



Continuum : un modèle spatio-temporel et sémantique pour la découverte de phénomènes dynamiques au sein d'environnements géospatiaux

Benjamin Harbelot

► **To cite this version:**

Benjamin Harbelot. Continuum : un modèle spatio-temporel et sémantique pour la découverte de phénomènes dynamiques au sein d'environnements géospatiaux. Intelligence artificielle [cs.AI]. Université de Bourgogne, 2015. Français. <tel-01274530>

HAL Id: tel-01274530

<https://hal.archives-ouvertes.fr/tel-01274530>

Submitted on 15 Feb 2016

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

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

SPIM

Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques

U N I V E R S I T É D E B O U R G O G N E

Continuum : un modèle
spatio-temporel et sémantique pour
la découverte de phénomènes
dynamiques au sein
d'environnements géospaciaux



BENJAMIN HARBELOT



SPIM

Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques
UNIVERSITÉ DE BOURGOGNE

THÈSE présentée par

BENJAMIN HARBELOT

pour obtenir le

Grade de Docteur de
l'Université de Bourgogne

Spécialité : **Informatique**

**Continuum : un modèle spatio-temporel et
sémantique pour la découverte de phénomènes
dynamiques au sein d'environnements géospatiaux**

Unité de Recherche :

Équipe Checksem, Le2i, UMR CNRS 6306, Université de Bourgogne

Soutenue publiquement le 17 décembre 2015 devant le Jury composé de :

MME ANNE RUAS	Rapporteur	Directrice de recherche, IFSTTAR
M. FRANK BOOCHS	Rapporteur	Pr., I3MAINZ
M. GILLES GESQUIERE	Examineur	Pr., Université Lumière Lyon 2
MME SANDRA BRINGAY	Examineur	MCF, Université de Montpellier
M. CHRISTOPHE CRUZ	Directeur de thèse	MCF HDR, Université de Bourgogne

REMERCIEMENTS

Je souhaite exprimer mes plus sincères remerciements à mon directeur de thèse, le Docteur Christophe Cruz, pour m'avoir fait partager son expérience et ses compétences ainsi que pour tout le temps et l'attention portée à mes travaux. Je souhaite également le remercier pour son soutien et sa confiance. Sa grande disponibilité, tant sur le plan personnel que professionnel, a grandement contribué à la qualité de ces travaux et a permis de faire de cette thèse une expérience enrichissante.

Je voudrais remercier la Direction Générale de l'Armement pour avoir financé les travaux présentés dans cette thèse.

J'adresse un remerciement particulier à Yoan Chabot pour ces nombreuses années d'amitié sans faille. Son soutien régulier et sa bonne humeur quotidienne ont contribué à mon bien-être au sein de l'équipe Checksem.

Je tiens à remercier mes collègues et amis de l'équipe Checksem : Tarcisio Mendes de Farias, Thomas Hassan, David Werner, Rafael Peixoto, Anett Hoppe, Helbert Arenas pour la bonne ambiance de travail et les bons moments passés pendant ces trois années. Je remercie également les membres permanents de l'équipe Checksem : Ana Roxin, Aurélie Bertaux et Ouassila Narsis pour leur bonne humeur.

Je tiens à remercier vivement le Professeur Christophe Nicolle pour m'avoir accueilli au sein de son équipe et dont le savoir et l'expérience m'ont été très favorable.

Je souhaite remercier très sincèrement Guillaume Broux pour son amitié. Ses réflexions et ses conseils m'ont toujours été bénéfiques.

Je voudrais remercier également ma famille et les amis fortement assimilés au réseau familiale pour l'intérêt manifesté à maintes occasions. Trop nombreux pour être tous cités, j'espère qu'ils se reconnaîtront. J'adresse un remerciement particulier à mes parents qui ont toujours cru en moi.

Pour finir, je remercie Charlotte Blouzard pour son attention, son soutien et sa bonne humeur quotidienne.

SOMMAIRE

I	Contexte et Problématiques	1
1	Introduction	3
1.1	Contexte	4
1.1.1	Réalité physique et réalité observable	5
1.1.2	Une compréhension partagée pour la définition des jeu de données	6
1.1.3	Définition du modèle pour la découverte de connaissance : de la donnée à la décision	7
1.2	Limites	12
1.3	Verrous	13
1.4	Contribution	14
1.4.1	L'ontologie : une modélisation flexible, accessible et compréhensible sans perte sémantique	14
1.4.2	L'ontologie comme support de l'information spatio-temporelle	15
1.4.3	Analyse qualitative et causalité	16
1.4.4	Aide à la décision	18
1.5	Plan de la thèse	19
1.5.1	Chapitre 2 : Etat de l'art	19
1.5.2	Chapitre 3 : le modèle Continuum	19
1.5.3	Chapitre 4 : Implémentation et calcul des couches 1 et 2	20
1.5.4	Chapitre 5 : calcul de la couche 3 : les motifs génériques	20
1.5.5	Chapitre 6 : calcul de la couche 4 et découverte des connaissances	20
1.5.6	Chapitre 7 : Expérimentation	20
1.5.7	Chapitre 8 : Conclusion	20
2	Etat de l'art	21
2.1	Modélisation spatio-temporelle	22
2.1.1	Notion de temps	22
2.1.2	Notion d'espace	26
2.1.3	Notions spatio-temporelles	28
2.1.3.1	Primitives de modélisation	29

