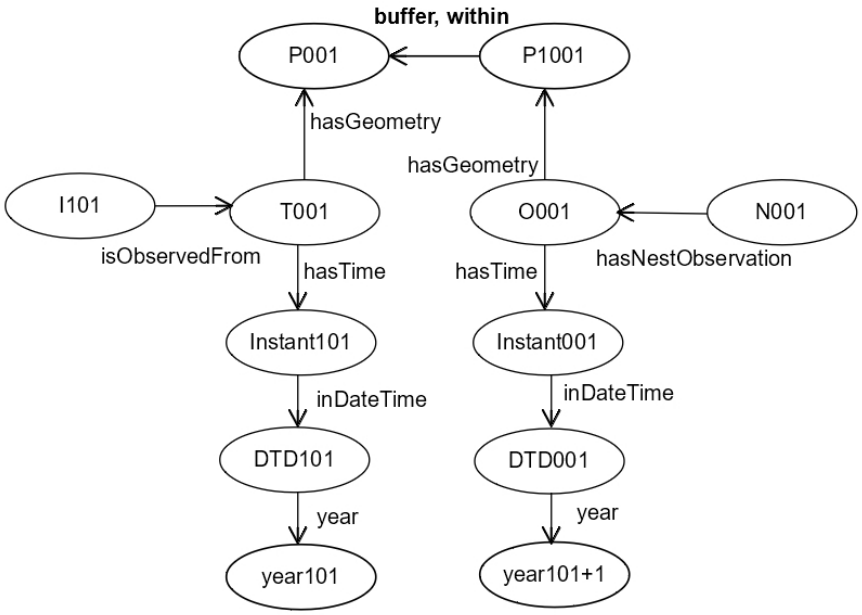


Figure 6.20 – Exemple d'un croisement de différentes observations.

Conclusion du chapitre

La zone atelier Plaine et Val de Sèvre et ses bases de données constituent notre cas d'étude. Il s'agit de bases de données environnementales de caractères spatio-temporels collectés par différentes équipes de chercheurs, ce qui produit des problèmes d'hétérogénéités. Néanmoins, il existe de forts besoins d'analyse et d'exploitation de l'ensemble de ces sources de données. Dès lors, il est nécessaire d'introduire une approche permettant le croisement de ces données en prenant en compte leurs relations temporelles et spatiales ou autrement dit, avec un raisonnement spatio-temporel.

Nous avons montré comment construire une ontologie spatio-temporelle utilisée comme un schéma global pour l'intégration des sources de données en réutilisant une ontologie du temps et une ontologie de l'espace. Afin de prendre en compte les changements ou l'évolution des objets spatio-temporels, l'approche perdurantiste a été appliquée. Dans cette modélisation, chaque objet spatio-temporel possède plusieurs tranches de temps correspondant aux différentes observations ou relevées. Celles-ci se composent de trois composantes : temporelle, spatiale et sémantique.

Bien que l'ontologie présentée ne soit appliquée qu'aux données environnementales de la zone atelier de Plaine et Val de Sèvre, elle peut être adaptée ou reconstruite de la même manière dans un nouveau contexte.

Chapitre 7

Analyse de données environnementales hétérogènes

Sommaire

7.1 Intégration de données 141
7.2 Intégration de données par un médiateur 142
7.3 Intégration de données par l'entrepôt 144
7.3.1 Architecture du système 145
7.3.2 Construction de la base de connaissances 147
7.3.3 Applications 148
7.4 Fouille de données spatio-temporelles sémantiques . 152
7.4.1 Architecture du système 153
7.4.2 Applications 156

Introduction du chapitre

Dans le chapitre précédent, une ontologie spatio-temporelle a été conçue pour l'intégration sémantique des données hétérogènes de la zone atelier. Afin de concrétiser l'approche proposée, des systèmes ont été mis en place pour répondre aux besoins d'analyse de la zone atelier. Le premier suit l'approche médiateur, alors que le deuxième applique l'approche entrepôt avec une extension appliquant les méthodes de fouille de données. Bien que chaque approche possède des avantages et des inconvénients, leur but final est de fournir un système d'intégration de données hétérogènes avec la prise en compte de leurs relations spatio-temporelles (Figure 7.1). Pour cela, ils s'appuient sur l'analyse sémantique qui peut être réalisée par des requêtes sémantiques ou par des méthodes de fouille de données.

Ce chapitre est consacré à la présentation de ces systèmes avec des exemples de cas d'utilisation.

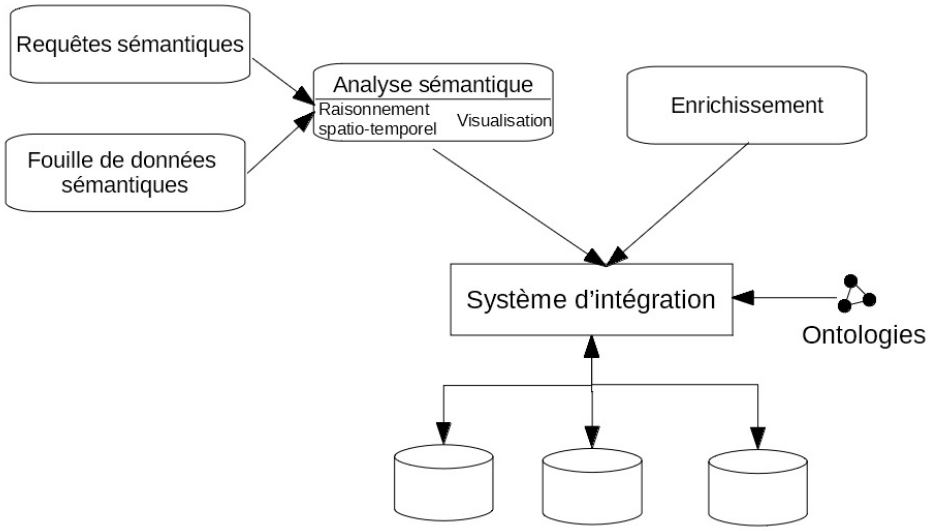


Figure 7.1 – Vue d'ensemble des fonctions du système.

7.1 Intégration de données

Dans le chapitre précédent, une ontologie a été proposée en tant que schéma global en vue de modéliser les sources de données et les intégrer. Pourtant, comme seuls les concepts et leurs relations sont représentés pour former la TBox, il est nécessaire de construire la ABox, autrement dit, c'est de peupler l'ontologie avec les données sources. Dans le peuplement de l'ontologie, lui-même, il existe des techniques pour exposer des données relationnelles en RDF : la matérialisation de données et la traduction de données à la demande. Quelle que soit la technique choisie, un outil de traduction est nécessaire pour l'exposition de ces données. Dans notre travail, l'outil D2RQ a été utilisé par sa simplicité, sa fonctionnalité et le support de nombreux SGBD. Il est à souligner que l'outil Ontop a récemment été développé et amélioré au niveau du fonctionnement et des performances. Il pourra être considéré dans nos futurs travaux.

En fonction de l'approche d'intégration appliquée, le serveur D2R ou l'engine D2RQ seul est utilisé. Dès lors, deux architectures comme ci-dessous sont proposées.

§1 Traduction de données à la demande

L'architecture (Figure 7.2) est appliquée dans les approches médiateurs. Dans cette architecture, un ou plusieurs serveurs D2R sont utilisés pour exposer des sources de données en RDF. Chaque serveur forme ainsi un endpoint à partir duquel les applications peuvent envoyer des requêtes SPARQL 1.1 à l'aide de l'API Jena [1].

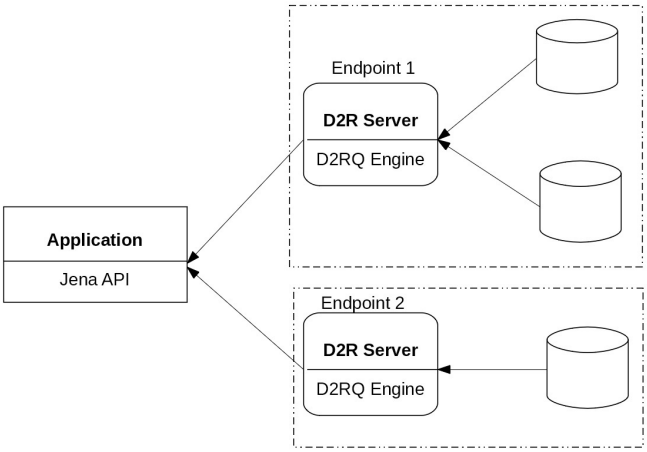


Figure 7.2 – Traduction de données à la demande avec D2R Server.

1. http://jena.apache.org/

§2 Matérialisation de données

L'architecture (Figure 7.3) est appliquée pour les approches entrepôt. Dans cette architecture, les étapes ETL se réalisent grâce à l'outil D2RQ qui convertit les données en triplets RDF. Ces derniers, stockés dans des fichiers RDF, sont ensuite chargés dans un triplestore.

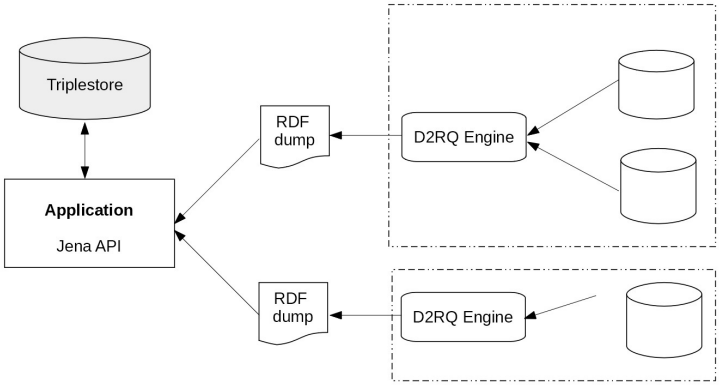


Figure 7.3 – Matérialisation de données avec D2RQ.

7.2 Intégration de données par un médiateur

Dans une première phase de notre travail, nous avons présenté un système d'intégration de données spatio-temporelles hétérogènes par l'approche médiateur [Tran *et al.*, 2014]. C'était l'architecture la plus simple à mettre en œuvre, incluant une application développée pour l'exploitation de données intégrées (Figure 7.4). Le système est composé de quatre parties : la traduction de données, le raisonnement sur des relations temporelles, le traitement des requêtes sémantiques et la visualisation des données. Une application web a été développée permettant de recevoir des requêtes SPARQL et de retourner le résultat en JSON qui est visualisé sur une carte.

Puisqu'il est basé sur l'approche médiateur, le système posait des problèmes de performances et de fonctionnement déjà présentés dans l'état de l'art. D'une part, les performances du système n'étaient pas satisfaisantes. En effet, selon une estimation, plus d'un million de triplets RDF pouvaient être exposés grâce à l'outil D2RQ. Pour une telle quantité de triplets, une recherche des parcelles suivant leur culture et une année donnée prenait de 2 à 10 secondes. Afin d'améliorer les performances du système, les relations temporelles entre les objets étaient pré-calculées puis unies avec les données RDF obtenues par la traduction à la volée lors d'une requête SPARQL. Cela rapproche le système d'une approche hybride. Malgré ces efforts, les requêtes sur les rotations de cultures nécessitant un raisonnement temporel sont résolues en 25 minutes. D'autre part, le système ne

permettait pas encore de raisonner sur les relations spatiales. Ces deux défauts majeurs du système peuvent s'expliquer par les raisons suivantes :

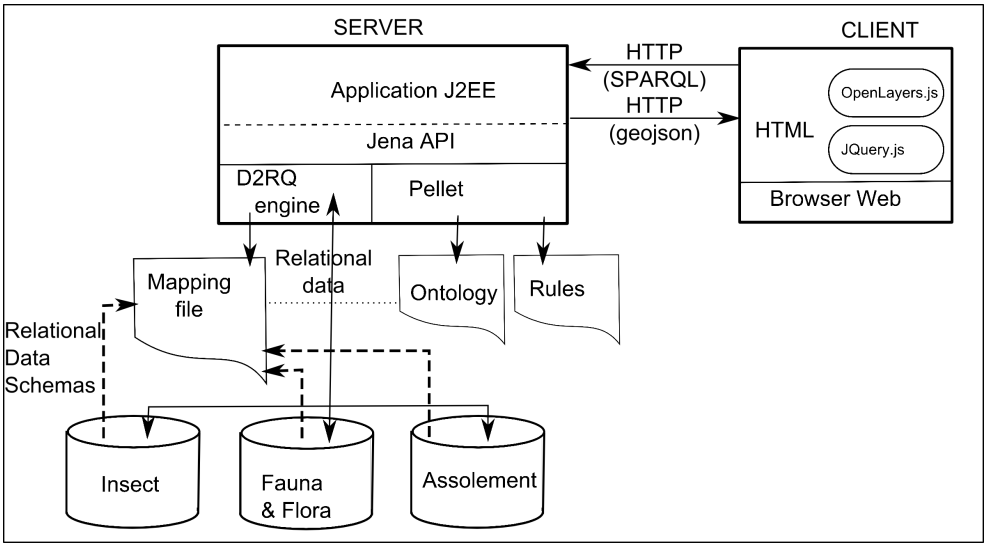


Figure 7.4 – Système d'intégration suivant l'approche médiation [Tran *et al.*, 2014].

— Il n'est pas encore possible d'utiliser les outils traduisant les bases de données relationnelles existantes en graphe RDF à la volée, car l'accès aux données serait alors bien plus lent qu'en utilisant soit des triplestores RDF, soit le modèle relationnel sous-jacent [Gray *et al.*, 2009].

— De plus, malgré des efforts pour permettre la traduction des données géospatiales, les outils de traduction actuels ne sont pas capables de réaliser une requête spatiale fédérée [2]. Par exemple, l'outil Ontop-spatial est très récent, il inclut les filtres spatiaux dans les requêtes, mais les requêtes fédérées qui effectuent des jointures spatiales provenant de différents bases de données géospatiales ne sont pas prises en charge par le système [Brüggemann *et al.*, 2016].

— Comme souligné, l'utilisation des approches telles que l'ajout du raisonnement spatial dans le Web sémantique et la traduction des relations spatiales en axiomes de classes OWL, permettant le raisonnement spatial dans le Web sémantique pose aussi des problèmes de performances.

Par conséquent, nous avons choisi ensuite de matérialiser les bases de données hétérogènes dans un triplestore géospatial. De cette façon, le raisonnement spatial sera pris en charge par le triplestore. Le système open-source Strabon [Kyzirakos *et al.*, 2012] a été utilisé pour la matérialisation en raison de bonnes performances globales et du support complet des fonctions géospatiales.

2. https://www.w3.org/TR/sparql11-federated-query/

7.3 Intégration de données par l'entrepôt

L'architecture du système est basée sur l'approche entrepôt où les techniques ETL sont appliqués pour matérialiser les données relationnelles hétérogènes dans un triplestore géospatial. Comme le triplestore Strabon a été choisi, la dimension spatiale de l'ontologie développée doit être liée à son modèle stRDF. Pour cela, la propriété *os:wkt(os:Geometry, xsd:string)* est remplacée par *os:geometry (os:Geometry, strdf:geometry)* (Figure 7.5).

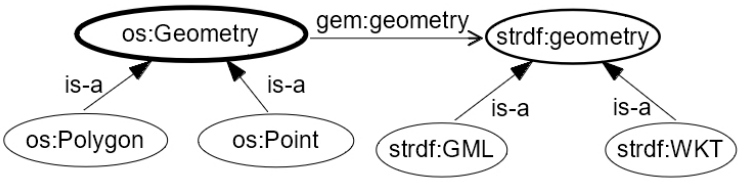


Figure 7.5 – Réutilisation du modèle stRDF pour la représentation des géométries.

Ainsi, les relations spatiales entre les objets sont déduites par les fonctions spatiales du langage stSPARQL. La Figure 7.1 énumère huit fonctions spatiales que supporte le système ainsi que les relations correspondantes dans le modèle *RCC8*.

RCC8 relation	OGC relation	Schema	Strabon function
disconnected [DC]	disjoint	DC(a,b)	strdf:disjoint(a,b)
externally connected [EC]	touches	EC(a,b)	strdf:touches(a,b)
partially overlapping [PO]	overlaps	PO (a,b)	strdf:overlaps(a,b)
equal [EQ]	equal	EQ(a,b)	strdf:equals(a,b)
tangential proper part [TPP]	within	TPP(a,b)	strdf:within(a,b)
non-tangential proper part [nTPP]	within	nTPP(a,b)	strdf:within(a,b)
tangential proper part inverse [TPPi]	contains	TPPi(a,b)	strdf:contains(a,b)
non-tangential proper part inverse [nTPPi]	contains	nTPPi(a,b)	strdf:contains(a,b)

Tableau 7.1 – Les fonctions spatiales du langage stSPARQL.

7.3.1 Architecture du système

Le nouveau système (Figure 7.6) se compose d'une base de connaissance, appelé Geminat. Celle-ci, gérée par Strabon, héberge un endpoint SPARQL [3] pour accéder à ses données. En plus, trois couches de fonctionnement peuvent être distinguées : la couche de base, la couche sémantique et la couche application.

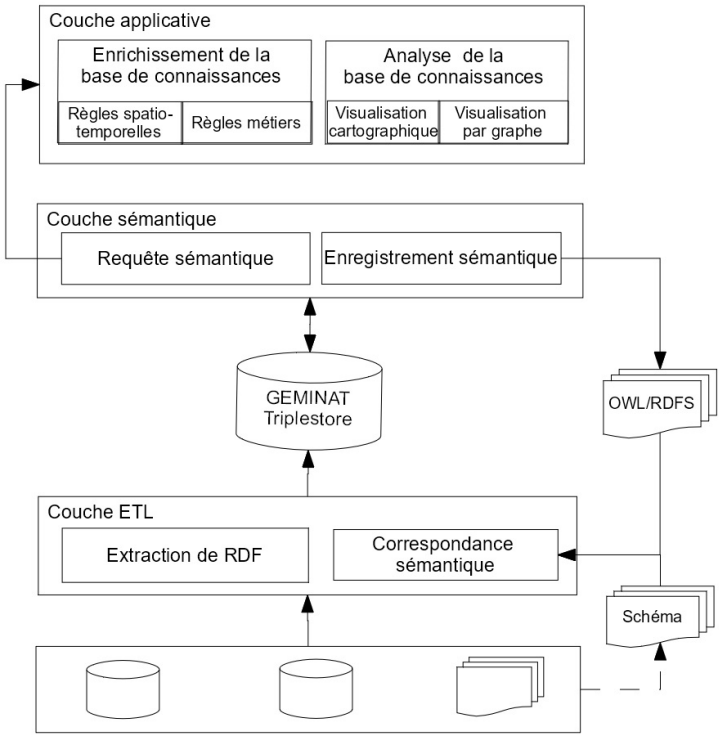


Figure 7.6 – Système d'intégration sémantique de données suivant l'approche entrepôt.

Couche de base

Étant basé sur l'approche entrepôt, le peuplement de l'ontologie est réalisé par un processus d'ETL. Les deux premières étapes du processus s'effectuent à la couche de base. L'informaticien définit des correspondances entre le schéma des bases de données et les ontologies. C'est grâce à ces correspondances que les données pertinentes sont extraites et transformées en RDF par le module *Extraction de RDF*. À noter que l'application d'un tel processus sur les bases non-relationnelles implique a priori un chargement de leurs données dans une base relationnelle intermédiaire pour des raisons soulignées dans 3.2.1.2. Après avoir été générés, les triplets RDF peuvent être ensuite importés dans la base de connaissances.

3. https://www.w3.org/TR/void/#sparql

— Dans le premier module, des requêtes SQL sont utilisées pour sélectionner les données désirées. Des vues peuvent être éventuellement construites afin de mettre en correspondance ces données à l'ontologie. Cela permet la transformation de données réalisée à l'étape suivante car il existe des différences entre la représentation et la modélisation de données en RDF et celle d'une base de données relationnelle [Martinez-Cruz *et al.*, 2012], surtout dans l'aspect spatio-temporel.

— Le deuxième module aide à exposer les données en RDF, celles-ci sont exportées dans des fichiers RDF. Pour cela, il gère les fichiers de *mapping* définissant comment se connecter à des bases de données et la correspondance entre les ontologies et le schéma des bases de données.

Couche sémantique

La couche sert à gérer la base de connaissances et à traiter les requêtes sémantiques provenant de la couche supérieure. Les modules développés profitent des fonctions existantes de l'endpoint SPARQL.