2.1.3.2	Le changement	29
2.1.3.3	Représentation d'une entité	29
2.1.3.4	Différents types d'entités spatio-temporelles	31
2.2	Étude des approches de modélisation spatio-temporelle	32
2.2.1	Approches basées sur les "snapshots"	32
2.2.2	Approches basées sur les entités	33
2.2.3	Approches basées sur les évènements	35
2.2.4	Approches basées sur les graphes	36
2.2.5	Discussion	37
2.3	Logiques de description et ontologies	39
2.3.1	Base de connaissances	39
2.3.1.1	TBox : terminologie	39
2.3.1.2	ABox : description du monde	40
2.3.2	Définition et usage d'ontologie en informatique	41
2.3.3	Le langage de représentation des ontologies : OWL	42
2.4	L'approche Web Sémantique et l'approche classique	47
2.4.1	Comparatif des caractéristiques intrinsèques des systèmes en monde ouvert et en monde fermé	47
2.4.1.1	Complétude des informations	47
2.4.1.2	Flexibilité du schéma	48
2.4.1.3	Gestion de l'unicité des entités	48
2.4.1.4	Interprétation du monde représenté	49
2.4.1.5	discussion	49
2.4.2	Interprétation des contraintes : de la contrainte d'intégrité à l'axiome logique	50
2.4.2.1	Les contraintes d'intégrité pour la détection d'inconsistance	50
2.4.2.2	L'axiome logique comme support d'inférence	51
2.4.2.3	Discussion	51
2.5	Modélisation spatio-temporelle dans le Web Sémantique	53
2.5.1	Modélisation spatiale dans le Web Sémantique	53
2.5.1.1	GML (Geography Markup Language)	54
2.5.1.2	Basic RDF Geo Vocabulary	54
2.5.1.3	GeoRSS et GeoOWL	54
2.5.1.4	SPARQL-ST, st-RDF	55
2.5.1.5	GeoSPARQL	56

2.5.2	Modélisation du temps dans le Web Sémantique	60
2.5.2.1	OWL-Time	60
2.5.2.2	SWRL temporal ontology	60
2.5.3	Ontologie de Fluent : Modélisation de la dynamique d'un système	61
2.5.3.1	Naissance des ontologies de fluent	62
2.5.3.2	4D-fluents	64
2.5.3.3	TOWL	64
2.6	Discussion	66
2.6.1	Apport pour le domaine de la modélisation spatio-temporelle	67
2.6.2	Apport pour le domaine de la dynamique d'utilisation des terres et des couvertures terrestres	69
II	Contribution	73
3	Continuum : un modèle sémantique spatio-temporel	75
3.1	Définition d'une couche spatio-temporelle	75
3.1.1	Organisation hiérarchique des concepts	76
3.1.2	Représentation du temps : intervalles et instants	76
3.1.3	Les quatre composantes d'une entité	77
3.2	Définition d'une transition spatio-temporelle	78
3.2.1	Expression des contraintes d'intégrité des relations de filiation	80
3.2.1.1	Contraintes d'intégrité de la première couche du modèle : la relation de filiation	81
3.2.1.2	Contraintes d'intégrité de la deuxième couche du modèle : la couche d'identité	82
3.2.1.3	Contraintes d'intégrité de la troisième couche du modèle : les motifs génériques	84
3.2.2	Contraintes d'intégrité sur les primitives du modèle Continuum	89
4	Implémentation et calcul de la 1ère et 2ème couche	93
4.1	Le jeu de données Corine Land Cover	93
4.2	Calcul de la couche 1 et 2	96
4.2.1	La transition spatio-temporelle comme base d'étude des relations de filiation	96
4.2.2	Calcul de la couche 1 : généalogie	98
4.2.3	Élimination des changements négligeables	98
4.2.4	Calcul de la couche 2 : cycle de vie	101