— Le premier module concerne la gestion des ontologies (TBox) et leur peuplement (ABox). Il sert à importer les instances stockées dans des fichiers RDF dans la base de connaissances ainsi qu'à enregistrer des ontologies.

— Le module de requête sémantique permet le traitement des requêtes sémantiques (en stSPARQL) qui sont adressées au triplestore. L'outil essentiel utilisé dans cette couche est le framework Jena.

Couche applicative

La couche propose trois fonctions à l'utilisateur final : l'enrichissement de la base de connaissances, l'analyse de données et la visualisation de données. Les fonctions sont développées comme des services web et réfèrent au module de requête sémantique de la couche inférieure pour communiquer avec la base de connaissances.

— Dans la première fonction, de nouvelles relations entre les entités peuvent être déduites grâce à l'expression de règles spatio-temporelles et de règles métiers. Ainsi, de nouvelles déclarations sont ajoutées à l'aide de requêtes SPARQL Update.

— La deuxième fonction sert à recevoir une demande d'analyse de l'utilisateur puis à la traiter et présenter le résultat retourné par le module de requête sémantique.

— Quant à la visualisation, la dimension spatiale d'un objet spatio-temporel issu de la base de connaissances ou du résultat de l'analyse de données peut être visualisée sur une carte. Pour cela, la bibliothèque

Concept	Nombre d'individus
Species	605
STIndividual	1 086
Parcel	16 670
Observation	51 935
NestObservation	21 460
TimeSlice	141 484
Trapset	4 890
LandUse	53

Tableau 7.2 – Quelques statistiques de la base de connaissances.

OpenLayers[4] avec des fonds de carte d'*OpenStreetMap*[5] sont utilisés. En même temps, le résultat d'une analyse statistique peut être visualisé par des diagrammes pour donner une vue globale à l'utilisateur. C'est aussi grâce à cette dernière qu'il peut vérifier ses connaissances métiers ou en découvrir de nouvelles.

7.3.2 Construction de la base de connaissances

L'intégration de données est réalisée grâce au processus d'ETL : l'extraction et la transformation de données relationnelles et le peuplement des ontologies du domaine. À l'aide des fichiers de mapping, les données relationnelles sélectionnées sont exportées en fichiers RDF avant d'être importées dans la base de connaissances pour composer son ABox. Parallèlement, la TBox peut être mise à jour par l'importation des ontologies en RDFS ou OWL. L'annexe A décrit en détail comment construire la base de connaissances en y ajoutant des modèles développés et individus obtenus par la traduction de données.

Une telle méthode a été appliquée pour intégrer des données de la zone atelier à partir des données disponibles (version 2013), notamment celles issues de la base d'assolement, des données d'observations d'oiseaux et des micromammifères. En sélectionnant les données pertinentes, plus de 930 000 triplets RDF ont été importés dans la base. Le tableau 7.2 ci-dessous présente plus de détails de la base. À noter que ce sont des données RDF d'origine importées de sources hétérogènes sans compter les relations spatio-temporelles.

4. http://openlayers.org/
5. http: //www.openstreetmap.org

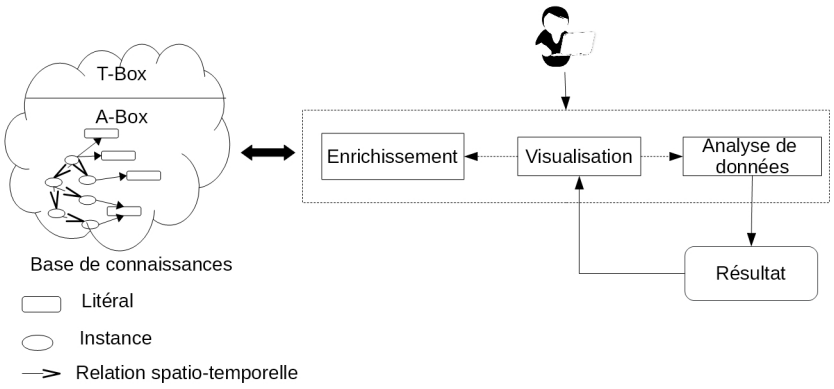


Figure 7.7 – Les fonctions accessibles de l'application et leurs relations à l'égard d'un expert.

7.3.3 Applications

Les utilisateurs finaux peuvent utiliser de manière directe les deux fonctions : enrichissement de la base de connaissances et analyse de données, alors que la visualisation ne peut être accédée qu'après l'obtention de résultats d'analyse. Grâce aux résultats obtenus et à la visualisation, ils peuvent ensuite modifier leurs analyses et/ou mettre à jour la base avec leurs connaissances pour améliorer ces résultats. La Figure 7.7 décrit les fonctions accessibles et leurs relations à l'égard d'un utilisateur final.

§1 Enrichissement de la base de connaissances

Comme décrit dans 3.1.3.2, les requêtes SPARQL servent non seulement à insérer de nouvelles déclarations au sein de la base de connaissances mais aussi à représenter les règles. Pour ces raisons, elles sont utilisées pour définir les règles spatio-temporelles, les règles métiers ou encore les connaissances d'expert pour enrichir la base de connaissances.

Enrichissement par les règles spatio-temporelles

Une règle temporelle peut se représenter par une requête SPARQL Update. De cette manière, nous pouvons déduire l'ensemble des 14 relations temporelles appliquées dans notre étude entre les deux concepts temporels, l'instant et l'intervalle (voir l'Annexe C). Dès lors, les règles spatio-temporelles peuvent être construites à partir d'une combinaison entre ces relations temporelles et les relations spatiales déduites par le triplestore Strabon. Par exemple, le code 7.1 décrit comment découvrir des événements d'intégration des parcelles par leurs relations temporelles (ligne 15), leurs relations spatiales (ligne 19) et une fonction métrique en plus (ligne 19). Cette analyse montre l'avantage et la nécessité d'utiliser le langage stPARQL permettant des fonctions d'analyse géométrique, y compris les fonctions d'agrégation

spatiale, les fonctions de construction et les fonctions métriques, comparé au GeoSPARQL.

Code 7.1 – Detection des événements d'intégration entre les parcelles

```
1  PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5  PREFIX  time: <http://www.w3.org/2006/time#>
6  PREFIX  gem: <http://za-geminat.cnrs.fr/Assolement.owl#>
7  INSERT
8  {
9    ?tsa gem:hasIntegration ?tsb.
10 }
11 WHERE
12 {
13  ?tsa rdf:type gem:TimeSlice.
14  ?tsa gem:hasTime ?intva.
15  ?intva time:intervalMeets ?intvb.
16  ?tsb gem:hasTime ?intvb.
17  ?tsa gem:pgeometry ?geoma.
18  ?tsb gem:pgeometry ?geomb.
19 FILTER(strdf:intersects(?geoma,?geomb) && strdf:area(?geomb)
      >strdf:area(?geoma))
20 }
```

En utilisant cette approche, nous avons de nouvelles connaissances comme suivantes :

Relation		**Nombre**
Événement territorial	gem:hasFusion	536
	gem:hasIntegration	2035
	gem:hasScission	300
	gem:hasExtraction	1464
Relation temporelle	time:insides	25275
	gem:hasFiliation	124290

Tableau 7.3 – Une proposition de regroupement des cultures de la zone atelier[Lazrak *et al.*, 2010].

Règles métiers

Les règles métiers conçues par les experts peuvent être également décrites par des requêtes SPARQL basées sur les relations enrichies. Le code 7.2 est un exemple d'une représentation de la succession de cultures peu probable

«Tournesol-Colza» par une requête SPARQL en utilisant la relation *hasFiliation* entre deux *timeslices* (ligne 13), celle-ci est déduite à l'aide de leur relation temporelle. Ensuite, à partir du résultat obtenu, il est possible de mettre à jour la base de connaissances par des valeurs corrigées.

Code 7.2 – Detection des succession peu probable.

```
1  PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5  PREFIX time: <http://www.w3.org/2006/time#>
6  PREFIX gem: <http://za-geminat.cnrs.fr/Assolement.owl#>
7  SELECT *
8  WHERE
9  {
10  ?ts1 rdf:type gem:TimeSlice.
11  ?ts1 gem:hasLandUse ?c1.
12  ?c1 gem:cname ?name1.
13  ?ts1 gem:hasFiliation ?ts2.
14  ?ts2 gem:hasLandUse ?c2.
15  ?c2 gem:cname ?name2.
16 FILTER (?name1="Tournesol" && ?name2="Colza")
17 }
```

Enrichissent par des connaissances d'expert

Grâce aux connaissances des experts du domaine, il est possible de mettre à jour la base de connaissances en corrigeant des déclarations erronées ou en y ajoutant de nouvelles. Par exemple, une proposition de regroupement de culture se réalise par le code 7.3 dans lequel, le blé (ligne 5), le blé barbu (ligne 6) et la céréale (ligne 7) sont regroupés en un seul type de culture «Blé» (ligne 3-4).

Code 7.3 – Ajout de nouvelles connaisances à la base.

```
1  INSERT DATA
2  {
3  <http://za-geminat.cnrs.fr/Assolement.owl#LUType/6> <http://
       www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://za-
       geminat.cnrs.fr/Assolement.owl#LUType> .
4  <http://za-geminat.cnrs.fr/Assolement.owl#LUType/6> <http://
       za-geminat.cnrs.fr/Assolement.owl#tname> "Blé" .
5  <http://za-geminat.cnrs.fr/Assolement.owl#LandUse/188> <http
       ://za-geminat.cnrs.fr/Assolement.owl#belongTo> <http://za
       -geminat.cnrs.fr/Assolement.owl#LUType/6>.
6  <http://za-geminat.cnrs.fr/Assolement.owl#LandUse/202> <http
       ://za-geminat.cnrs.fr/Assolement.owl#belongTo> <http://za
```

```
        -geminat.cnrs.fr/Assolement.owl#LUType/6>.
7   <http://za-geminat.cnrs.fr/Assolement.owl#LandUse/180> <http
        ://za-geminat.cnrs.fr/Assolement.owl#belongTo> <http://za
        -geminat.cnrs.fr/Assolement.owl#LUType/6>.
8   }
```

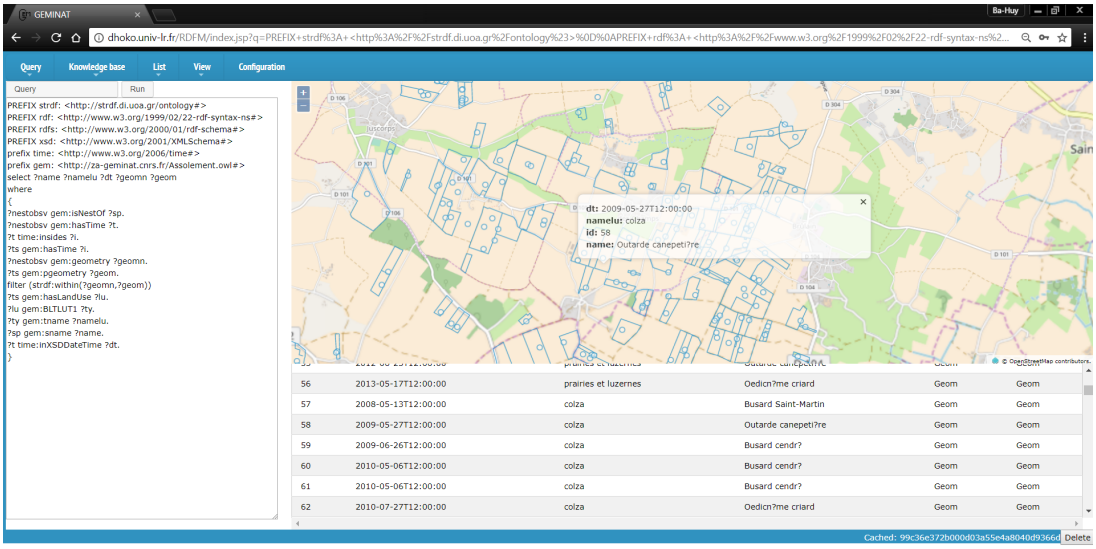


Figure 7.8 – L'interface principale pour l'analyse de données et la visualisation.

§2 Analyse de données et visualisation

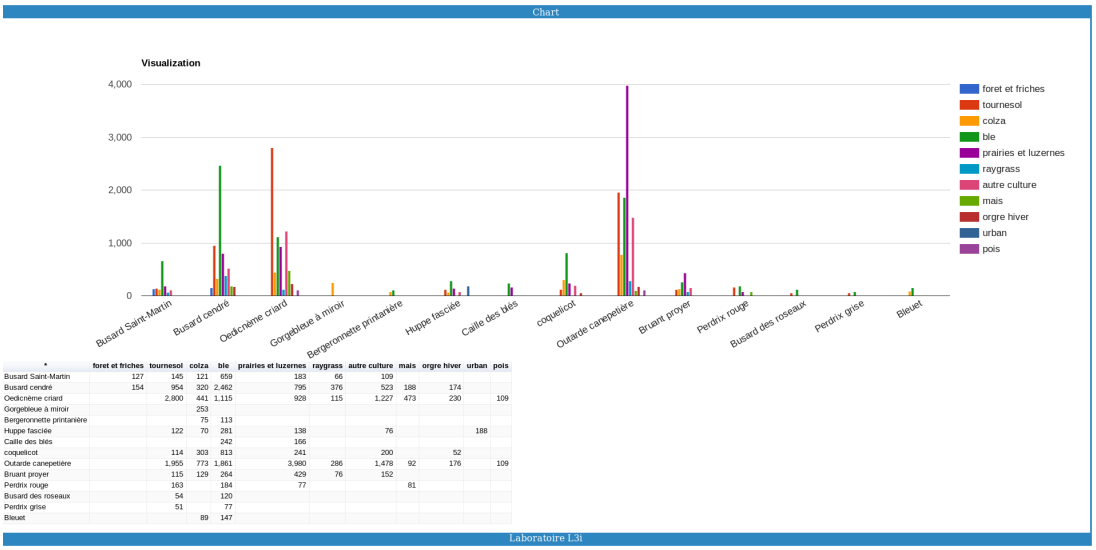
La Figure 7.8 représente l'interface principale dans laquelle l'utilisateur réalise des analyses sous forme de requêtes SPARQL (1), et ensuite reçoit le résultat (2) qui peut être visualisé sur une carte si celui-ci contient des valeurs géométriques (3) ou par un graphique si il concerne des données statistiques.

Le code 7.4 présente une analyse simple de croisement entre deux sources de données décrit dans §3 : les observations d'oiseaux et les relevées d'assolement. Les données sont croisées par leur relation temporelle (ligne 12) et leur relation spatiale (ligne 16). Le résultat obtenu est visualisé sur une carte (Figure 7.8) ou par un graphique (Figure 7.9) si des opérations d'agrégation ont été appliquées.

Code 7.4 – Croisement entre les observations et l'assolement

```
1  PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5  PREFIX time: <http://www.w3.org/2006/time#>
6  PREFIX gem: <http://za-geminat.cnrs.fr/Assolement.owl#>
7  SELECT ?name ?name2 ?geomn ?geom
```

```
 8 WHERE
 9 {
10  ?nestobsv gem:isNestOf ?sp.
11  ?nestobsv gem:hasTime ?t.
12  ?t time:inside ?i.
13  ?ts gem:hasTime ?i.
14  ?nestobsv gem:geometry ?geomn.
15  ?ts gem:pgeometry ?geom.
16 FILTER (strdf:within(?geomn,?geom))
17  ?ts gem:hasLandUse ?lu.
18  ?lu gem:BLTLUT ?ty.
19  ?ty gem:tname ?name2.
20  ?sp gem:sname ?name.
21 }
```



*	foret et friches	tournesol	colza	ble	prairies et luzernes	raygrass	autre culture	mais	orgre hiver	urban	pois
Busard Saint-Martin	127	145	121	659	183	66	109				
Busard cendré	154	954	320	2,462	795	376	523	188	174		
Oedicnème criard		2,800	441	1,115	928	115	1,227	473	230		109
Gorgebleue à miroir			253								
Bergeronnette printanière			75	113							
Huppe fasciée		122	70	281	138		76			188	
Caille des blés				242	166						
coquelicot		114	303	813	241		200		52		
Outarde canepetière		1,955	773	1,861	3,980	286	1,478	92	176		109
Bruant proyer		115	129	264	429	76	152				
Perdrix rouge		163		184	77			81			
Busard des roseaux		54		120							
Perdrix grise		51		77							
Bleuet			89	147							

Figure 7.9 – Visualisation du résultat d'une analyse par un graphe.

7.4 Fouille de données spatio-temporelles sémantiques

Les méthodes de fouille de données ont été appliquées afin de fournir aux utilisateurs un outil d'analyse automatique et efficace. Dans ce contexte, on parle de la fouille de données spatio-temporelles sémantiques qui est considérée comme une étape dans l'extraction de connaissances à partir de données sémantiques. Pour cela, nous proposons une approche pouvant exploiter davantage la puissance des technologies du Web sémantique. Dans cette approche, représentée par la Figure 7.10, l'enrichissement de données est ajouté comme une étape optionnelle

dans l'extraction de connaissances. Hors la fouille de données, les technologies du Web sémantique participent à toutes les autres étapes du processus d'extraction. Il forme ainsi une boucle allant de l'étape d'enrichissement jusqu'à l'évaluation et à l'interprétation du résultat.

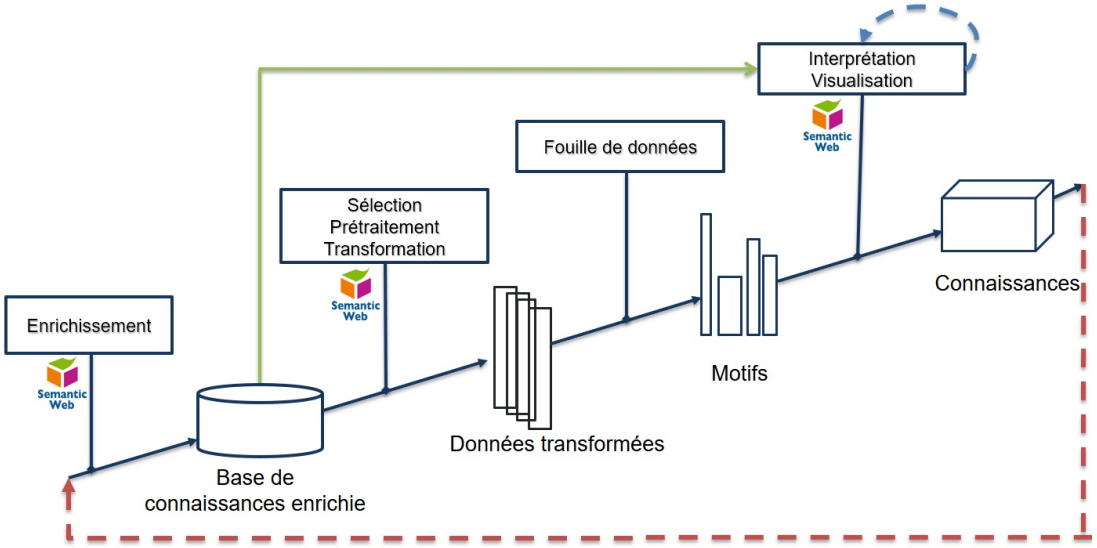


Figure 7.10 – Un pipeline de fouille de données sémantiques.

— L'enrichissement peut être utilisé pour prétraiter les données. En effet, les bruits ou les données aberrantes de la base de connaissances peuvent être éliminés ou corrigés par des connaissances d'experts. Ils peuvent être aussi remplacés par des valeurs provenant du Web de données.

— Les connaissances extraites peuvent servir à améliorer la base de connaissances. À partir de ces informations, on peut vérifier des anomalies ou enrichir de nouveau sa base.

De cette manière, la relation entre le Web sémantique et l'extraction de connaissances est renforcée pour un meilleur résultat. En effet, l'enrichissement de la base peut améliorer le résultat de l'extraction, à l'inverse, l'extraction de connaissances permet une compréhension plus profonde de la base.

7.4.1 Architecture du système

Les techniques de fouille de données sont introduites dans le troisième développement du système. Pour cela, le module de fouille de données sémantiques a été implémenté à la couche application et interagit avec les autres modules comme décrit dans la Figure 7.11.

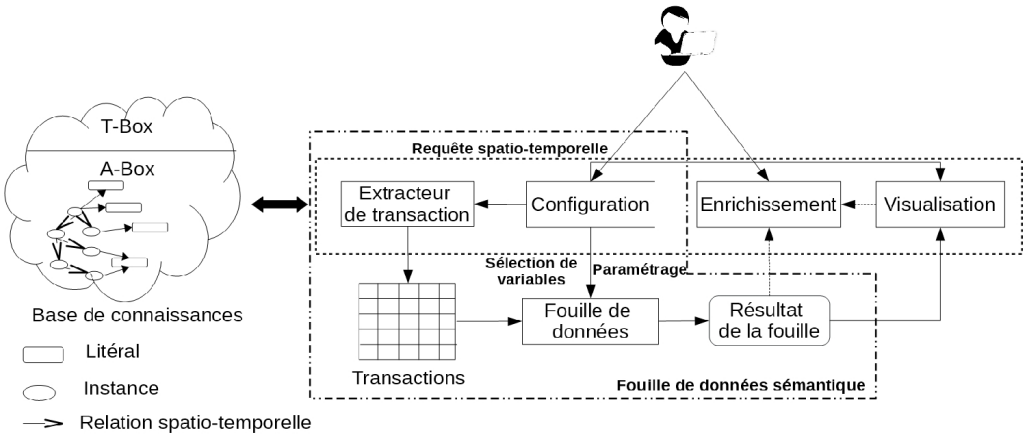


Figure 7.11 – Application de techniques de fouilles dans l'analyse de données spatio-temporelles sémantiques.

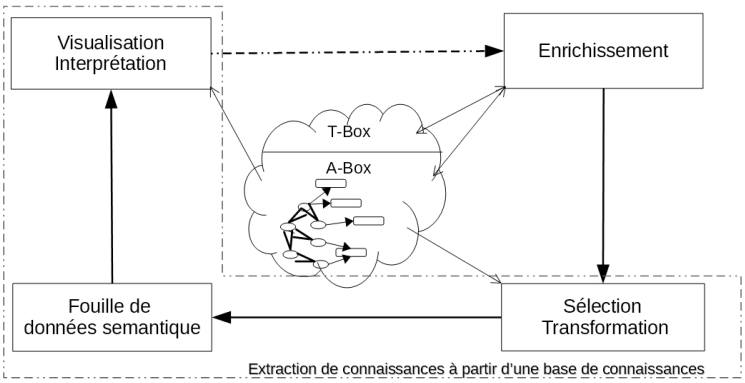


Figure 7.12 – Relation entre l'extraction de données et le Web sémantique dans le système présenté.

À l'aide de ses connaissances du domaine et de l'ontologie (TBox), l'utilisateur définit les paramètres d'analyse, y compris les requêtes spatio-temporelles, la sélection des attributs, le choix de l'algorithme et ses paramètres. Le résultat provenant du module de requête sémantique est converti en transactions dont les attributs sont sélectionnés sur demande avant d'être renvoyés à l'algorithme de fouille. En se basant sur le résultat de la fouille et de la visualisation, l'utilisateur peut ensuite enrichir de nouveau la base de connaissances pour les prochaines analyses. La Figure 7.12 décrit une autre vue de l'application de méthodes de fouille. Elle se concentre sur la relation entre le Web sémantique et l'extraction de connaissance.

Dans le système, trois algorithmes de fouille sont implémentés à l'aide de l'application Weka.

— *Apriori* pour la recherche des règles d'association.

— *Part* pour la recherche des règles de classification.

— *K-moyennes* pour la recherche de partition - «clusters».

Chaque algorithme nécessite ses propres paramètres d'entrée qui ont une influence directe sur le résultat final du processus. Pour ces raisons, l'utilisateur doit posséder des connaissances du domaine et consulter les ontologies qui les décrivent pour pouvoir choisir l'algorithme ainsi que les paramètres adaptés au contexte donné.

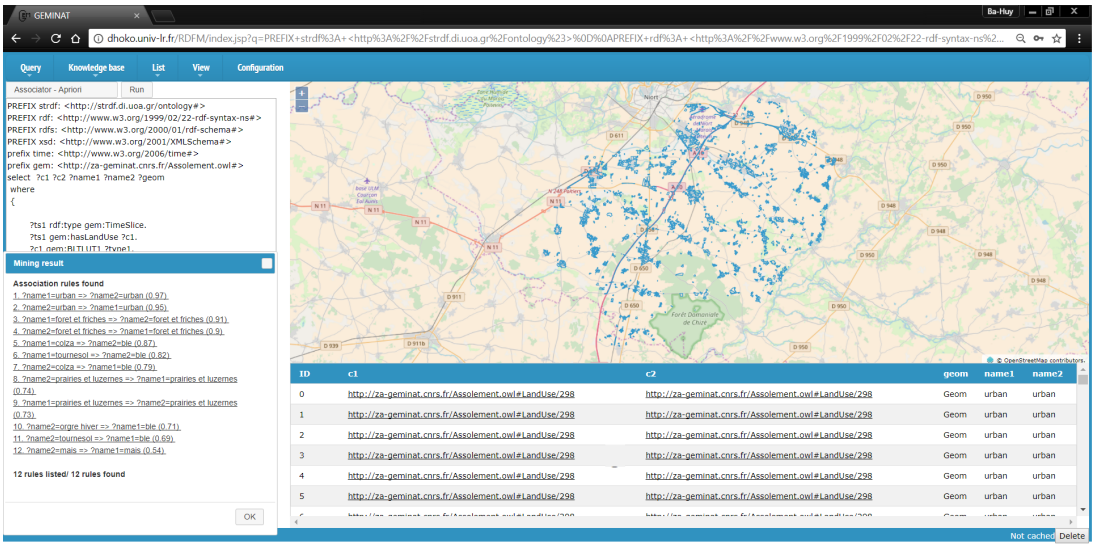


Figure 7.13 – Implémentation des techniques de fouille dans l'analyse de données.

La Figure 7.13 représente l'interface principale du système avec le but d'extraire des connaissances à partir de la base Geminat. Le processus d'extraction de connaissances se réalise de manière suivante :

Sélection, pré-traitement et transformation

L'utilisateur doit choisir les données souhaitées en formulant une requête stSPARQL. Le résultat retourné est converti en format tabulaire pour être adaptée aux méthodes de fouille.

Fouille de données

Ensuite, il sélectionne l'algorithme et les paramètres adaptés au contexte ainsi que les attributs pour la fouille. Les données sectionnées sont prise en charge par l'algorithme de fouille choisi.

Visualisation et validation

Le résultat de la fouille est listé dans un tableau grâce auquel, l'utilisateur peut sélectionner l'information souhaitée, par exemple une règle ou un cluster, pour accéder aux détails. Les individus concernant l'information choisie sont ensuite affichés. Dès ce moment-là, il est possible de consulter à nouveau les informations affichées avec le système. S'il s'agit des informations géométriques, elles peuvent être affichées sur une carte. De plus, on peut obtenir plus de détails sur ces informations en envoyant de nouvelles requêtes à la base de connaissances. De cette façon, à l'aide des technologies du Web sémantique, l'utilisateur peut non seulement connaître le résultat de la fouille, mais aussi les détails supplémentaires liés à ce dernier fournis par la base de connaissances.

1. ?y1=Urban et ?y2=Urban et ?y3=Urban => ?y4=Urban (0.98)
2. ?y1=Forêt et friches et ?y2=Forêt et friches et ?y3=Forêt et friches => ?y4=Forêt et friches (0.97)
3. ?y1=Tournesol et ?y2=Blé et ?y3=Colza => ?y4=Blé (0.92)
4. ?y1=Colza et ?y2=Blé et ?y3=Tournesol => ?y4=Blé (0.91)
5. ?y1=Tournesol et ?y2=Blé et ?y3=Tournesol => ?y4=Blé (0.89)
6. ?y1=Blé et ?y2=Blé et ?y3=Colza => ?y4=Blé (0.89)
7. ?y1=Blé et ?y2=Blé et ?y3=Tournesol => ?y4=Blé (0.88)
8. ?y1=Autre culture et ?y2=Autre culture et ?y3=Autre culture => ?y4=Autre culture (0.85)
9. ?y1=Prairies et luzernes et ?y2=Prairies et luzernes et ?y3=Prairies et luzernes => ?y4=Prairies et luzernes (0.76)
10. ?y1=Mais et ?y2=Mais et ?y3=Mais => ?y4=Mais (0.61)

Tableau 7.4 – Les règles d'association trouvées par l'Apriori pour une succession de cultures sur quatre années.

7.4.2 Applications

Afin de montrer la faisabilité du système dans l'extraction de connaissances et l'utilité de la boucle dans ce processus, nous prenons deux exemples suivants.

§1 Fouille de données sémantiques

Une recherche des règles de succession de cultures sur quatre années a été réalisée. Le tableau 7.4 liste les règles principales obtenues par l'algorithme Apriori avec la confiance=0.5 et le support=0.01. À noter qu'un regroupement (Tableau 7.6) de cultures est appliqué dans cette recherche.

Les règles obtenues sont presque identiques à celles présentées par [Lazrak *et al.*, 2009] appliquant des modèles de Markov cachés sur la base de données

d'Assolement pour la période 1994-2008. De la même manière, les hypothèses déjà démontrées par [Lazrak, 2012] peuvent être aussi vérifiées par le système :

— Les successions de cultures sont relativement courtes et organisées dans le temps. En d'autres termes, la culture d'une année dépend des cultures pratiquées au même endroit quelques années précédentes ;

— L'agriculteur alloue une culture à une parcelle en tenant compte du voisinage de cette parcelle. En d'autres termes, les cultures sont organisées dans l'espace et dépendent de leur voisinage proche ;

— Le choix d'une succession de cultures en un point de l'espace dépend du choix des successions de cultures voisines.

§2 Relation entre l'extraction et l'enrichissement de connaissances

Afin de montrer l'intérêt de l'introduction de la boucle d'extraction et d'enrichissement de connaissances, un exemple de la recherche des règles de succession de cultures sur deux années est pris.

1. Extraction de règles d'association : première analyse

On réalise une fouille de données sur la succession de cultures sur deux années. Le tableau 7.5 décrit le résultat obtenu par l'algorithme Apriori avec le support=0.01 et la confiance=0.5.

1. ?c1=Bâti => ?c2=Bâti (0.98)
2. ?c1=Bois ou haie => ?c2=Bois ou haie (0.96)
3. ?c1=Péri-village => ?c2=Péri-village (0.95)
4. ?c1=Vigne => ?c2=Vigne (0.93)
5. ?c1=Prairie age inconnu => ?c2=Prairie age inconnu (0.64)
6. ?c1=Luzerne => ?c2=Luzerne (0.56)
7. ?c1=Prairie permanente (age>3 ans) => ?c2=Prairie permanente (age>3 ans) (0.56)

Tableau 7.5 – Règles d'association trouvée par l'Apriori pour une succession de cultures en deux années.

Bien que le résultat puisse identifier des successions stables comme montrées par les quatre premières règles, il restait encore des successions dominantes qui ne sont pas listées comme souhaitées. Par conséquent, l'expert a proposé ensuite un regroupement de cultures qui lui convient. Il enrichit donc la base de connaissances par son expertise.

Culture	**Type de culture**
Blé	Blé, blé barbu, céréale
Tournesol	Tournesol, ray-grass suivi Tournesol
Colza	Colza
Urbain	Bâti, péri-village, route
Prairies et Luzernes	Prairie permanente, prairie année 1, prairie temporaire (2- 3 ans), prairie âge inconnu, luzerne 1 an, luzerne 2 ans, luzerne 3 ans, luzerne > 3 ans
Maïs	Maïs, ray-grass suivi maïs
Forêts et friches	Forêt ou haie, friche
Orge d'hiver	Orge d'hiver
Ray-grass	Ray-grass, ray-grass suivi ray-grass
Pois	Pois
Autres	Orge de printemps, vigne, jachère spontanée juin, moha, lin, avoine, Trèfle, fèverole, ray-grass suivi labour, ray-grass suivi inconnu, jachère spontanée suivie labour, mélange céréale légumineuse, culture printemps, moutarde, jardin/culture maraîchère, Sorgho/Millet, Sorgho, Millet, Labour, Tabac, Autre culture

Tableau 7.6 – Une proposition de regroupement des cultures de la zone atelier[Lazrak *et al.*, 2010].

2. Enrichissement de la base de connaissances

En constatant que le résultat n'était pas satisfaisant à cause d'une grande quantité de types de cultures, le regroupement ci-dessous (Tableau 7.6) a été réalisé. Pour cela, de nouvelles déclarations ont été ajoutées à la base de connaissances comme décrit par le code 7.3.

3. Extraction des règles d'association : nouvelle analyse

L'expert a réalisé à nouveau l'analyse sur la succession de cultures pendant deux années, mais cette fois ci, le regroupement de cultures est appliqué.

1. ?c1=Urbain => ?c2=Urbain (0.97)
2. ?c1=Forêt et friches => ?c2=Forêt et friches (0.91)
3. ?c1=Colza => ?c2=Blé (0.87)
4. ?c1=Tournesol => ?c2=Blé (0.82)
5. ?c1=Prairies et luzernes => ?c2=Prairies et luzernes (0.73)

Tableau 7.7 – Les règles d'association trouvées par l'Apriori pour une succession de cultures sur deux années.

Comme listées par le tableau 7.7, comparé au premier résultat, certaines règles ont disparues, la valeur de confiance d'autres a changé à cause du

regroupement. Pourtant, le regroupement a fait sortir de nouvelles règles dominantes, la règle 3 et 4 qui n'existaient pas dans le premier résultat, comme souhaitées par l'expert.

L'exemple a démontré l'hypothèse que l'enrichissement de la base de connaissances peut améliorer le résultat des méthodes de fouille ainsi que l'utilité de ce processus dans l'extraction de connaissances à partir de données sémantiques.

Conclusion du chapitre

Ce chapitre présente les prototypes permettant l'intégration et l'exploitation de données spatio-temporelles hétérogènes à l'aide des ontologies. Le premier suit l'approche médiation qui implique le problème de fonctionnalité et de performances. Le suivant applique l'approche entrepôt avec une extension permettant l'exploitation des données intégrées à l'aide des méthodes de fouille de données.

Nous avons montré comment l'introduction de l'enrichissement de connaissances peut améliorer le résultat des méthodes de fouille de données sémantique. Ainsi, avec le support des technologies du Web sémantique, les processus d'extraction et d'enrichissement de connaissances forment une boucle pour faciliter l'analyse de la base de connaissances.

Bien que les exemples n'aient été réalisés que sur les sources de données de la zone atelier, les systèmes peuvent être appliqués à d'autres jeux de données tant qu'il existe une ontologie qui les modélise.

Troisième partie

Conclusions et perspectives

Chapitre 8

Conclusion et perspectives

Les travaux menés dans ce manuscrit s'inscrivent dans le cadre du projet interdisciplinaire GEMINAT (GEo-connaisances des MIlieux NATurels) qui a pour objectif d'exploiter des données environnementales hétérogènes. Pour cela, trois axes de recherches ont été étudiés, premièrement, l'intégration sémantique de données hétérogènes, deuxièmement la gestion de connaissances, troisièmement l'exploitation de connaissances. Leur implémentation au sein d'un système permettant l'intégration et l'exploitation de données environnementales a été présentée. Ce chapitre présente les principaux apports et le bilan de l'approche proposés pour ensuite conclure sur les perspectives.

Sommaire

8.1 Bilan 164
8.2 Perspectives 166

8.1 Bilan

En introduction de ce manuscrit, nous avons présenté la problématique scientifique motivant nos travaux de la manière suivante : l'intégration sémantique et l'exploitation de données spatio-temporelles, typiquement de données environnementales. Ainsi, nous avons montré tout au long de ce document de quelle manière nous envisagions la résolution d'une telle problématique, à la fois en termes de modélisation, de techniques et d'implémentation. Pour atteindre cet objectif, nous nous appuyons sur les travaux de différents domaines de recherches :

— **Intégration de données** : Les différents types d'hétérogénéités de données ainsi que les approches d'intégration de données ont été étudiés afin d'en choisir une adaptée à notre contexte.

— **Gestion de données** : La modélisation de données spatio-temporelles, plus particulièrement celle basée sur l'approche ontologique, a été étudiée en vue de développer des ontologies spatio-temporelles. En effet, ces dernières permettent non seulement de modéliser les connaissances du domaine mais aussi d'intégrer des sources hétérogènes. En parallèle, les technologies du Web sémantique ont été choisies et examinées pour la construction et la gestion de la base de connaissances.

— **Exploitation de connaissances** : L'exploitation de connaissances est réalisée, à la base, grâce aux technologies du Web sémantique. Des méthodes de fouilles automatiques sont également proposées pour favoriser cette exploitation.

À partir de ces études, trois apports essentiels nous semblent découler de nos travaux. Le premier concerne une proposition de modélisation ontologique de données environnementales prenant en compte la dimension spatiale, la dimension temporelle et l'évolution des objets. Le second porte sur la conception d'un prototype permettant l'intégration sémantique et l'exploitation de données spatio-temporelles hétérogènes. Le dernier relève de l'application de processus d'extraction de connaissances aux données spatio-temporelles sémantiques.

§1 Conception d'un modèle pour l'intégration de données environnementales

L'hétérogénéité de données se réfère à la façon dont les données sont organisées ou interprétées dans les différents systèmes. De toute évidence, l'autonomie de la conception constitue un obstacle majeur à la réalisation de l'interopérabilité sémantique, ou l'échange d'information utile, entre systèmes. L'intégration de données sémantiques est le processus d'utilisation d'une représentation conceptuelle de données et leurs relations pour gérer les hétérogénéités. Au cœur de cette intégration se trouve le concept d'ontologie qui fait communiquer les systèmes d'information via une conceptualisation. Cette dernière permet d'expliciter et

de préciser le sens des données pour être interprété correctement par différents systèmes.

Par conséquent, une ontologie a été introduite afin d'intégrer des données environnementales hétérogènes. Elle se compose de trois composantes : la composante spatiale, la composante temporelle, et la composante sémantique. Les ontologies du temps et de l'espace ont été réutilisées pour représenter les différents concepts et les relations temporelles et spatiales. Le processus de construction de l'ontologie, la réutilisation des ontologies et la construction des ontologies pour chaque source de données a été décrit pour notre cas d'étude.

§2 Conception et développement d'un système d'intégration et d'exploitation de données environnementales hétérogènes

Des prototypes ont été développé au fur et à mesure afin de concrétiser et démontrer la faisabilité de l'approche proposée. Dans ces prototypes, deux techniques d'intégration de données ont été appliquées. Le premier système s'est appuyé sur un médiateur qui contient le schéma global permettant l'accès à toutes les sources de données locales. Dans ce cas, l'outil de traduction D2RQ a été utilisé afin d'exposer ces sources en graphes RDF virtuels. Le deuxième système a suivi l'approche entrepôt qui utilise les processus ETL pour matérialiser les sources relationnelles dans un triplestore géospatial qui a ainsi formé la base de connaissances. Dans ce système, la plateforme D2RQ a aussi été utilisée pour transformer ces données en triplets RDF.

Un mécanisme de raisonnement spatio-temporel a été présenté pour chaque système. Dans l'architecture de médiation, un raisonneur a été utilisé alors que les fonctions spatiales du triplestore géospatial et les requêtes SPARQL Update ont été appliquées dans l'architecture de l'entrepôt. Logiquement, le premier système a des problèmes de performances alors que le deuxième implique des problèmes liées à la réplication de données.

§3 Application de l'extraction de connaissances à partir d'une base de connaissances spatio-temporelle

Afin de faciliter les tâches d'analyse de données, des méthodes de fouille de données ont été adaptées au deuxième système. Pour cela, le résultat des requêtes sémantiques est transformé en données tabulaires. En outre, les autres étapes du processus de l'extraction de connaissances ont été exploitées grâce aux technologies du Web sémantique. En effet, nous avons introduit l'enrichissement de données comme une étape optionnelle débutant à l'extraction de connaissances. Celle-ci est alors considérée comme une boucle allant de l'étape d'enrichissement à l'évaluation et l'interprétation du résultat. Ainsi, l'enrichissement de la base de connaissances peut améliorer le résultat des techniques de fouille, à l'inverse, les techniques de fouille permettent de mieux comprendre la base et d'en décou-

vrir de nouvelles connaissances. C'est grâce à ces dernières et aux connaissances d'expert qu'on peut a posteriori enrichir à nouveau la base.

En ce qui concerne l'implémentation, trois algorithmes de fouille ont été implémentés : Apriori pour la recherche des règles d'association, Part pour la recherche des règles de classification et K-means pour la recherche des clusters. Ces algorithmes nécessitent différents paramètres d'entrée et s'appliquent aux différents types d'analyses. Comme souligné, c'est à l'utilisateur de les déterminer à l'aide de ses connaissances d'expert et de l'ontologie construite. De plus, c'est grâce à l'ontologie qu'ils peuvent éditer des requêtes sémantiques pour sélectionner les données souhaitées. Tout cela démontre le rôle primordial des ontologies et des technologies du Web sémantique dans les processus d'extraction de connaissances à partir de donnés sémantiques.