5	Calcul de la couche 3 : les motifs génériques spatiaux	105
5.1	Relations de continuation unaire	106
5.2	Relations de dérivation unaire	108
5.3	Motifs d'évolution N-aire	109
6	Couche 4 : découverte de connaissance	115
6.1	Étude de la profondeur du changement	115
6.2	Les motifs de tendance : du phénomène au processus	123
6.3	Étude des évolutions connexes	128
7	Expérimentation	131
7.1	Intégration du jeu de données Corine Land Cover	131
7.2	Création des transitions spatio-temporelles	133
7.3	Calcul des couches du modèle	133
7.4	Architecture de l'application	136
III	Conclusion	141
8	Conclusion générale	143
8.1	Bilan	143
8.2	Perspectives	144
IV	Annexes	159
A	Liste des publications sur les travaux	161
B	Introduction aux logiques de description	163
B.1	Les familles des logiques de description	163
B.1.1	\mathcal{AL} : la logique de description de base	164
B.1.2	Le constructeur de hiérarchie de rôles \mathcal{H}	165
B.1.3	Le constructeur de conjonction de rôles \mathcal{R}	166
B.1.4	Le constructeur de rôles transitifs \mathcal{R}^+	166
B.1.5	Le constructeur d'énumération d'individus \mathcal{O}	166
B.1.6	Le constructeur d'union \mathcal{U}	166
B.1.7	Le constructeur de quantification existentielle complète \mathcal{E}	167
B.1.8	Le constructeur de fonction \mathcal{F}	167

B.1.9	Le constructeur d'inverse \mathcal{I}	167
B.1.10	Les restrictions de cardinalité \mathcal{N}	168
B.1.11	Les restrictions de cardinalité qualifiée \mathcal{Q}	169
B.1.12	La négation sans restriction \mathcal{C}	169
B.1.13	Discussion	169
B.2	Les raisonnements en logiques de description	172
B.2.1	Raisonnement sur la TBox	172
B.2.2	Raisonnement sur la ABox	173
B.2.3	Complexité de l'inférence	174
C	Vérification de contraintes d'intégrité	177
C.1	Les contraintes de subsomption	177
C.2	Les contraintes domain-range	178
C.3	Les contraintes de participation	179
C.4	Les contraintes de cardinalité	181
C.5	Les contraintes de propriétés	184
C.6	Les contraintes complexes	185
D	Raisonnements d'inférences en OWL	189
D.1	Présentation de OWL 2 DL	189
D.2	Les règles d'inférence de OWL Horst	191
D.2.1	Règles RDFS	191
D.2.2	Règles OWL	192
D.3	Raisonnements sur la TBOX	192
D.3.1	Inférence sur les classes	194
D.3.1.1	Héritage de la disjonction avec le complément d'une super-classe	194
D.3.1.2	Héritage de la contrainte de disjonction	194
D.3.1.3	owl :equivalentClass	194
D.3.2	Inférence sur les propriétés	195
D.3.2.1	Transitivité et relation inverse	195
D.3.2.2	Héritage des propriétés subsumées	195
D.3.2.3	owl :equivalentProperty	196
D.4	Raisonnements sur la ABOX	196
D.4.1	Inférence d'instance de classe	197
D.4.1.1	L'héritage via rdfs :subclassof	197

D.4.1.2	Les contraintes rdfs :domain et rdfs :range	197
D.4.1.3	L'axiome owl :oneOf	198
D.4.1.4	La restriction owl :hasvalue	198
D.4.1.5	La restriction owl :allvaluesfrom	199
D.4.1.6	La restriction owl :someValuesFrom	199
D.4.2	Inférence d'instances de propriétés	200
D.4.2.1	restriction owl :hasvalue	200
D.4.2.2	La transitivité à l'aide de owl :TransitiveProperty	200
D.4.2.3	La symétrie à l'aide de owl :SymmetricProperty	201
D.4.2.4	La propriété inverse à l'aide de owl :inverseOf	201
D.4.2.5	owl :AllDifferent et owl :differentFrom	201
D.4.2.6	Propriétés fonctionnelles	202
D.4.2.7	Propriétés inversement fonctionnelles	202



CONTEXTE ET PROBLÉMATIQUES

INTRODUCTION

Ce chapitre introduit le contexte des travaux, les limites, les verrous scientifiques ainsi que la contribution des travaux présentés. Enfin, une description des différents chapitres de ce rapport conclut cette introduction. Les systèmes d'information géographique offrent une vision simplifiée du monde réel. Par exemple, l'absence de la connaissance qu'un arbre ou un bâtiment existe sur une parcelle de terrain, dans l'absolu, ne permet pas de conclure qu'il n'y a pas d'arbre ou de bâtiment sur cette parcelle. Le manque de connaissance sur l'environnement géographique est une caractéristique récurrente du domaine spatio-temporel. Dès lors, il est nécessaire, d'une part, de définir des outils permettant d'offrir une compréhension partagée de la réalité modélisée à la fois par des humains et des machines. D'autre part, de fournir des capacités d'analyse pour la découverte de connaissances implicites susceptibles d'améliorer la compréhension de la dynamique des territoires et faciliter les décisions pour des experts ou chargés de territoire. Plusieurs modèles ont été proposés dans la littérature pour représenter la dynamique d'entités spatio-temporelles mais ne permettent pas de représenter et prendre en compte le contexte de l'environnement géospatial. Pour répondre à cette problématique majeure, ces travaux envisagent les technologies du Web Sémantique comme support de l'information spatio-temporelle afin de proposer un modèle riche pour l'analyse qualitative de la dynamique des entités.