8.2 Perspectives

Nos travaux ouvrent plusieurs voies de recherche pour l'amélioration du système présenté, à l'égard non seulement des performances mais aussi du fonctionnement. Ainsi, le système pourrait trouver des applications pour l'intégration et l'exploitation d'autres types de données, à références spatiales et temporelles. Cette section liste quelques-unes des perspectives potentielles de ce travail de recherche.

§1 Approfondir l'intégration de données spatio-temporelles

Bien que fonctionnel, les prototypes présentés peuvent être encore améliorés. En effet, ils présentent des inconvénients liés aux approches utilisées, soit dans les performances et le fonctionnement du système, soit dans le traitement et la mise à jour des données. Par conséquent, la troisième approche d'intégration de données, nommée l'approche hybride, est envisagée. Celle-ci ne matérialise qu'une partie des sources de données, le reste est interrogé directement depuis les sources. Elle est reconsidérée pour les raisons suivantes :

— Les outils de mapping comme Ontop [Calvanese *et al.*, 2016] deviennent plus matures. En effet, les expériences présentées dans [Lanti *et al.*, 2015] ont montré que malgré quelques faiblesses, Ontop est considérablement plus rapide que le triplestore standard Stardog, en exploitant les moteurs de base de données hautement optimisés.

— Les outils de mapping commencent à prendre en compte la correspondance des données et des relations spatiales. L'extension Ontop spatial [Bereta et Koubarakis, 2016] récemment publié en est un exemple typique.

La Figure 8.1 décrit l'architecture du système envisagé suivant l'approche hybride. Les relations spatiales et temporelles entre les objets, l'ontologie et de

nouvelles déclarations déduites par des règles métiers sont stockées dans un triplestore, celui-ci avec l'ensemble de données RDF virtuelles accessibles par des endpoints forment la base de connaissances.

Une autre alternative possible pour un tel système est d'utiliser l'outil Ontop avec son extension spatiale permettant l'exposition des données et de relations spatiales en graphes RDF virtuels. Pour cela, les sources de données hétérogènes doivent se réunir en une seule base, avec laquelle Ontop Spatial interagit pour donner l'accès aux données via un endpoint SPARQL (Figure 8.2). Cette architecture permet une déduction des relations spatiales à la volée et évite donc la matérialisation de ces relations dans la base de connaissances. Par conséquent, cette dernière ne contient que les ontologies, les relations temporelles et de nouvelles connaissances enrichies par l'expert.

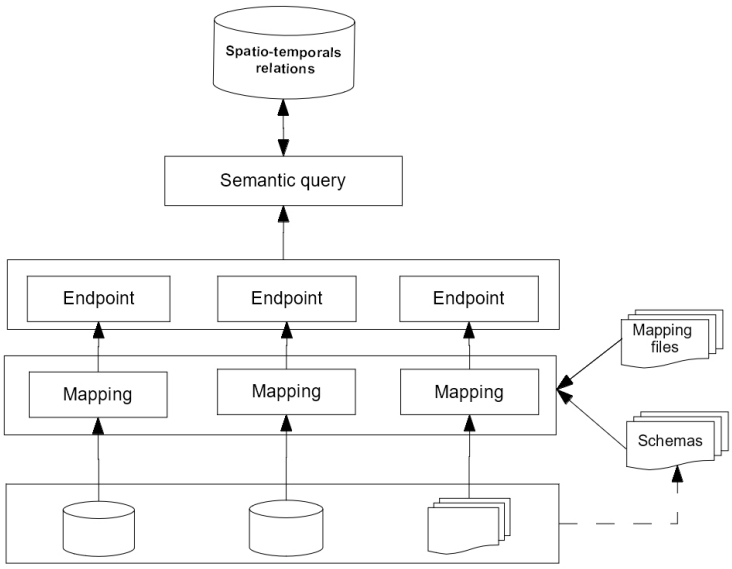


Figure 8.1 – Architecture hybride pour l'intégration de données.

§2 Vers le Web de données

Afin de partager les données collectées avec d'autres équipes de chercheurs, la publication d'une partie de la base de connaissances sous forme de Web des données est possible. À cet effet, on souhaite implémenter un mécanisme d'authentification permettant le contrôle d'accès aux données partagées. En effet, un tel mécanisme, comme proposé dans [Hollenbach *et al.*, 2009, Villata *et al.*, 2011] facilite la gestion d'accès aux données pour chaque profil d'utilisateur ou encore de journaliser les changements pour les abandonner si nécessaire.

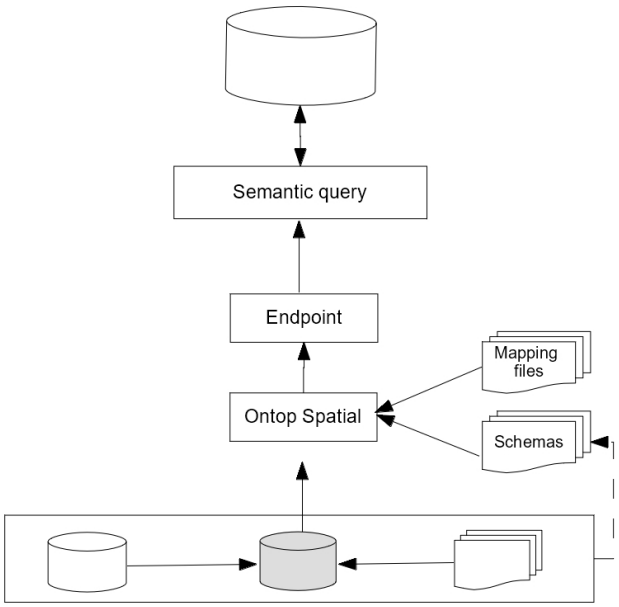


Figure 8.2 – Architecture hybride avec Ontop spatial pour l'intégration de données spatio-temporelles.

D'autre part, on souhaite permettre l'annotation des données collectées. Ainsi, nous proposons une modélisation des annotations comme illustrée par la Figure 8.3. Une ou plusieurs annotations sont attachées à un élément spatio-temporel. Dans cet objectif, il est possible de réutiliser le vocabulaire FOAF pour la représentation des informations des utilisateurs et l'ontologie OWL-Time pour la journalisation des annotations. De cette façon, nous pouvons comparer les différentes annotations et connaître leur origine pour les valider et les utiliser ultérieurement.

Une autre perspective intéressante concerne le mapping entre les concepts (TBox) et les instances (ABox) de la base de connaissances avec ceux provenant du Web des données, par exemple, ceux d'Argrovoc [1] ou de DBpedia [2]. De cette manière, nous pouvons enrichir la base non seulement par des connaissances des experts intéressés mais aussi par des sources de données externes. Par exemple, la culture Tournesol dans notre base http://za-geminat.cnrs.fr/LandUse/137 correspond à :

— http://dbpedia.org/resource/Helianthus chez DBpedia

— ou http://aims.fao.org/aos/agrovoc/c_3539.html chez Agrovoc.

1. http://aims.fao.org/fr/agrovoc/linked-open-data
2. http://fr.dbpedia.org/sparql

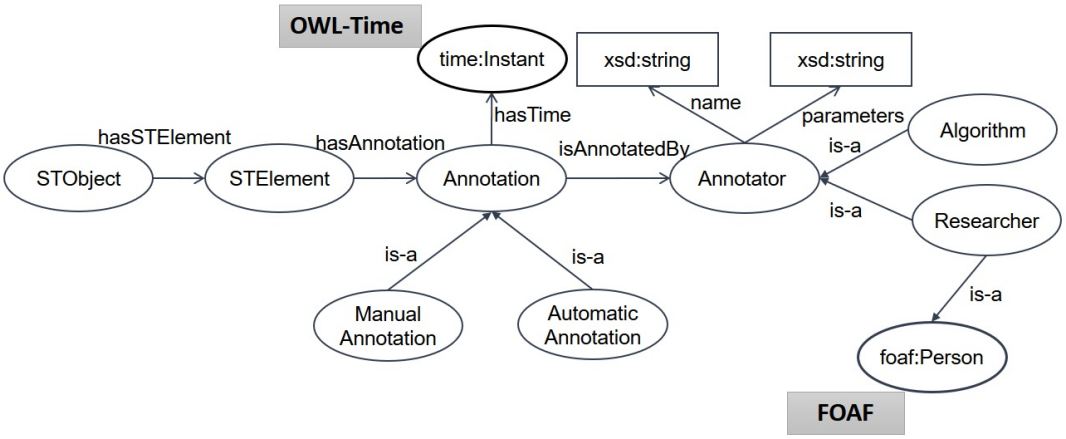


Figure 8.3 – Une annotation sur le composante sémantique.

§3 Application du système

À court terme, afin de compléter les objectifs initiaux de la thèse, l'intégration des autres bases de données de la zone atelier est prévue. De nouvelles rencontres seront nécessaires pour identifier de nouveaux besoins et les améliorations futures à apporter. Cela monstre l'exigence d'une collaboration étroite entre nous, les informaticiens, et les experts dans le développement et le déploiement du système complet.

À moyen terme, comme le système n'oblige pas de conditions particulières, son application pour l'intégration et l'exploitation des bases de données hétérogènes, à références spatiales et temporelles, d'un autre fournisseur ou dans un autre domaine peut être considérée. Cela ne nécessite qu'une adaptation de l'ontologie au nouveau contexte. C'est grâce à des retours de ces applications que nous pouvons perfectionner le système au fur et à mesure.

Dans le cadre du projet DYPOMAR (DYnamiques POrtuaires, Milieux urbains et environnement mARitime) de l'établissement, les travaux présentés pourront être étendus aux trajectoires [Mefteh, 2013] pour l'étude et l'analyse du trafic des ports de la Rochelle.

Quatrième partie

Annexes

Annexe A

Intégration sémantique de données

L'intégration sémantique de données provenant de sources hétérogènes suivant l'approche entrepôt se déroule en plusieurs étapes comme illustré par la figure A.1.

1. **Construction des ontologies** : Les ontologies sont construites à partir des schémas des sources relationnelles.
2. **Définition des correspondances** : Dans cette étape, on définit les correspondances entre les schémas des sources et les ontologies construites.
3. **Traduction des données** : Les données issues des sources hétérogènes sont traduites en triplet RDF à l'aide des correspondances définies dans l'étape précédente.
4. **Construction de la base de connaissances** : La base de connaissances est construite en chargeant les modèles développés à l'étape 1 et les individus obtenus à l'étape 3, qui constituent respectivement sa TBox et son ABox.

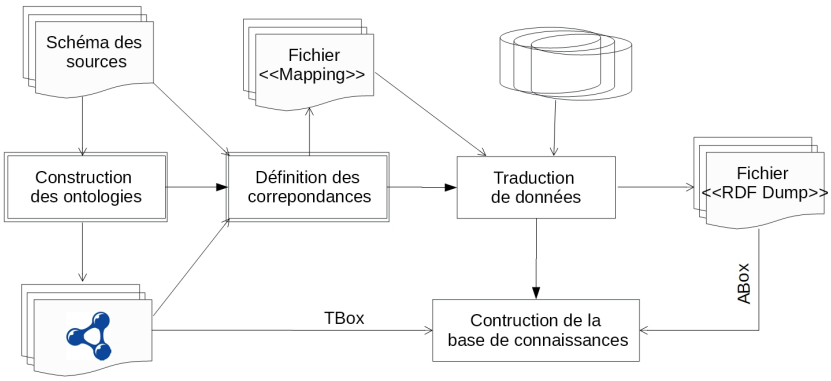


Figure A.1 – Les étapes de l'intégration sémantique de données suivant l'approche entrepôt

Il est à noter que les deux premières étapes se réalisent manuellement et nécessitent des connaissances sur les sources à intégrer. Pour ces raisons, divers

entretiens avec les gestionnaires de bases ont été réalisés afin de mieux les comprendre et de réaliser la saisie des métadonnées comme recommandé [1] par le réseau national des zones ateliers.

Comme souligné dans 3.2.1.2, en ce qui concerne les sources non relationnelles, il est recommandé de les transférer dans des bases de données relationnelles (PostgreSQL) avant de commencer les processus d'intégration.

Nous allons montrer comment réaliser ces étapes par la suite.

A.1 Construction des ontologies

Les ontologies présentées ont été développées à l'aide de l'outil Protégé [2]. L'ontologie du temps OWL-Time et le modèle stRDF sont accessibles en ligne [3]. La figure A.2 représente l'interface de l'éditeur comprenant une liste des concepts et la visualisation de ces concepts et de leurs relations.

Une fois qu'une ontologie est construite, il est possible de l'exporter sous plusieurs formats pour la charger ensuite dans notre système.

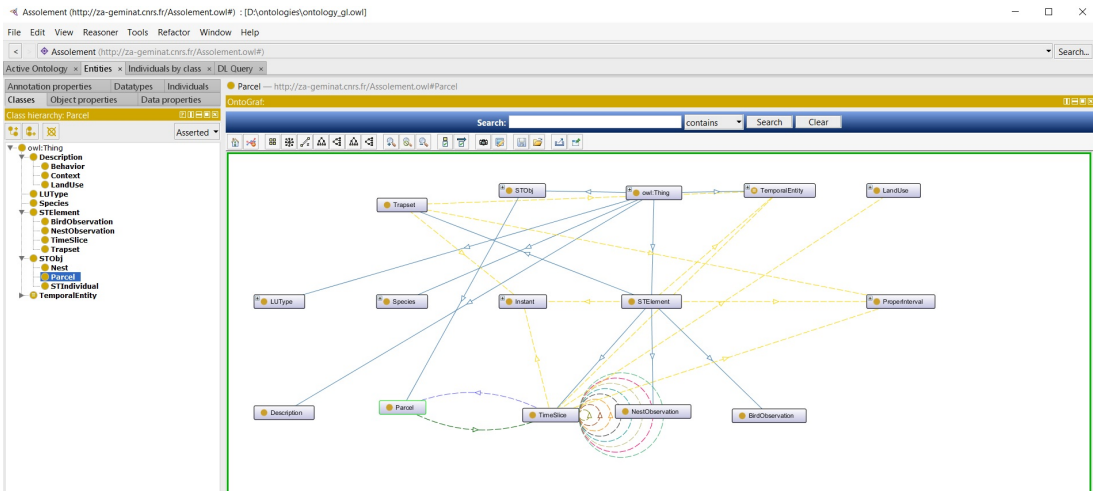


Figure A.2 – Construction des ontologies avec Protégé.

A.2 Enregistrement des ontologies

Les ontologies concernées sont enregistrées grâce à une page web développée sur les fonctions existantes de l'endpoint Strabon. L'endpoint Strabon est une application web qui prend en charge la construction et la mise à jour de la base ainsi que la réponse aux requêtes sémantiques. Les syntaxes acceptées sont

1. http://www.za-inee.org/fr/metaform
2. https://protege.stanford.edu/
3. https://www.w3.org/2006/time et http://strdf.di.uoa.gr/ontology#

RDF/XML, Turtle et N3. L'enregistrement des ontologies se fait en entrant leur contenu directement («Direct input») ou en choisissant un fichier («URI Input»). La figure A.3 présente l'enregistrement du modèle stRDF.

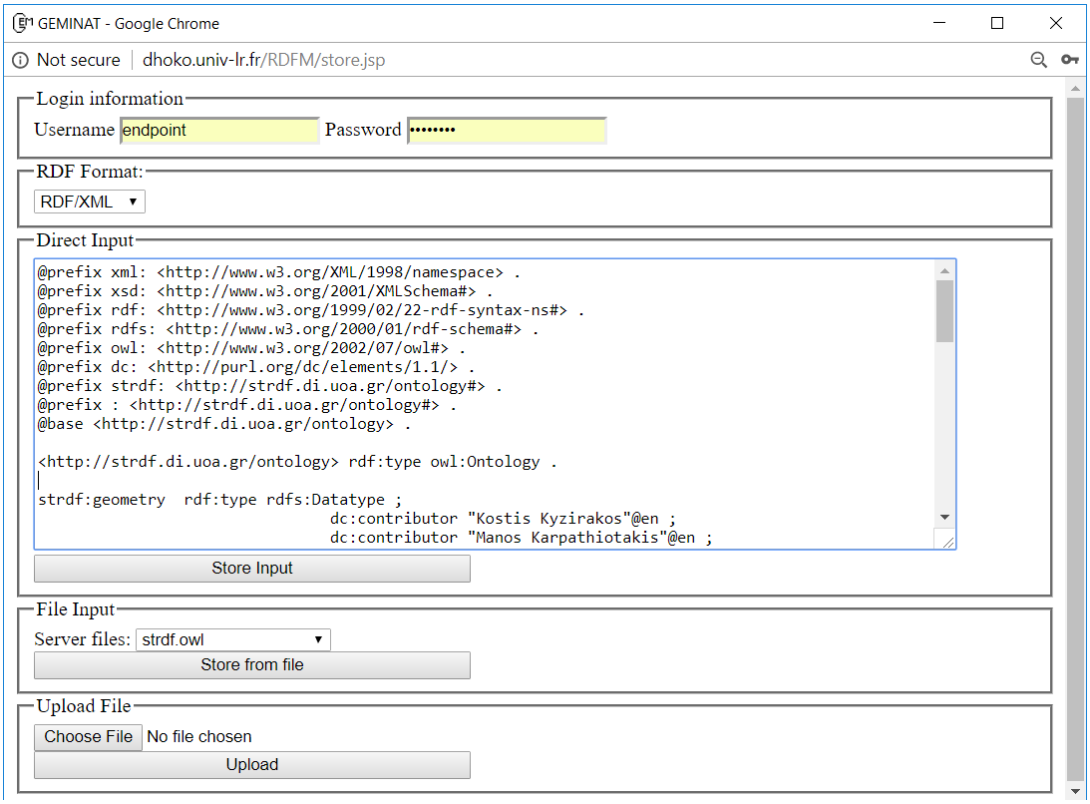


Figure A.3 – Enregistrement des ontologies.

A.3 Traduction des données

À partir des ontologies développées et du schéma des bases relationnelles, il est nécessaire de définir les correspondances qui seront utilisées par les outils de traduction, dans notre cas, l'outil D2RQ. Ce processus est réalisé de manière supervisée pour les deux raisons suivantes :

— Une traduction automatique implique la construction automatique d'une ontologie générique et la définition automatique des correspondances entre le schéma des bases et cette ontologie. Or, cette dernière est rarement identique aux ontologies développées.

— De plus, la traduction automatique ne prend pas en considération la pertinence des données, autrement dit, on traduit toute la base. Afin de réaliser la traduction, il est nécessaire de paramétrer l'outil et de définir des correspondances dans la syntaxe du système D2RQ.

Le code A.1 est utilisé pour la traduction des données de la table «Culture» de la base Assolement. Dans ce code, les 12 premières lignes ont pour but de définir des espaces de nom des vocabulaires utilisés, les 5 lignes suivantes paramètrent l'outil en donnant les informations pour la connexion à la source de données (base de données), les dernières lignes permettent de définir des individus de type «Landuse» avec «name» comme propriété, celle-ci est traduite à partir de la colonne «culture.libelle».

Code A.1 – Mise en correspondances des données de la table «Culture» de la base Assolement

```
@prefix map: <#> .
@prefix db: <> .
@prefix vocab: <vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ
    /0.1#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .
@prefix time: <http://www.w3.org/2006/time#>.
@prefix strdf: <http://strdf.di.uoa.gr/ontology#>.
@prefix gem: <http://za-geminat.cnrs.fr/Assolement.owl#>.
@prefix d2r: <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-
    server/config.rdf#> .
<> a d2r:Server;
d2r:sparqlTimeout 3000;
.
map:landuse a d2rq:Database;
d2rq:jdbcDriver "org.postgresql.Driver";
d2rq:jdbcDSN "jdbc:postgresql://localhost/landuse";
d2rq:username "postgres";
d2rq:password "123123";
.

# Table Culture
map:landuse a d2rq:ClassMap;
d2rq:dataStorage map:landuse;
d2rq:uriPattern "http://za-geminat.cnrs.fr/LandUse/@@culture.
    culture_id@@";
d2rq:class gem:Culture;
.
map:name a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:landuse;
d2rq:property gem:name;
d2rq:column "culture.libelle";
.
```

Une fois que les correspondances sont définies, il est possible d'utiliser la fonction «RDF Dump» pour traduire ces données. La figure A.4 représente la traduction de la table «Culture». Les données RDF obtenues sont affichées et stockées en même temps dans le système pour être chargées ensuite à la demande.

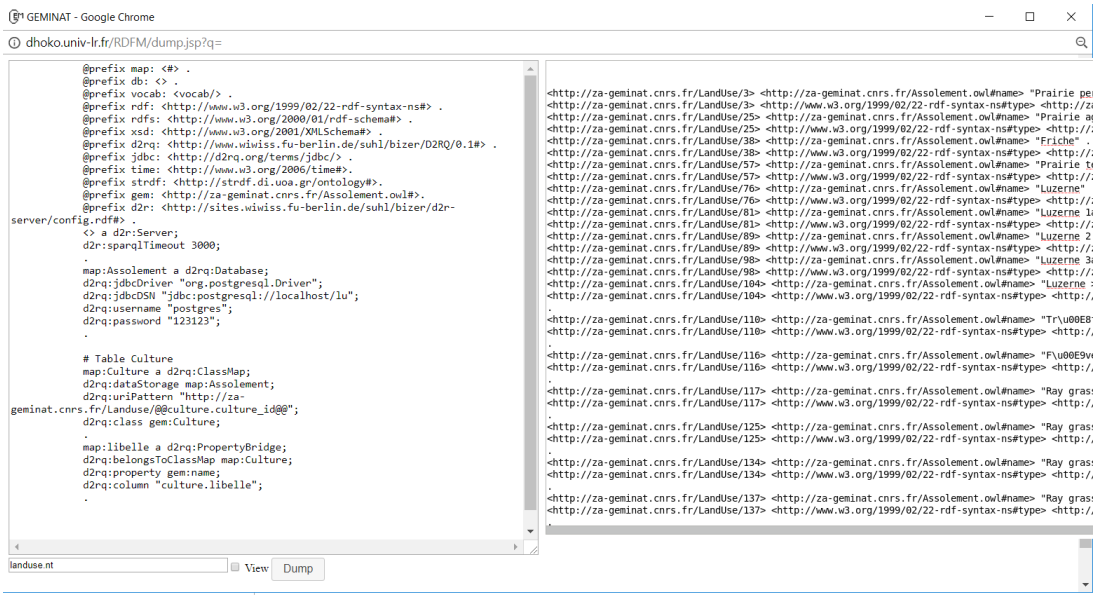


Figure A.4 – Traduction de données.

A.4 Chargement de données

Comme l'enregistrement des modèles (A.2) qui forment la TBox, les individus (ABox) sont chargés dans le système en entrant les données RDF directement («Direct input») ou en choisissant le fichier «Dump» existant dans le système. La figure A.5 montre comment charger les données «Culture» en fournissant les informations de connexion et les correspondances.

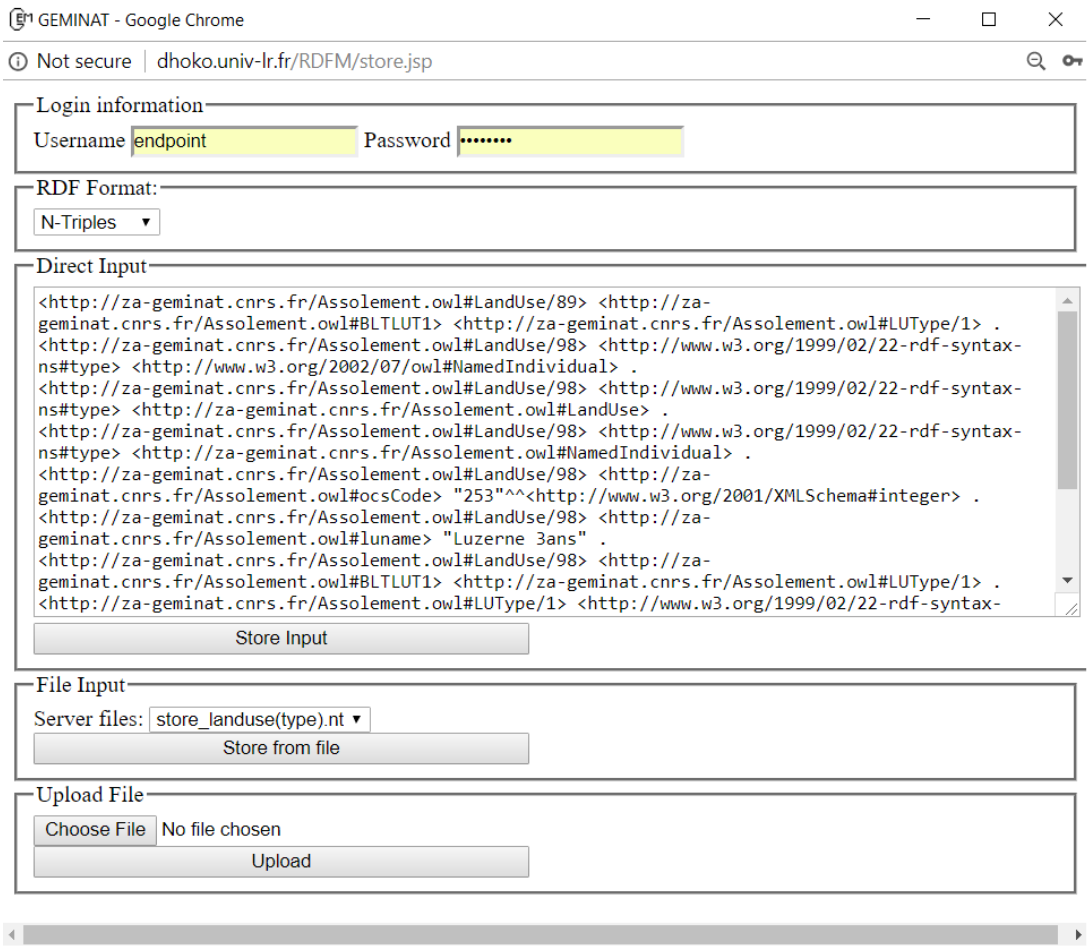


Figure A.5 – Chargement de données.

Annexe B

Prototype et exemples d'analyse

Dans cette section, nous présentons les fonctions proposées par notre prototype, hors celles déjà abordées dans l'annexe précédente. Quelques exemples d'analyse sur les données de la zone atelier seront réalisés pour démontrer ses capacités.

B.1 Présentation du prototype

L'interface principale du prototype se compose de quatre parties (Figure B.1) :

— Requête sémantique (1)

— Résultat de la fouille de données (2)

— Résultat d'une requête ou la liste des instances appartenant au motif sélectionné (3)

— Visualisation (4)

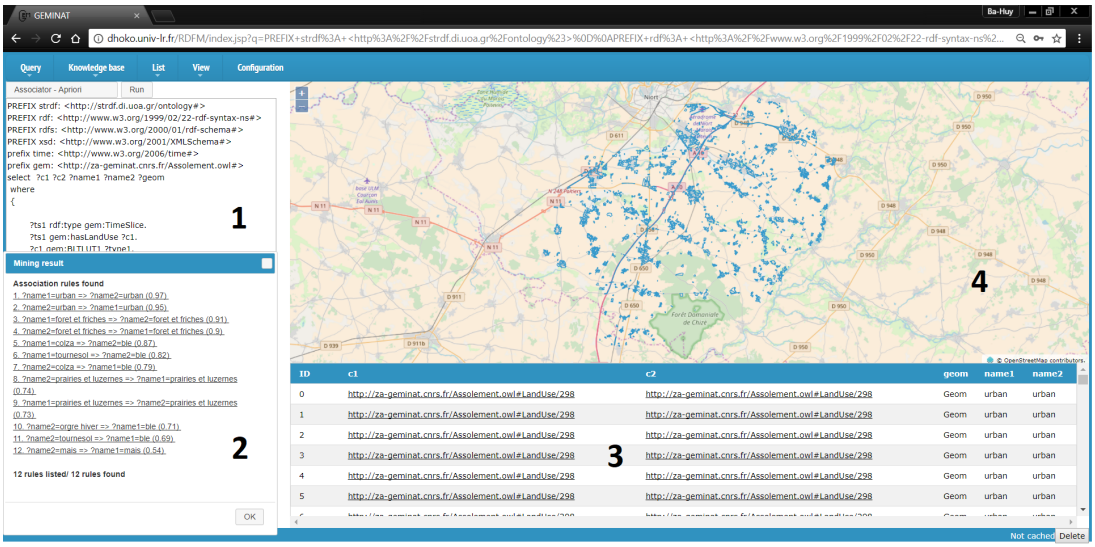


Figure B.1 – Interface principale du prototype.

Les fonctions proposées sont :

- Base de connaissances
  - Traduire des données («RDF Dump»)
  - Charger des données dans la base («Store»)
  - Mettre à jour la base par des requêtes («Update»)
- Requête sémantique
  - Exécuter des requêtes sémantiques en stSPARQL («Run»)
  - Sauvegarder et charger des requêtes («Save» et «Saved Queries»)
- Fouille de données («Run»)
- Visualisation et affichage
  - Visualisation cartographique («Map»)
  - Visualisation par graphe («Chart»)
  - Affichage du détail de chaque motif obtenu par la fouille
  - Affichage des informations concernant un élément de la liste
- Exportation du résultat («Export»)

B.2 Exemples d'analyse

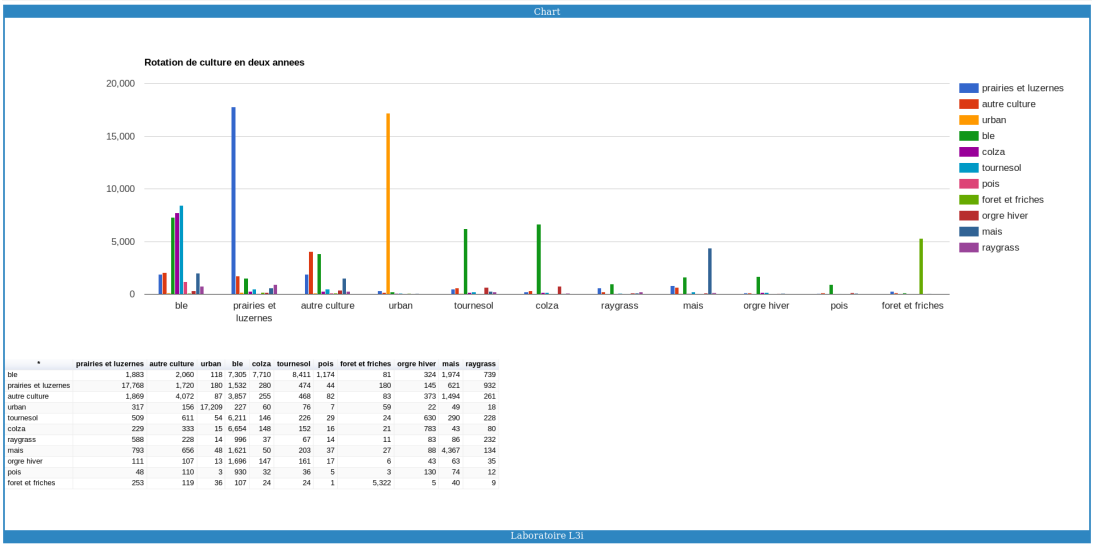
B.2.1 Rotation de cultures en deux années

La rotation de cultures (illustré par le Code B.1) peut être examinée en utilisant la relation *hasFiliation* (ligne 11) entre deux relevés qui forment chacun une tranche de temps de la parcelle étudiée.

Code B.1 – Rotation de cultures en deux années

```
1  PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX time: <http://www.w3.org/2006/time#>
4  PREFIX gem: <http://za-geminat.cnrs.fr/Assolement.owl#>
5  SELECT  *
6  WHERE
7  {
8   ?ts1 rdf:type gem:TimeSlice.
9   ?ts1 gem:hasLandUse ?c1.
10  ?c1 gem:lname ?name1.
11  ?ts1 gem:hasFiliation ?ts2.
12  ?ts2 gem:hasLandUse ?c2.
13  ?c2 gem:lname ?name2.
14  ?ts1 gem:pgeometry ?geom.
15 }
```

De la même manière, nous pouvons obtenir les statistiques comme illustrées par la Figure B.2.



*	prairies et luzernes	autre culture	urban	ble	colza	tournesol	pois	foret et friches	orgre hiver	mais	raygrass
ble	1,883	2,060	118	7,305	7,710	8,411	1,174	81	324	1,974	739
prairies et luzernes	17,768	1,720	180	1,532	280	474	44	180	145	621	932
autre culture	1,869	4,072	87	3,857	255	468	82	83	373	1,494	261
urban	317	156	17,209	227	60	76	7	59	22	49	18
tournesol	509	611	54	6,211	146	226	29	24	630	290	228
colza	229	333	15	6,654	148	152	16	21	783	43	80
raygrass	588	228	14	996	37	67	14	11	83	86	232
mais	793	656	48	1,621	50	203	37	27	88	4,367	134
orgre hiver	111	107	13	1,696	147	161	17	6	43	63	35
pois	48	110	3	930	32	36	5	3	130	74	12
foret et friches	253	119	36	107	24	24	1	5,322	5	40	9

Figure B.2 – Représentation statistique de la rotation de cultures.

B.2.2 Croisement entre l'assolement et les observations

Le code suivant (Code B.2) illustre le croisement entre l'assolement et les observations qui peut être réalisé en utilisant la relation temporelle (ligne 12) et spatiale (ligne 16).

Code B.2 – Croisement entre l'assolement et les observations

```
1  PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5  PREFIX time: <http://www.w3.org/2006/time#>
6  PREFIX gem: <http://za-geminat.cnrs.fr/Assolement.owl#>
7  SELECT *
8  WHERE
9  {
10  ?obsv a gem:Observation.
11  ?obsv gem:hasTime ?t.
12  ?t time:inside ?i.
13  ?ts gem:hasTime ?i.
14  ?obsv gem:geometry ?geomn.
15  ?ts gem:pgeometry ?geom.
16 FILTER (strdf:within(?geomn,?geom))
17  ?obsv gem:isObsvOf ?indv.
```

```
18   ?indv gem:belongTo ?sp.
19   ?sp gem:sname ?name.
20   ?ts gem:hasLandUse ?lu.
21   ?type gem:tname ?name2.
22  }
```

Le résultat de la requête peut être visualisé sur une carte comme la Figure B.3.

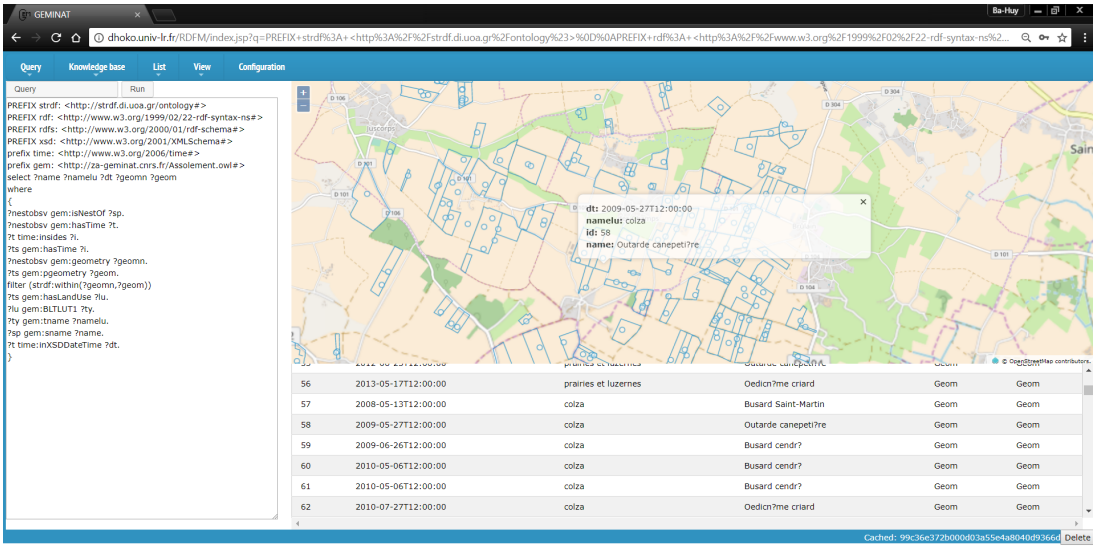
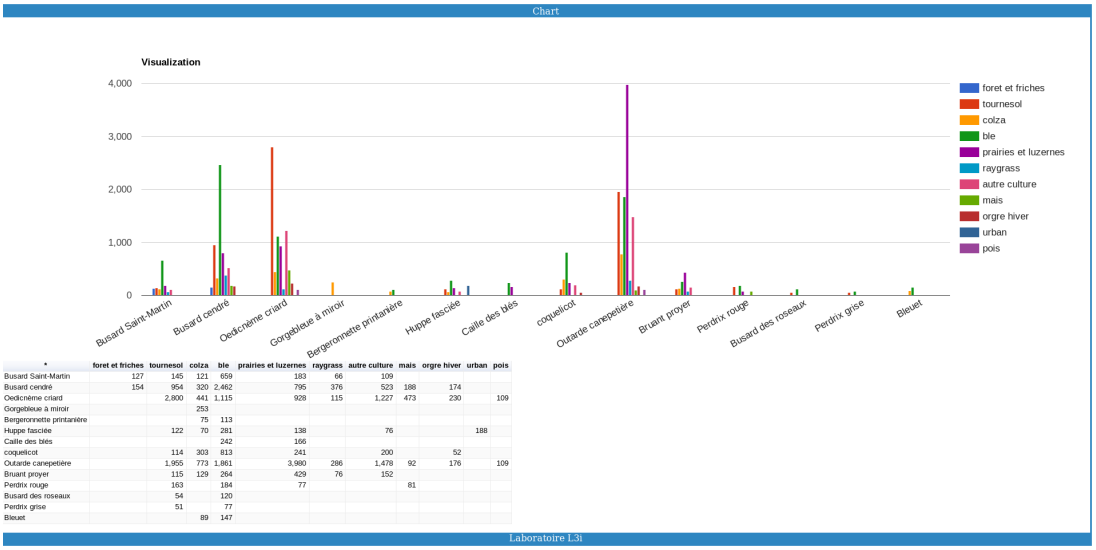


Figure B.3 – Visualisation cartographique d'un croisement de l'assolement et des observations.

Da la même façon, les informations statistiques sont obtenues et représentées par la Figure B.4.



*	foret et friches	tournesol	colza	ble	prairies et luzernes	raygrass	autre culture	mais	orgre hiver	urban	pois
Busard Saint-Martin	127	145	121	659	183	66	109				
Busard cendré	154	954	320	2,462	795	376	523	188	174		
Oedicnème criard		2,800	441	1,115	928	115	1,227	473	230		109
Gorgebleue à miroir			253								
Bergeronnette printanière			75	113							
Huppe fasciée		122	70	281	138		76			188	
Caille des blés				242	166						
coquelicot		114	303	813	241		200		52		
Outarde canepetière		1,955	773	1,861	3,980	286	1,478	92	176		109
Bruant proyer		115	129	264	429	76	152				
Perdrix rouge		163		184	77			81			
Busard des roseaux		54		120							
Perdrix grise		51		77							
Bleuet			89	147							

Figure B.4 – Représentation statistique d'un croisement des observations et des relevées de l'assolement.

B.2.3 Croisement entre les observations

Le code suivant (Code B.3) illustre un croisement entre les capture des micro-mammifères et les observations des nids qui se réalise en appliquant les relations ou les fonctions temporelles (ligne 19, 20, 22, 23, 24) et spatial (ligne 15).

Code B.3 – Croisement entre les observations

```
1  PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5  PREFIX time: <http://www.w3.org/2006/time#>
6  PREFIX gem: <http://za-geminat.cnrs.fr/Assolement.owl#>
7  PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
8  SELECT *
9  WHERE
10 {
11  ?trap a gem:Trapset.
12  ?trap gem:geometry ?geomt.
13  ?nestobs a gem:NestObsv.
14  ?nestobs gem:geometry ?geomn.
15 FILTER(strdf:distance(?geomn, ?geomt, uom:metre)<="300"^^xsd:
       double)
16  ?trap gem:hasTime ?t1.
17  ?nestobs gem:hasTime ?t2.
18  ?t1 time:inXSDDateTime ?dt1.
19 BIND(month(?dt1) as ?m1).
20 BIND(year(?dt1) as ?y1).
21  ?t2 time:inXSDDateTime ?dt2.
22 BIND(month(?dt1) as ?m2).
23 BIND(year(?dt1) as ?y2).
24 FILTER(?m1=?m2 && ?y1=?y2)
25  ?trap gem:numOfIndv ?num.
26  ?nestobs gem:isNestOf ?sp.
27  ?sp gem:sname ?name.
28 }
```

Le résultat de la requête peut être visualisé sur une carte comme la Figure B.5.

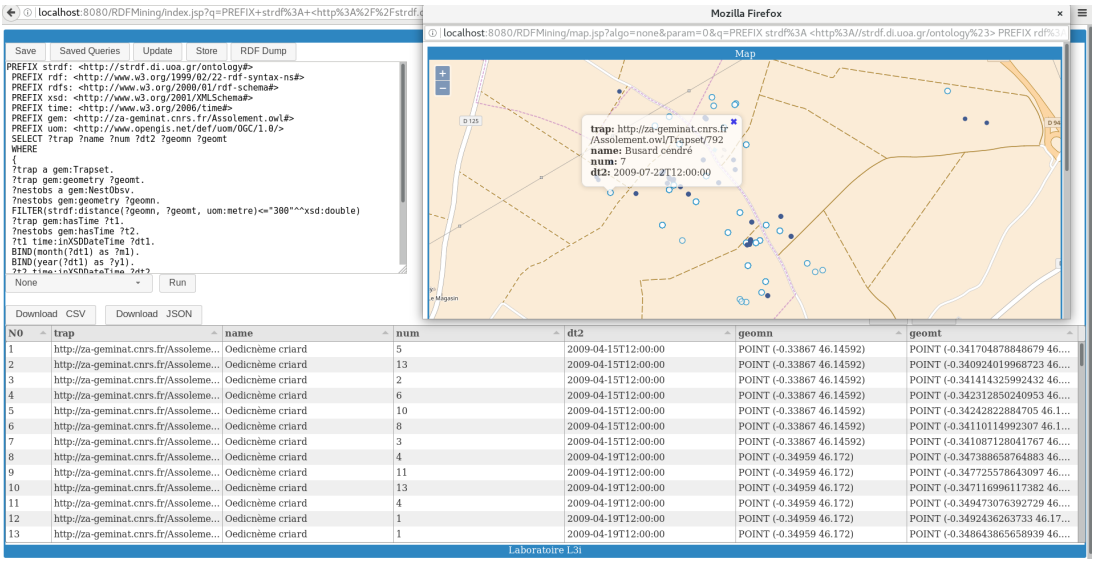


Figure B.5 – Visualisation d'un croisement entre les observations.

Da la même façon, les informations statistiques sont obtenues et représentées par la Figure B.6.

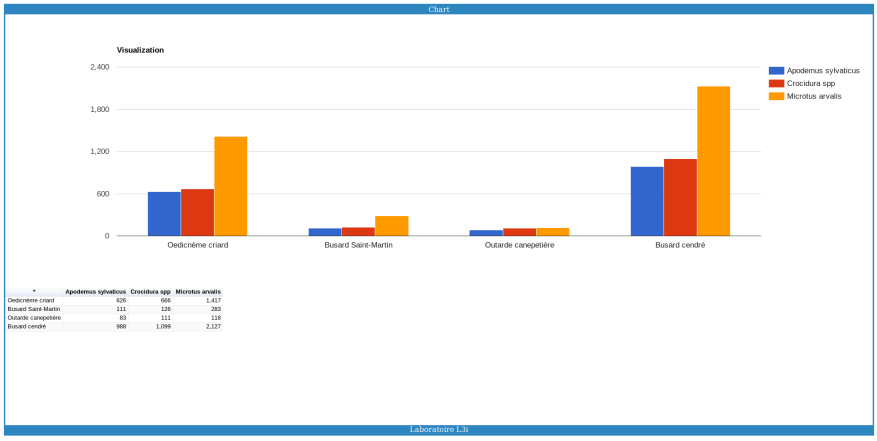


Figure B.6 – Représentation statistique d'un croisement entre les observations.

B.2.4 Détection des événements territoriaux

Pour détecter un événement territorial, par exemple, l'intégration entre deux parcelles, nous nous appuyons sur les relations spatio-temporelle entre les parcelles (ligne 15 et 19 du Code B.4). Le résultat est affiché sur une carte comme illustre la Figure B.7.

Code B.4 – Detection des événements d'intégration entre les parcelles

```
1  PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
```

```
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5  PREFIX  time: <http://www.w3.org/2006/time#>
6  PREFIX  gem: <http://za-geminat.cnrs.fr/Assolement.owl#>
7  SELECT *
8  WHERE
9  {
10  ?tsa rdf:type gem:TimeSlice.
11  ?tsa gem:hasTime ?intva.
12  ?intva time:intervalMeets ?intvb.
13  ?tsb gem:hasTime ?intvb.
14  ?tsa gem:pgeometry ?geoma.
15  ?tsb gem:pgeometry ?geomb.
16 FILTER(strdf:intersects(?geoma,?geomb) && strdf:area(?geomb)>
      strdf:area(?geoma))
17 }
```

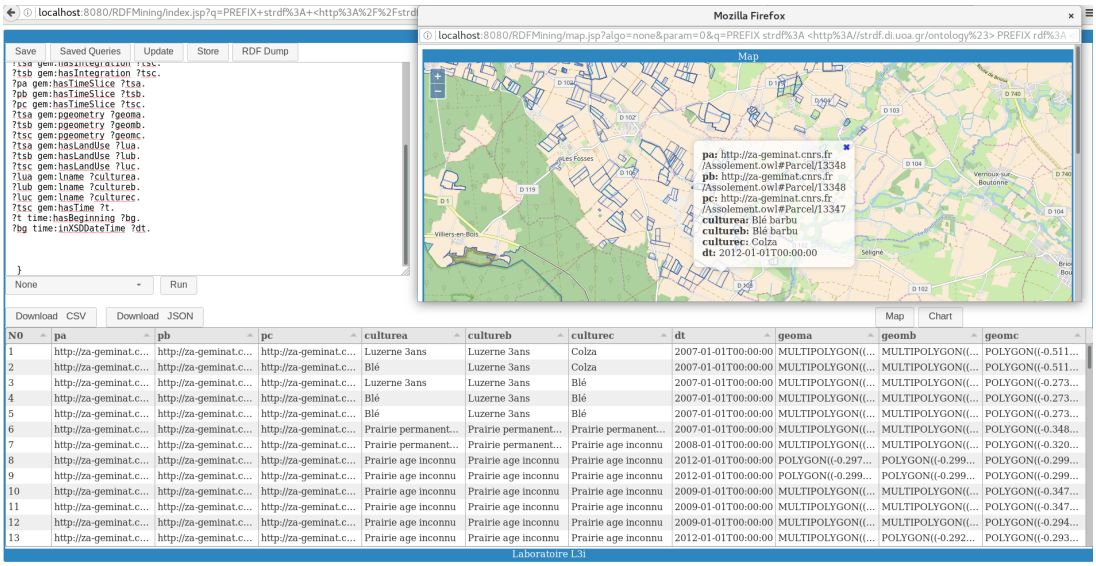


Figure B.7 – Une détection des événements d'intégration entre les parcelles.

Annexe C

Règles temporelles

Cette section présente les règles exprimées en requêtes SPARQL Update pour déduire les 14 relations temporelles utilisées dans nos prototypes.

C.1 time:intervalBefore et time:intervalAfter

Code C.1 – Inférence de la relation *intervalBefore* et *intervalAfter* entre deux intervalles de temps

```
1  PREFIX time: <http://www.w3.org/2006/time#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  INSERT
4  {
5   ?i time:intervalBefore ?j.
6   ?j time:intervalAfter ?i.
7  }
8  WHERE
9  {
10  ?i rdf:type time:ProperInterval.
11  ?j rdf:type time:ProperInterval.
12  ?i time:hasEnd ?endi.
13  ?j time:hasBeginning ?bgj.
14  ?endi time:inXSDDateTime ?dti.
15  ?bgj time:inXSDDateTime ?dtj.
16 FILTER(?dti<?dtj)
17 }
```

C.2 time:intervalContains et time:intervalDuring

Code C.2 – Inférence de la relation *intervalContains* et *intervalDuring* entre deux intervalles de temps

```
1
2  PREFIX time: <http://www.w3.org/2006/time#>
3  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4  INSERT
5  {
6   ?i time:intervalDuring  ?j.
7   ?j time:intervalContains ?i.
8  }
9  WHERE
10 {
11  ?i rdf:type time:ProperInterval.
12  ?j rdf:type time:ProperInterval.
13  ?i time:hasBeginning ?bgi.
14  ?i time:hasEnd ?endi.
15  ?j time:hasBeginning ?bgj.
16  ?j time:hasEnd ?endj.
17  ?bgi time:inXSDDateTime ?dti1.
18  ?endi time:inXSDDateTime ?dti2.
19  ?bgj time:inXSDDateTime ?dtj1.
20  ?endj time:inXSDDateTime ?dtj2.
21 FILTER(?dti1>?dtj1 && ?dti2<?dtj2)
22 }
```

C.3 time:intervalFinishes et time:intervalFinishedBy

Code C.3 – Inférence de la relation *intervalFinishes* et *intervalFinishedBy* entre deux intervalles de temps

```
1  PREFIX time: <http://www.w3.org/2006/time#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  INSERT
4  {
5   ?i time:intervalFinishes ?j.
6   ?j time:time:intervalFinishedBy ?i.
7  }
8  WHERE
9  {
10  ?i rdf:type time:ProperInterval.
11  ?j rdf:type time:ProperInterval.
12  ?i time:hasBeginning ?bgi.
13  ?i time:hasEnd ?endi.
14  ?j time:hasBeginning ?bgj.
15  ?j time:hasEnd ?endj.
16  ?bgi time:inXSDDateTime ?dti1.
```

```
17  ?endi time:inXSDDateTime ?dti2.
18  ?bgj time:inXSDDateTime ?dtj1.
19  ?endj time:inXSDDateTime ?dtj2.
20 FILTER(?dti2=?dtj2 && ?dti1>?dtj1)
21 }
```

C.4 time:intervalMeets et time:intervalMetBy

Code C.4 – Inférence de la relation *intervalMeets* et *intervalMetBy* entre deux intervalles de temps

```
1  PREFIX time: <http://www.w3.org/2006/time#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  INSERT
4  {
5   ?i time:intervalMeets ?j.
6   ?j time:intervalMetBy ?i.
7  }
8  WHERE
9  {
10  ?i rdf:type time:ProperInterval.
11  ?j rdf:type time:ProperInterval.
12  ?i time:hasEnd ?endi.
13  ?j time:hasBeginning ?bgj.
14  ?endi time:inXSDDateTime ?dti.
15  ?bgj time:inXSDDateTime ?dtj.
16 FILTER(?dti=?dtj)
17 }
```

C.5 time:intervalOverlaps et time:intervalOverlappedBy

Code C.5 – Inférence de la relation *intervalOverlaps* et *intervalOverlappedBy* entre deux intervalles de temps

```
1  PREFIX time: <http://www.w3.org/2006/time#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  INSERT
4  {
5   ?i time:intervalOverlaps  ?j.
6   ?j time:intervalOverlappedBy ?i.
7  }
8  WHERE
```

```
 9 {
10  ?i rdf:type time:ProperInterval.
11  ?j rdf:type time:ProperInterval.
12  ?i time:hasEnd ?endi.
13  ?j time:hasBeginning ?bgj.
14  ?j time:hasEnd ?endj.
15  ?endi time:inXSDDateTime ?dti2.
16  ?bgj time:inXSDDateTime ?dtj1.
17  ?endj time:inXSDDateTime ?dtj2.
18 FILTER(?dti2>?dtj1 && ?dti2<?dtj2)
19 }
```

C.6 time:intervalStarts et time:intervalStartedBy

Code C.6 – Inférence de la relation *intervalStarts* et *intervalStartedBy* entre deux intervalles de temps

```
1  PREFIX time: <http://www.w3.org/2006/time#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  INSERT
4  {
5   ?i time:intervalStarts  ?j.
6   ?j time:intervalStartedBy ?i.
7  }
8  WHERE
9  {
10  ?i rdf:type time:ProperInterval.
11  ?j rdf:type time:ProperInterval.
12  ?i time:hasBeginning ?bgi.
13  ?i time:hasEnd ?endi.
14  ?j time:hasBeginning ?bgj.
15  ?j time:hasEnd ?endj.
16  ?bgi time:inXSDDateTime ?dti1.
17  ?endi time:inXSDDateTime ?dti2.
18  ?bgj time:inXSDDateTime ?dtj1.
19  ?endj time:inXSDDateTime ?dtj2.
20 FILTER(?dti1=?dtj1 && ?dti2<?dtj2)
21 }
```

C.7 time:intervalEquals

Code C.7 – Inférence de la relation *intervalEquals* entre deux intervalles de temps

```
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
INSERT
{
 ?i time:intervalEquals ?j.
 ?j time:intervalEquals ?i.
}
WHERE
{
 ?i rdf:type time:ProperInterval.
 ?j rdf:type time:ProperInterval.
 ?i time:hasBeginning ?bgi.
 ?i time:hasEnd ?endi.
 ?j time:hasBeginning ?bgj.
 ?j time:hasEnd ?endj.
 ?bgi time:inXSDDateTime ?dti1.
 ?endi time:inXSDDateTime ?dti2.
 ?bgj time:inXSDDateTime ?dtj1.
 ?endj time:inXSDDateTime ?dtj2.
FILTER(?dti1=?dtj1 && ?dti2=?dtj2)
}
```

C.8 time:inside

Code C.8 – Inférence de la relation *inside* entre un intervalle et un instant temps

```
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
INSERT
{
 ?x time:inside ?a.
}
WHERE
{
 ?x rdf:type time:Instant.
 ?x time:inXSDDateTime ?dt.
 ?a rdf:type time:Interval.
 ?a time:hasBeginning ?be.
 ?a time:hasEnd ?end.
 ?be time:inXSDDateTime ?dt1.
 ?end time:inXSDDateTime ?dt2.
```

```
16 FILTER(?dt>?dt1 && ?dt<?dt2)
17 }
```

Table des figures

1.1 Intégration de deux sources avec un raisonnement spatio-temporel. 3

2.1 Intégration sémantique de données. 10
2.2 Les hétérogénéités survenues dans un système d'information . . . 12
2.3 Système d'intégration de données. 14
2.4 Architecture d'un système médiateur 15
2.5 Processus de construction d'un entrepôt de données 17
2.6 Typologie des ontologies selon leur niveau de conceptualisation. . 21
2.7 Intégration sémantique de données. 25
2.8 Approches à ontologie unique. 26
2.9 Approches à ontologies multiples. 26
2.10 Approches hybrides. 27
2.11 *Mapping* de deux ontologies. 28
2.12 Fusion d'ontologies. 28

3.1 Intégration sémantique de données à l'aide du Web sémantique. . 32
3.2 Pile du Web sémantique. 33
3.3 Un graph RDF. 34
3.4 Un extrait du graph RDF représentant la ville de Paris 35
3.5 Les composantes du langage RDFS. 37
3.6 Un extrait de l'ontologie FOAF 38
3.7 Un extrait de l'ontologie Geonames 40
3.8 Illustration d'une base de connaissances 47
3.9 Archecture du système de D2RQ 49
3.10 Architecture du système de D2R Server 50
3.11 Architecture du système Ontop 51
3.12 Ontologie de GeoSPARQL 55
3.13 Le modèle stRDF 56

4.1 Le temps, l'espace et le spatio-temporel. 61
4.2 Les relations définies dans l'algèbre de l'intervalle. 64
4.3 Les relations de base définies dans l'algèbre de point. 65
4.4 Relations entre les intervalles et les instants. 65
4.5 Ontologie OWL-Time. 67

4.6 SWRL Temporal Ontology. 68
4.7 Exemple de la réification de RDF. 69
4.8 Exemple de la réification des relations N-aire. 70
4.9 Exemple du versionnement d'ontologie. 71
4.10 Exemple de l'approche 4D-fluent. 72
4.11 Modèle «raster» et modèle vectoriel. 75
4.12 Diagramme de classe de l'OpenGIS SFS for SQL 76
4.13 Les classes des géométries introduites dans WKT. 77
4.14 Représentation UML du schéma pour GML. 78
4.15 Les relations topologiques définies par l'OGC. 80
4.16 Les relations de l'algèbre RCC-8 et leurs transitions. 81
4.17 Décomposition d'une région en trois ensemble de points 82
4.18 Matrice de comparaison du modèle des 9-IM. 82
4.19 Représentation des relations topologiques dans la méthode des 9-IM 83
4.20 Représentation de la relation *intersects* 84
4.21 Représentation de la relation *crosses* 84
4.22 Le vocabulaire Basic Geo. 85
4.23 Le modèle GeoRSS-Simple. 86
4.24 Le modèle GeoRSS-GML. 87
4.25 Représentation du GeoRSS au format OWL 87
4.26 L'ontologie owlOGCSpatial 88
4.27 Système hybride pour le raisonnement spatial basé sur le RCC. . . 90
4.28 Traduction des règles SWRL spatiales 91
4.29 Traduction des règles SWRL spatiales 92
4.30 Les composantes d'une ontologie spatio-temporelle. 94
4.31 Le modèle SOWL 95
4.32 Le modèle Continuum 96
4.33 Quatre composantes du *Timeslice* 97

5.1 Extraction de connaissances à partir de données sémantiques. . . 100
5.2 Processus de découverte de connaissances 102
5.3 Un arbre de décision diagnostiquant les maladies. 105
5.4 Un partitionnement appliqué aux données de fleurs d'iris. 107
5.5 Système de fouille de données sémantiques 109
5.6 Un pipeline RDF - apprentissage automatique 111

6.1 La zone atelier Plaine et Val de Sèvre. 118
6.2 Relations Parcelle-Microparcelle dans le modèle des composites spatio-temporels 119
6.3 Modèle des composites spatio-temporels pour l'évolution de parcelles. 120
6.4 Architecture du système de collecte de données. 122
6.5 Les événements territoriaux 123

6.6 Intégration de deux sources de données avec un raisonnement spatio-temporel . . . 124
6.7 Représentation d'un *timeslice*. . . . 125
6.8 Réutilisation de l'ontologie. . . . 126
6.9 Une ontologie pour l'assolement. . . . 129
6.10 Une ontologie pour les observations des oiseaux. . . . 129
6.11 Une ontologie pour les observations des nids. . . . 130
6.12 Une ontologie pour les observations des campagnols. . . . 130
6.13 Exemple d'une généralisation des concepts réutilisés . . . 131
6.14 Exemple d'une généralisation des concepts. . . . 131
6.15 Exemple d'une généralisation de propriété. . . . 132
6.16 Une ontologie pour l'environnement. . . . 132
6.17 Représentation d'une rotation de culture. . . . 134
6.18 Exemple d'un événement d'intégration de deux parcelles. . . . 135
6.19 Exemple d'un croisement des observations et des relevées de l'assolement. . . . 135
6.20 Exemple d'un croisement de différentes observations. . . . 136

7.1 Vue d'ensemble des fonctions du système. . . . 140
7.2 Traduction de données à la demande avec D2R Server. . . . 141
7.3 Matérialisation de données avec D2RQ. . . . 142
7.4 Système d'intégration suivant l'approche médiation . . . 143
7.5 Réutilisation du modèle stRDF . . . 144
7.6 Système d'intégration sémantique de données . . . 145
7.7 Les fonctions accessibles de l'application . . . 148
7.8 L'interface principale de l'application . . . 151
7.9 Visualisation du résultat d'une analyse par un graphe. . . . 152
7.10 Un pipeline de fouille de données sémantiques. . . . 153
7.11 Application de techniques de fouilles dans l'analyse de données . . . 154
7.12 Relation entre l'extraction de données et le Web sémantique . . . 154
7.13 Implémentation des techniques de fouille dans l'analyse de données. 155

8.1 Architecture hybride pour l'intégration de données. . . . 167
8.2 Architecture hybride avec Ontop spatial pour l'intégration de données spatio-temporelles. . . . 168
8.3 Une annotation sur le composante sémantique. . . . 169

A.1 Les étapes de l'intégration sémantique de données . . . 173
A.2 Construction des ontologies avec Protégé . . . 174
A.3 Enregistrement des ontologies . . . 175
A.4 Traduction de données . . . 177
A.5 Chargement de données . . . 178

B.1 Interface principale du prototype. . . . 179

B.2 Représentation statistique de la rotation de cultures. 181
B.3 Visualisation cartographique d'un croisement de l'assolement et des observations. 182
B.4 Représentation statistique d'un croisement des observations et des relevées de l'assolement. 183
B.5 Visualisation d'un croisement entre les observations. 185
B.6 Représentation statistique d'un croisement entre les observations. 185
B.7 Une détection des événements d'intégration entre les parcelles. . . 186

Liste des tableaux

2.1 Évaluation des approches ontologiques 29

6.1 Exemple d'une intégration de l'ontologie OWL-Time. 127
6.2 Relations temporelles entre les entités spatio-temporelles. 127
6.3 Les concepts utilisés dans l'ontologie pour l'environnement 133

7.1 Les fonctions spatiales de stSPARQL 144
7.2 Quelques statistiques de la base de connaissances. 147
7.3 Une proposition de regroupement des cultures 149
7.4 Les règles d'association trouvées pour une succession de cultures sur quatre années 156
7.5 Règles d'association trouvée pour une succession de cultures sur deux années 157
7.6 Une proposition de regroupement des cultures 158
7.7 Les règles d'association trouvées pour une succession de cultures sur deux années 158

Tableau des codes

3.1 Données RDF de la ville de Paris récupérées sur Geonames 36
3.2 Récupération des villes de la France sur Geonames par une requête SPARQL 41
3.3 Calcul du nombre des départements de la France dont la population est plus de 500000 habitants 42
3.4 Exemple du régime d'inférence de RDFS 43
3.5 Inférence de la relation *inside* entre un instant et un intervalle de temps par une règle SWRL 45
3.6 Inférence de la relation *inside* entre un instant et un intervalle de temps par une requête SPARQL Update 45
3.7 Mise en correpondance entre une ontologie et un schéma relationel 50
3.8 Recherche des monuments se situant dans un polygone avec une fonction de filtrage 55
3.9 Recherche des monuments se situant dans un polygone avec une fonction de filtrage 56
4.1 Définition d'un point par GML 79
7.1 Detection des événements d'intégration entre les parcelles 149
7.2 Detection des succession peu probable. 150
7.3 Ajout de nouvelles connaisances à la base. 150
7.4 Croisement entre les observations et l'assolement 151
A.1 Mise en correspondances des données de la table «Culture» de la base Assolement 176
B.1 Rotation de cultures en deux années 180
B.2 Croisement entre l'assolement et les observations 181
B.3 Croisement entre les observations 184
B.4 Detection des événements d'intégration entre les parcelles 185
C.1 Inférence de la relation *intervalBefore* et *intervalAfter* entre deux intervalles de temps 187
C.2 Inférence de la relation *intervalContains* et *intervalDuring* entre deux intervalles de temps 187
C.3 Inférence de la relation *intervalFinishes* et *intervalFinishedBy* entre deux intervalles de temps 188
C.4 Inférence de la relation *intervalMeets* et *intervalMetBy* entre deux intervalles de temps 189

C.5 Inférence de la relation *intervalOverlaps* et *intervalOverlappedBy* entre deux intervalles de temps 189
C.6 Inférence de la relation *intervalStarts* et *intervalStartedBy* entre deux intervalles de temps 190
C.7 Inférence de la relation *intervalEquals* entre deux intervalles de temps 191
C.8 Inférence de la relation *inside* entre un intervalle et un instant temps 191

Bibliographie

[Abedjan et Naumann, 2013] Abedjan, Z. et Naumann, F. (2013). Improving RDF Data Through Association Rule Mining. *Datenbank-Spektrum*, 13(2): 111–120. 111

[Agrawal *et al.*, 1993] Agrawal, R., Imieliński, T. et Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Databases. *SIGMOD Rec.*, 22(2):207–216. 103

[Agrawal et Srikant, 1994] Agrawal, R. et Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. *In Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 104

[Al-Debei *et al.*, 2012] Al-Debei, M. M., al Asswad, M. M., de Cesare, S. et Lycett, M. (2012). Conceptual Modelling and The Quality of Ontologies: Endurantism Vs. Perdurantism. *CoRR*. 63

[Allen, 1983] Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM*, 26:11. 64, 126

[Allen, 1984] Allen, J. F. (1984). Towards a General Theory of Action and Time. *Artif. Intell.*, 23(2):123–154. 64

[Allen et Ferguson, 1994] Allen, J. F. et Ferguson, G. (1994). Actions and Events in Interval Temporal Logic. Rapport technique, Rochester, NY, USA. 64

[Anagnostopoulos *et al.*, 2013] Anagnostopoulos, E., Batsakis, S. et Petrakis, E. G. (2013). CHRONOS: A Reasoning Engine for Qualitative Temporal Information in OWL. *Procedia Computer Science*, 22:70 – 77. 45

[Angles et Gutierrez, 2008] Angles, R. et Gutierrez, C. (2008). The expressive power of SPARQL. *In International Semantic Web Conference*, pages 114–129. Springer. 46

[Antoniou et van Harmelen, 2004] Antoniou, G. et van Harmelen, F. (2004). *Web Ontology Language: OWL*, pages 67–92. Springer Berlin Heidelberg, Berlin, Heidelberg. 39

[Atemezing *et al.*, 2014] Atemezing, G. A., Abadie, N., Troncy, R. et Bucher, B. (2014). Publishing Reference Geodata on the Web: Opportunities

and Challenges for IGN France. *In* Kyzirakos, K., Grütter, R., Kolas, D., Perry, M., Compton, M., Janowicz, K. et Taylor, K., éditeurs : *TC/SSN@ISWC*, volume 1401 de *CEUR Workshop Proceedings*, pages 9–20. CEUR-WS.org. 48

[Auer *et al.*, 2013] Auer, S., Lehmann, J., Ngonga Ngomo, A.-C. et Zaveri, A. (2013). Introduction to linked data and its lifecycle on the web. *In Proceedings of the 9th International Conference on Reasoning Web: Semantic Technologies for Intelligent Data Access*, RW'13, pages 1–90, Berlin, Heidelberg. Springer-Verlag. 70

[Baader *et al.*, 2003] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D. et Patel-Schneider, P. F., éditeurs (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA. 46, 47

[Bachimont, 2000] Bachimont, B. (2000). Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances. *Ingénierie des Connaissances: Evolutions récentes et nouveaux défis*, 1:1–16. 21

[Batsakis et Petrakis, 2011] Batsakis, S. et Petrakis, E. G. M. (2011). SOWL: A Framework for Handling Spatio-temporal Information in OWL 2.0. *In Proceedings of the 5th International Conference on Rule-based Reasoning*, and Applications, RuleML'2011. Programming. 45, 95

[Battle et Kolas, 2011] Battle, R. et Kolas, D. (2011). Geosparql: enabling a geospatial semantic web. *Semantic Web Journal*, 3(4):355–370. 54, 55

[Beddoe *et al.*, 1999] Beddoe, D., Cotton, P., Uleman, R., Johnson, S. et Herring, J. R. (1999). OpenGIS Simple Features Specification For SQL. Rapport technique. 76

[Bellatreche *et al.*, 2004] Bellatreche, L., Pierra, G., Xuan, D. N., Hondjack, D. et Ameur, Y. A. (2004). *An a Priori Approach for Automatic Integration of Heterogeneous and Autonomous Databases*, pages 475–485. Springer Berlin Heidelberg, Berlin, Heidelberg. 26

[Bereta et Koubarakis, 2016] Bereta, K. et Koubarakis, M. (2016). *Ontop of Geospatial Databases*, pages 37–52. Springer International Publishing, Cham. 51, 166

[Bereta *et al.*, 2016] Bereta, K., Xiao, G., Koubarakis, M., Hodrius, M., Bielski, C. et Zeug, G. (2016). Ontop-spatial: Geospatial Data Integration using GeoSPARQL-to-SQL Translation. *In Proceedings of the ISWC 2016 Posters & Demonstrations Track. Co-located with the 15th International Semantic Web Conference (ISWC 2016)*, volume 1690 de *CEUR Electronic Workshop Proceedings*. CEUR-WS.org. 51

[Berners-Lee, 1998] Berners-Lee, T. (1998). The Semantic Web Roadmap. 33

[Berners-lee *et al.*, 2008] Berners-lee, T., Connolly, D., Kagal, L., Scharf, Y. et Hendler, J. (2008). N3Logic: A Logical Framework for the World Wide Web. *Theory Practice Log Program*, 8(3):249–269. 44

[Bizer, 2004] Bizer, C. (2004). D2RQ - treating non-RDF databases as virtual RDF graphs. *In In Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*. 49

[Bizer et Cyganiak, 2006] Bizer, C. et Cyganiak, R. (2006). D2r server-publishing relational databases on the semantic web. *The 5th International Semantic Web*. 50

[Bizer et Schultz, 2010] Bizer, C. et Schultz, A. (2010). The R2R Framework: Publishing and Discovering Mappings on the Web. *In Proceedings of the First International Conference on Consuming Linked Data - Volume 665*, COLD'10, pages 97–108, Aachen, Germany, Germany. CEUR-WS.org. 46

[Bloem et De Vries, 2014] Bloem, P. et De Vries, G. K. (2014). Machine learning on linked data, a position paper. *Linked Data for Knowledge Discovery*, page 69. 110, 111

[Boley *et al.*, 2001] Boley, H., Tabet, S. et Wagner, G. (2001). Design Rationale of RuleML: A Markup Language for Semantic Web Rules. *In Proceedings of the First International Conference on Semantic Web Working*, SWWS'01, pages 381–401, Aachen, Germany, Germany. CEUR-WS.org. 44

[Borst, 1997] Borst, W. (1997). Construction of Engineering Ontologies for Knowledge Sharing and Reuse. 19

[Breiman *et al.*, 1984] Breiman, L., Friedman, J., Stone, C. J. et Olshen, R. A. (1984). *Classification and regression trees*. CRC press. 105

[Bretagnolle *et al.*, 2012] Bretagnolle, V., Aupinel, P., Odoux, J. F. et Henry, M. (2012). Présentation de la zone atelier "plaine et val de sèvre". 117

[Brüggemann *et al.*, 2016] Brüggemann, S., Bereta, K., Xiao, G. et Koubarakis, M. (2016). Ontology-Based Data Access for Maritime Security. *In Extended Semantic Web Conference*, pages 741–757. Springer International Publishing. 143

[Bruijn *et al.*, 2006] Bruijn, J. d., Ehrig, M., Feier, C., Martíns-Recuerda, F., Scharffe, F. et Weiten, M. (2006). *Ontology Mediation, Merging, and Aligning*, pages 95–113. John Wiley & Sons, Ltd. 27

[C. Bessière et Schwer, 1997] C. Bessière, J. Euzenat, R. J. G. L. et Schwer, S. (1997). Raisonnement spatial et temporel. *In Actes 6e journées nationales PRC-GDR intelligence artificielle*, page 77–88. 62

[Calvanese *et al.*, 2016] Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M. et Xiao, G. (2016). Ontop: Answering SPARQL Queries over Relational Databases. *Semantic Web Journal. (to appear)*. 51, 166

[Chen *et al.*, 2004] Chen, S., Liu, B. et Rundensteiner, E. A. (2004). Multiversion-based View Maintenance over Distributed Data Sources. *ACM Trans. Database Syst.*, 29(4):675–709. 18

[Chentout et Vaisman, 2013] Chentout, K. et Vaisman, A. (2013). *Querying Brussels Spatiotemporal Linked Open Data*, pages 378–387. Springer Berlin Heidelberg, Berlin, Heidelberg. 48, 93

[Chi *et al.*, 2004] Chi, Y., Muntz, R. R., Nijssen, S. et Kok, J. N. (2004). Frequent Subtree Mining - An Overview. *Fundam. Inf.*, 66(1-2):161–198. 108

[Choi *et al.*, 2006] Choi, N., Song, I.-Y. et Han, H. (2006). A Survey on Ontology Mapping. *SIGMOD Rec.*, 35(3):34–41. 27

[Christophe, 2005] Christophe, R. (2005). Terminologie et ontologie. 18

[Claramunt et Jiang, 2001] Claramunt, C. et Jiang, B. (2001). An integrated representation of spatial and temporal relationships between evolving regions. *Journal of Geographical Systems*, 3(4):411–428. 94

[Clarke, 1981] Clarke, B. L. (1981). A calculus of individuals based on connection. *Notre Dame Journal of Formal Logic Notre-Dame, Ind.*, 22(3):204–219. 81

[Clementini *et al.*, 1993] Clementini, E., Felice, P. D. et Oosterom, P. v. (1993). A Small Set of Formal Topological Relationships Suitable for End-User Interaction. *In Proceedings of the Third International Symposium on Advances in Spatial Databases*, SSD '93, pages 277–295, London, UK, UK. Springer-Verlag. 83

[Collard, 2003] Collard, M. (2003). *Data Mining, Contributions in Methods and Applications*. Habilitation à diriger des recherches, Université Nice Sophia Antipolis. 101

[Corby *et al.*, 2004] Corby, O., Dieng-Kuntz, R. et Faron-Zucker, C. (2004). Querying the Semantic Web with Corese Search Engine. *In* de Mántaras, R. L. et Saitta, L., éditeurs : *ECAI*, pages 705–709. IOS Press. 46

[Cruz et Xiao, 2005] Cruz, I. F. et Xiao, H. (2005). The Role of Ontologies in Data Integration. *Journal of Engineering Intelligent Systems*, 13:245–252. 10, 23

[Cui *et al.*, 1993] Cui, Z., Cohn, A. G. et Randell, D. A. (1993). Qualitative and Topological Relationships in Spatial Databases. *In Proceedings of the Third International Symposium on Advances in Spatial Databases*, SSD '93, pages 296–315, London, UK, UK. Springer-Verlag. 82

[Cyganiak et Bizer, 2006] Cyganiak, R. et Bizer, C. (2006). D2R Server: A Semantic Web front-end to existing relational databases. *XML Tage*, pages 171–173. 50

[Davidson *et al.*, 1995] Davidson, S., Overton, C. et Buneman, P. (1995). Challenges in Integrating Biological Data Sources. *Journal of Computational Biology*, 2(4):557–572. 16

[Durbha et King, 2005] Durbha, S. S. et King, R. L. (2005). Interoperability in costal zone monitoring systems: Resolving semantic heterogeneities through ontology driven middleware. *In Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS'05.*, volume 6, pages 4236–4239. IEEE. 23

[Egenhofer, 1989] Egenhofer, M. J. (1989). *A formal definition of binary topological relationships*, pages 457–472. Springer Berlin Heidelberg, Berlin, Heidelberg. 80

[Egenhofer et Franzosa, 1991] Egenhofer, M. J. et Franzosa, R. D. (1991). Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161–174. 82

[Eisenberg et Kanza, 2012] Eisenberg, V. et Kanza, Y. (2012). D2RQ/update: updating relational data via virtual RDF. *In WWW*. 50

[Elbyed, 2009] Elbyed, A. (2009). Romie, une approche d'alignement d'ontologies à bases d'instances. 29

[Ermilov *et al.*, 2013] Ermilov, I., Auer, S. et Stadler, C. (2013). CSV2RDF: User-Driven CSV to RDF Mass Conversion Framework. *In Proceedings of the ISEM '13, September 04 - 06 2013, Graz, Austria*. 52

[Fayyad *et al.*, 1996] Fayyad, U. M., Piatetsky-Shapiro, G. et Smyth, P. (1996). Advances in Knowledge Discovery and Data Mining. chapitre From Data Mining to Knowledge Discovery: An Overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA. 101, 102

[Fisher, 1936] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188. 106

[Fleischhacker *et al.*, 2012] Fleischhacker, D., Völker, J. et Stuckenschmidt, H. (2012). *Mining RDF Data for Property Axioms*, pages 718–735. Springer Berlin Heidelberg, Berlin, Heidelberg. 108

[Forgy, 1965] Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–769. 107

[Frasincar *et al.*, 2010] Frasincar, F., Milea, V. et Kaymak, U. (2010). *tOWL: Integrating Time in OWL*, pages 225–246. Springer Berlin Heidelberg, Berlin, Heidelberg. 131

[Frawley *et al.*, 1992] Frawley, W. J., Piatetsky-Shapiro, G. et Matheus, C. J. (1992). Knowledge discovery in databases: An overview. *AI Mag.*, 13(3): 57–70. 101

[García-Solaco *et al.*, 1995] García-Solaco, M., Saltor, F. et Castellanos, M. (1995). Object-oriented Multidatabase Systems. chapitre Semantic Heterogeneity in Multidatabase Systems, pages 129–202. Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK. 12

[Giese *et al.*, 2015] Giese, M., Jiménez-Ruiz, E., Lanti, D., Özçep, O., Rezk, M., Rosati, R., Soylu, A., Vega-Gorgojo, G., Waaler, A. et Xiao, G. (2015). Optique–zooming in on big data access. *IEEE Computer*, 48(3):60–67. 51

[Goh, 1997] Goh, C. H. (1997). Representing and Reason- ing about Semantic Conflicts in Heterogeneous Information Sources. 13

[Grau *et al.*, 2008] Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P. et Sattler, U. (2008). Owl 2: The next step for owl. *Web Semant.*, 6(4):309–322. 41

[Gray *et al.*, 2009] Gray, A. J., Gray, N. et Ounis, I. (2009). Can RDB2RDF Tools Feasibily Expose Large Science Archives for Data Integration ? *In Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC 2009 Heraklion, pages 491–505. Springer-Verlag. 48, 143

[Gruber, 1993] Gruber, T. R. A. (1993). translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220. 19, 20, 22

[Grütter et Bauer-Messmer, 2007] Grütter, R. et Bauer-Messmer, B. (2007). *Towards Spatial Reasoning in the Semantic Web: A Hybrid Knowledge Representation System Architecture*, pages 349–364. Springer Berlin Heidelberg, Berlin, Heidelberg. 89

[Guarino, 1997] Guarino, N. (1997). Understanding, Building and Using Ontologies. *Int. J. Hum.-Comput. Stud.*, 46(2-3):293–310. 21

[Guarino, 1998] Guarino, N. (1998). *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy.* IOS Press, Amsterdam, The Netherlands, The Netherlands, 1st édition. 21, 22

[Guarino et Giaretta, 1995] Guarino, N. et Giaretta, P. (1995). Ontologies and knowledge bases: towards a terminological clarification. *In Towards very large knowledge bases: knowledge building & knowledge sharing 1995*, page 25. 19

[Hacid et Reynaud, 2004] Hacid, M.-S. et Reynaud, C. (2004). L'intégration de sources de données. *Revue Information - Interaction et Intelligence (I3) Une Revue en Sciences du Traitement de l'I*, 4(2). 16, 17

[Hales et Johnson, 2003] Hales, S. D. et Johnson, T. A. (2003). Endurantism, Perdurantism and Special Relativity. *The Philosophical Quarterly*, 53(213): 524–539. 63

[Han *et al.*, 2008] Han, L., Finin, T., Parr, C., Sachs, J. et Joshi, A. (2008). *RDF123: From Spreadsheets to RDF*, pages 451–466. Springer Berlin Heidelberg, Berlin, Heidelberg. 52

[Harbelot, 2015] Harbelot, B. (2015). *Continuum : a spatio-temporal and semantic model for the discovery of dynamic phenomena within geospatial environments*. Thèse de doctorat, Université de Bourgogne. 96, 97

[Hobbs et Pan, 2004] Hobbs, J. R. et Pan, F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing*, 3:66–85. 67

[Hollenbach *et al.*, 2009] Hollenbach, J., Presbrey, J. et Berners-Lee, T. (2009). Using rdf metadata to enable access control on the social semantic web. *In Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK2009)*, volume 514. 167

[Horrocks *et al.*, 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosofand, B. et Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission. 44

[Jarrar *et al.*, 2003] Jarrar, M., Demey, J. et Meersman, R. (2003). *On Using Conceptual Data Modeling for Ontology Engineering*, pages 185–207. Springer Berlin Heidelberg, Berlin, Heidelberg. 63

[JOzefowska *et al.*, 2010] JOzefowska, J., Lawrynowicz, A. et Lukaszewski, T. (2010). The role of semantics in mining frequent patterns from knowledge bases in description logics with rules. *Theory Pract. Log. Program.*, 10(3):251–289. 107

[Karmacharya *et al.*, 2010] Karmacharya, A., Cruz, C., Boochs, F. et Marzani, F. S. (2010). Spatial Rules through Spatial Rule built-ins in SWRL. *Journal of Global Research in Computer Science*, 1(2):1–4. 91

[Katz et Grau, 2005] Katz, Y. et Grau, B. C. (2005). Representing Qualitative Spatial Information in OWL-DL. *In Proceedings of the OWL: Experiences and Directions Workshop. Galway.* 90

[Kiefer *et al.*, 2008] Kiefer, C., Bernstein, A. et Locher, A. (2008). Adding Data Mining Support to SPARQL via Statistical Relational Learning Methods. *In Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC'08, pages 478–492, Berlin, Heidelberg. Springer-Verlag. 108

[Kifer et Boley, 2013] Kifer, M. et Boley, H. (2013). RIF Overview (Second Edition). 44

[Kim et Seo, 1991] Kim, W. et Seo, J. (1991). Classifying Schematic and Data Heterogeneity in Multidatabase Systems. *Computer*, 24(12):12–18. 12

[Klein et Fensel, 2001] Klein, M. et Fensel, D. (2001). Ontology Versioning on the Semantic Web. *In Proceedings of the First International Conference on Semantic Web Working*, SWWS'01, pages 75–91, Aachen, Germany, Germany. CEUR-WS.org. 71

[Knublauch *et al.*, 2011] Knublauch, H., Hendler, J. A. et Idehen, K. (2011). SPIN-overview and motivation. *W3C Member Submission*, 22. 46

[Koubarakis *et al.*, 2016] Koubarakis, M., Kyzirakos, K., Nikolaou, C., Garbis, G., Bereta, K., Dogani, R., Giannakopoulou, S., Smeros, P., Savva, D., Stamoulis, G., Vlachopoulos, G., Manegold, S., Kontoes, C., Herekakis, T., Papoutsis, I. et Michail, D. (2016). Managing Big, Linked, and Open Earth-Observation Data: Using the TELEIOS/LEO software stack. *IEEE Geoscience and Remote Sensing Magazine*, 4(3):23–37. 48

[Krokhin et Jonsson, 2002] Krokhin, A. et Jonsson, P. (2002). Extending the Point Algebra into the Qualitative Algebra. *In Proceedings of the Ninth International Symposium on Temporal Representation and Reasoning (TIME'02)*, TIME '02, pages 28–, Washington, DC, USA. IEEE Computer Society. 65

[Kuramochi et Karypis, 2001] Kuramochi, M. et Karypis, G. (2001). Frequent Subgraph Discovery. *In Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 313–320, Washington, DC, USA. IEEE Computer Society. 108

[Kyzirakos *et al.*, 2012] Kyzirakos, K., Karpathiotakis, M. et M., K. (2012). Strabon: A semantic geospatial dbms. *In The Semantic Web ISWC 2012*, pages 295–311, Berlin. Springer. 53, 56, 57, 143

[Ladkin et Reinefeld, 1992] Ladkin, P. B. et Reinefeld, A. (1992). Effective Solution of Qualitative Interval Constraint Problems. *Artif. Intell.*, 57(1):105–124. 64

[Laïla et Dalila, 2006] Laïla, b. et Dalila, C. (2006). Vers l'interopérabilité des systèmes d'information hétérogènes. 18

[Langegger et Wöß, 2009] Langegger, A. et Wöss, W. (2009). XLWrap — Querying and Integrating Arbitrary Spreadsheets with SPARQL. *In Proceedings of the 8th International Semantic Web Conference*, ISWC '09, pages 359–374, Berlin, Heidelberg. Springer-Verlag. 52

[Langran et Chrisman, 1998] Langran, G. E. et Chrisman, N. R. A. (1998). framework for temporal geographic information. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 25(3):1–14. 119

[Lanti *et al.*, 2015] Lanti, D., Rezk, M., Xiao, G. et Calvanese, D. (2015). The NPD benchmark: reality check for OBDA systems. *In In: Proceedings of the 18th International Conference on Extending Database Technology (EDBT*, pages 617–628. 166

[Lazrak *et al.*, 2010] Lazrak, E., Mari, J.-F. et Benoît, M. (2010). Landscape regularity modelling for environmental challenges in agriculture. *Landscape Ecology*, 25(2):169–183. 117, 149, 158

[Lazrak, 2012] Lazrak, E. G. (2012). *Stochastic data-mining for understanding time and space dynamics of agricultural landscapes. Contribution to numerical*

methods for landscape agronomy. Thèse de doctorat, Université de Lorraine. 157

[Lazrak *et al.*, 2009] Lazrak, E. G., Mari, J.-F. et Marc, B. (2009). Landscape regularity modelling for environmental challenges in agriculture. *Landscape Ecology*, 25(2):169–183. 156

[Lo Bue et Machì, 2015] Lo Bue, A. et Machì, A. (2015). Open Data Integration Using SPARQL and SPIN: A Case Study for the Tourism Domain. *In* Gavanelli, M., Lamma, E. et Riguzzi, F., éditeurs : *AI*IA 2015 Advances in Artificial Intelligence*, volume 9336 de *Lecture Notes in Computer Science*, pages 316–326. Springer International Publishing. 48

[M. A. Khan et Dengel, 2010] M. A. Khan, G. A. G. et Dengel, A. (2010). Two pre-processing operators for improved learning from semanticweb data. *In First RapidMiner Community Meeting And Conference (RCOMM 2010)*. 108, 111

[Macqueen, 1967] Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. *In In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. 107

[Maedche et Staab, 2001] Maedche, A. et Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79. 21

[Malhotra et Melton, 2008] Malhotra, A. et Melton, J. (2008). Progress Report from the RDB2RDF XG. *In Proceedings of the 2007 International Conference on Posters and Demonstrations - Volume 401*, ISWC-PD'08, pages 5–7, Aachen, Germany, Germany. CEUR-WS.org. 48

[Martinez-Cruz *et al.*, 2012] Martinez-Cruz, C., Blanco, I. J. et Vila, M. A. (2012). Ontologies Versus Relational Databases: Are They So Different ? A Comparison. *Artif. Intell. Rev.*, 38(4):271–290. 146

[Mefteh, 2013] Mefteh, W. (2013). *Ontological approach for modeling and reasoning about trajectories : taking into account the thematics, temporals and spatials aspects*. Thèse de doctorat, Université de La Rochelle. 125, 169

[Mefteh *et al.*, 2012] Mefteh, W., Bouju, A. et Malki, J. (2012). Une approche ontologique pour la structuration de données spatio-temporelles de trajectoires. Application à l'étude des déplacements de mammifères marins. *Revue Internationale de Géomatique*, 22(1):55–75. 88, 92, 95, 128

[Meiri, 1996] Meiri, I. (1996). Combining Qualitative and Quantitative Constraints in Temporal Reasoning. *Artif. Intell.*, 87(1-2):343–385. 65

[Michel *et al.*, 2014] Michel, F., Montagnat, J. et Faron-Zucker, C. (2014). A survey of RDB to RDF translation approaches and tools. Research report, I3S. ISRN I3S/RR 2013-04-FR 24 pages. 48

[Miron, 2009] Miron, A. D. (2009). *Semantic association discovery for the Geospatial Semantic Web - the ONTOAST framework*. Thèse de doctorat, Université Joseph-Fourier - Grenoble I. 87

[Mitra *et al.*, 2000] Mitra, P., Wiederhold, G. et Kersten, M. (2000). *A Graph-Oriented Model for Articulation of Ontology Interdependencies*, pages 86–100. Springer Berlin Heidelberg, Berlin, Heidelberg. 26

[Mizoguchi, 2003] Mizoguchi, R. (2003). Part 1: Introduction to Ontological Engineering. *New Gen. Comput.*, 21(4):365–384. 21

[Mizoguchi et Bourdeau, 2000] Mizoguchi, R. et Bourdeau, J. (2000). Using ontological engineering to overcome common AI-ED problems. *Journal of Artificial Intelligence and Education*, 11:107–121. 22

[Muller, 2002] Muller, P. (2002). Topological Spatio–Temporal Reasoning and Representation. *Computational Intelligence*, 18(3):420–450. 94

[Naiman et Ouksel, 1995] Naiman, C. F. et Ouksel, A. M. (1995). A Classification of Semantic Conflicts in Heterogeneous Database Systems. *J. Organ. Comput.*, 5(2):167–193. 12

[Narasimha *et al.*, 2011] Narasimha, V., Kappara, P., Ichise, R. et Vyas, O. (2011). LiDDM: A data mining system for linked data. *In Workshop on Linked Data on the Web. CEUR Workshop Proceedings*, volume 813. 108

[Nebot et Berlanga, 2012] Nebot, V. et Berlanga, R. (2012). Finding Association Rules in Semantic Web Data. *Know.-Based Syst.*, 25(1):51–62. 109

[Neches *et al.*, 1991] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. et Swartout, W. R. (1991). Enabling Technology for Knowledge Sharing. *AI Mag.*, 12(3):36–56. 19

[Nguyen, 2006] Nguyen, D. X. (2006). *Intégration de bases de données hétérogènes par articulation à priori d'ontologies: application aux catalogues de composants industriels*. Theses, Université de Poitiers. 26

[Nigro *et al.*, 2007] Nigro, H. O., Cisaro, S. G. et Xodo, D. H. (2007). *Data Mining With Ontologies: Implementations, Findings and Frameworks*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA. 109

[Noël *et al.*, 2015] Noël, D., Villanova-Oliver, M., Gensel, J. et Le Quéau, P. (2015). Modélisation de trajectoires sémantiques intégrant perspectives multiples et facteurs explicatifs. *In SAGEO Spatial Analysis and GEOmatics*, Hammamet, Tunisia. 95

[Nolle *et al.*,] Nolle, A., Nemirovski, G. et Sicilia, A. An approach for accessing linked open data for data mining purposes. 111

[Noy *et al.*, 2006] Noy, N., Rector, A., Hayes, P. et Welty, C. (2006). Defining n-ary relations on the semantic web. 70

[Noy, 2004] Noy, N. F. (2004). Semantic Integration: A Survey of Ontology-based Approaches. *SIGMOD Rec.*, 33(4):65–70. 27

[O'Connor et Das, 2009] O'Connor, M. et Das, A. (2009). SQWRL: A Query Language for OWL. *In Proceedings of the 6th International Conference on*

OWL: Experiences and Directions - Volume 529, OWLED'09, pages 208–215, Aachen, Germany, Germany. CEUR-WS.org. 45

[O'Connor et Das, 2010] O'Connor, M. J. et Das, A. K. (2010). A method for representing and querying temporal information in OWL. *In International Joint Conference on Biomedical Engineering Systems and Technologies*, pages 97–110. Springer Berlin Heidelberg. 68

[Perry *et al.*, 2011] Perry, M., Jain, P. et Sheth, A. P. (2011). *SPARQL-ST: Extending SPARQL to Support Spatiotemporal Queries*, pages 61–86. Springer US, Boston, MA. 54

[Peuquet, 1994] Peuquet, D. J. (1994). It's about Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems. *Annals of the Association of American Geographers*, 84(3):441–461. 61

[Pinto et Martins, 2000] Pinto, H. S. et Martins, J. (2000). Reusing ontologies. *In AAAI 2000 Spring Symposium on Bringing Knowledge to Business Processes*, volume 2. 125

[Plumejeaud *et al.*, 2011] Plumejeaud, C., Mathian, H., Gensel, J. et Grasland, C. (2011). Spatio-temporal analysis of territorial changes from a multi-scale perspective. *International Journal of Geographical Information Science*, 25(10):1597–1612. 123

[Pokharel *et al.*, 2014] Pokharel, S., Sherif, M. A. et Lehmann, J. (2014). Ontology Based Data Access and Integration for Improving the Effectiveness of Farming in Nepal. *In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02*, WI-IAT '14, pages 319–326, Washington, DC, USA. IEEE Computer Society. 48, 93

[Polleres *et al.*, 2007] Polleres, A., Scharffe, F. et Schindlauer, R. (2007). SPARQL++ for mapping between RDF vocabularies. *In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 878–896. Springer. 46

[Potoniec et Lawrynowicz, 2011] Potoniec, J. et Lawrynowicz, A. (2011). Rmonto: ontological extension to rapidminer. *In 10th International Semantic Web Conference*. 111

[Priyatna *et al.*, 2014] Priyatna, F., Corcho, O. et Sequeda, J. (2014). Formalisation and Experiences of R2RML-based SPARQL to SQL Query Translation Using Morph. *In Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 479–490, New York, NY, USA. ACM. 49, 50

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 106

[Rahm et Bernstein, 2001] Rahm, E. et Bernstein, P. A. (2001). A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal*, 10(4):334–350. 27

[Randell *et al.*, 1992] Randell, D. A., Cui, Z. et Cohn, A. G. (1992). A Spatial Logic based on Regions and Connection. *In Proceedings 3rd International Conference On Knowledge Representation And Reasoning*. 81

[Ren *et al.*, 2009] Ren, W., Singh, S., Singh, M. et Zhu, Y. (2009). State-of-the-art on spatio-temporal information-based video retrieval. *Pattern Recognition*, 42(2):267 – 282. Learning Semantics from Multimedia Content. 94

[Renz, 2002] Renz, J. (2002). *Qualitative Spatial Reasoning with Topological Information*. Springer-Verlag, Berlin, Heidelberg. 82

[Ristoski *et al.*, 2015] Ristoski, P., Bizer, C. et Paulheim, H. (2015). Mining the Web of Linked Data with RapidMiner. *Web Semant.*, 35(P3):142–151. 109, 111

[Ristoski et Paulheim, 2016] Ristoski, P. et Paulheim, H. (2016). Semantic Web in Data Mining and Knowledge Discovery. *Web Semant.*, 36(C):1–22. 108

[Schaller *et al.*, 2012] Schaller, N., Lazrak, E., Martin, P., Mari, J.-F., Aubry, C. et Benoît, M. (2012). Combining farmers' decision rules and landscape stochastic regularities for landscape modelling. *Landscape Ecology*, 27(3):433–446. 117

[Scharffe *et al.*, 2012] Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., Hamdi, F., Bihanic, L., Képéklian, G., Cotton, F., Euzenat, J., Fan, Z., Vandenbussche, P.-Y. et Vatant, B. (2012). Enabling linked data publication with the Datalift platform. *In Proc. AAAI workshop on semantic cities*, page No pagination., Toronto, Canada. scharffe2012a. 49

[Schenk et Staab, 2008] Schenk, S. et Staab, S. (2008). Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web. *In Proceedings of the 17th international conference on World Wide Web*, pages 585–594. ACM. 46

[Seliverstov et Rossetti, 2015] Seliverstov, A. et Rossetti, R. J. F. (2015). An ontological approach to spatio-temporal information modelling in transportation. *In 2015 IEEE First International Smart Cities Conference (ISC2)*, pages 1–6. 48, 52

[Sheth, 1999] Sheth, A. P. (1999). *Changing Focus on Interoperability in Information Systems:From System, Syntax, Structure to Semantics*, pages 5–29. Springer US, Boston, MA. 11, 12

[Sheth et Kashyap, 1993] Sheth, A. P. et Kashyap, V. (1993). So Far (Schematically) Yet So Near (Semantically). *In Proceedings of the IFIP WG 2.6 Database Semantics Conference on Interoperable Database Systems (DS-5)*, pages

283–312, Amsterdam, The Netherlands, The Netherlands. North-Holland Publishing Co. 12

[Sider, 2001] Sider, T. (2001). *Four-Dimensionalism: an Ontology of Persistence and Time*. Oxford University Press. 63

[Skjæveland *et al.*, 2013] Skjæveland, M. G., Lian, E. H. et Horrocks, I. (2013). *Publishing the Norwegian Petroleum Directorate's FactPages as Semantic Web Data*, pages 162–177. Springer Berlin Heidelberg, Berlin, Heidelberg. 48

[Solar et Doucet, 2002] Solar, G. V. et Doucet, A. (2002). Médiation de données: solutions et problèmes ouverts. *Actes des 2ème assises nationales du GDR I*, 3. 11

[Soltan Mohammadi *et al.*, 2017] Soltan Mohammadi, M., Isabelle, M., Thérèse, L. et Christophe, F. (2017). A Semantic Modeling of Moving Objects Data to Detect the Remarkable Behavior. *In AGILE 2017*, Wageningen, Netherlands. Wageningen University, Chair group GIS & Remote Sensing (WUR-GRS). 95

[Staab et Maedche, 2000] Staab, S. et Maedche, A. (2000). Axioms are objects, too - ontology engineering beyond the modeling of concepts and relations. Rapport technique. 20

[Stocker et Sirin, 2009] Stocker, M. et Sirin, E. (2009). PelletSpatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine. *In Proceedings of the 6th International Conference on OWL: Experiences and Directions - Volume 529*, OWLED'09, pages 39–48, Aachen, Germany, Germany. CEUR-WS.org. 90

[Strobl, 2008] Strobl, C. (2008). *Dimensionally Extended Nine-Intersection Model (DE-9IM)*, pages 240–245. Springer US, Boston, MA. 83

[Studer *et al.*, 1998] Studer, R., Benjamins, V. R. et Fensel, D. (1998). Knowledge Engineering: Principles and Methods. *Data Knowl. Eng.*, 25(1-2):161–197. 19

[Tran *et al.*, 2014] Tran, B.-H., Plumejeaud-Perreau, C., Bouju, A. et Bretagnolle, V. (2014). Conception d'un système d'information géographique résilient pour l'environnement. *In Conférence internationale de Géomatique et Analyse Spatiale 2014*, Grenoble, France. Conférence internationale de Géomatique et Analyse Spatiale 2014. 142, 143

[Tufféry, 2012] Tufféry, S. (2012). *Data mining et statistique décisionnelle : l'intelligence des données*. Éd. Technip, Paris. En appendice, choix de documents. 106

[Uschold et Gruninger, 1996] Uschold, M. et Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136. 21, 22, 23

[Valle *et al.*, 2010] Valle, E. D., Qasim, H. M. et Celino, I. (2010). Towards Treating GIS as Virtual RDF Graphs. *In Proceedings of 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services (WebMGS 2010)*. 50

[van Heijst *et al.*, 1997] van Heijst, G., Schreiber, A. T. et Wielinga, B. J. (1997). Using Explicit Ontologies in KBS Development. *Int. J. Hum.-Comput. Stud.*, 46(2-3):183–292. 21

[Vandecasteele, 2012] Vandecasteele, A. (2012). *Expert knowledge ontological modeling for the analyse of abnormal behaviour : application for maritime surveillance*. Theses, Ecole Nationale Supérieure des Mines de Paris. 92

[Vandecasteele et Napoli, 2012] Vandecasteele, A. et Napoli, A. (2012). Spatial ontologies for detecting abnormal maritime behaviour. *In 2012 Oceans - Yeosu*, pages 1–7. 92, 95

[Vilain *et al.*, 1990] Vilain, M., Kautz, H. et van Beek, P. (1990). Readings in Qualitative Reasoning About Physical Systems. chapitre Constraint Propagation Algorithms for Temporal Reasoning: A Revised Report, pages 373–381. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 64

[Vilain, 1982] Vilain, M. B. (1982). A System for Reasoning About Time. *In Proceedings of the Second AAAI Conference on Artificial Intelligence*, AAAI'82, pages 197–201. AAAI Press. 65

[Villata *et al.*, 2011] Villata, S., Delaforge, N., Gandon, F. et Gyrard, A. (2011). A Social Semantic Web Access Control Model. *In SDoW 2011 : 4th international workshop Social Data on the Web (SDoW2011), co-located with ISWC2011*, Bonn, Germany. 167

[Visser et Schuster, 2002] Visser, U. et Schuster, G. (2002). Finding and integration of information-a practical solution for the semantic web. *Ontologies and Semantic Interoperability*, page 74. 23

[Wache *et al.*, 2001] Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. et Hübner, S. (2001). Ontology-Based Integration of Information - A Survey of Existing Approaches. *In IJCAI-01 Workshop: Ontologies and Information*, pages 108–117. 24, 29

[Wannous, 2014] Wannous, R. (2014). *Computational inference of conceptual trajectory model : considering domain temporal and spatial dimensions*. Thèse de doctorat, Université de La Rochelle. 92, 95

[Welty et Fikes, 2006] Welty, C. et Fikes, R. A. (2006). reusable ontology for fluents in owl. *In Proceedings of the conference on formal ontology in information systems*, pages 226–236. p.. IOS Press. 71

[Wiederhold, 1992] Wiederhold, G. (1992). Mediators in the Architecture of Future Information Systems. *Computer*, 25(3):38–49. 14

[Wolter et Zakharyaschev, 2000] Wolter, F. et Zakharyaschev, M. (2000). Spatio-temporal representation and reasoning based on RCC-8. *In In Proceedings of the seventh Conference on Principles of Knowledge Representation and Reasoning, KR2000*, pages 3–14. Morgan Kaufmann. 94

[Yuan, 1999] Yuan, M. (1999). Use of a Three-Domain Repesentation to Enhance GIS Support for Complex Spatiotemporal Queries. *Transactions in GIS*, 3(2):137–159. 61