Sommaire

1.1 Contexte	4
1.1.1 Réalité physique et réalité observable	5
1.1.2 Une compréhension partagée pour la définition des jeu de données	6
1.1.3 Définition du modèle pour la découverte de connaissance : de la donnée à la décision	7
1.2 Limites	12
1.3 Verrous	13
1.4 Contribution	14
1.4.1 L'ontologie : une modélisation flexible, accessible et compréhensible sans perte sémantique	14
1.4.2 L'ontologie comme support de l'information spatio-temporelle	15
1.4.3 Analyse qualitative et causalité	16
1.4.4 Aide à la décision	18
1.5 Plan de la thèse	19
1.5.1 Chapitre 2 : Etat de l'art	19
1.5.2 Chapitre 3 : le modèle Continumm	19
1.5.3 Chapitre 4 : Implémentation et calcul des couches 1 et 2	20

1.5.4	Chapitre 5 : calcul de la couche 3 : les motifs génériques	20
1.5.5	Chapitre 6 : calcul de la couche 4 et découverte des connaissances	20
1.5.6	Chapitre 7 : Expérimentation	20
1.5.7	Chapitre 8 : Conclusion	20

1.1/ CONTEXTE

Les Systèmes d'Informations Géographiques ont été conçus afin de rendre compte de l'état du monde réel. La correspondance entre la réalité et la représentation qui en est faite au sein du SIG est un enjeu majeur dans la conception de ce système.

Cependant, la réalité physique, telle qu'elle existe dans la nature ne peut qu'être observée et approximée par les outils de stockage existants. La représentation de cette réalité est alors, par nature, inexacte et incertaine à cause de l'approximation imposée par les systèmes de représentation. Plusieurs différences découlent de ce décalage entre réalité et représentation :

- Les propriétés physiques du monde réel sont continues tandis que les systèmes d'information sont uniquement capables de donner une approximation discrète de l'espace et du temps. D'autre part, le monde réel évolue de manière continue tandis que le système d'information tente de retracer et d'enregistrer les changements qui s'opèrent.
- La réalité est soumise à la simplification et à l'approximation aussi bien dans l'espace que dans le temps créant ainsi un décalage entre le monde réel et sa représentation.

La modélisation spatio-temporelle d'un environnement géographique tente de donner une représentation du monde réel de sorte que celle-ci soit compréhensible et utilisable par un public expert ou non-expert. En somme, la modélisation se doit de mettre en évidence les entités essentielles composant l'environnement géographique, prendre en compte les interactions et les relations entre ces différentes entités, enfin la connaissance établie à partir de ces données doit permettre d'inférer de nouvelles connaissances et d'aider à la prise de décision. Dans [Frank, 2003], l'auteur définit une hiérarchie ontologique en 5 étapes pour décrire l'environnement géographique.

- Etape 1 : **La réalité physique**. La première étape décrit la réalité physique. Par conséquent, il existe une infinité de points définis dans le temps et l'espace. Chacun de ces points possède des propriétés déterminées. L'espace et le temps sont représentés comme des dimensions fondamentales de la réalité.
- Etape 2 : **La réalité observable** : La réalité observable est une perception de la réalité physique. Elle permet d'observer les propriétés d'un point dans l'espace à un instant précis. Du décalage entre la réalité et sa perception résulte l'estimation. Par conséquent, les observations sont toutes incomplètes et imprécises.
- Etape 3 : **Les objets de l'environnement observés** : à partir de la réalité observée, des objets sont définis selon des propriétés uniformes dans l'espace

et dans le temps. Par exemple, un ensemble d'arbres se résume à une figure polygonale qui évolue dans le temps en fonction de la taille de la forêt. En outre, des attributs descriptifs peuvent être associés à cette géométrie.

- Etape 4 : **La réalité sociale** : La plupart des informations contenues dans une base de données spatio-temporelle ne sont pas des propriétés physiques du monde réel. La réalité physique se compose d'un très grand nombre de choses qui interagissent entre elles. Cependant, bien souvent, la réalité physique n'est pas représentée comme telle au sein des systèmes d'information géographique, mais plutôt interprétée dans une réalité sociale. Bien que cette réalité sociale nous semble aussi réelle que la réalité physique, elle est en fait totalement artificielle, car elle n'existe qu'au travers d'un contexte défini par des interactions entre des entités nommées ainsi que des faits et des règles sociales établis.
- Etape 5 : **Les agents cognitifs** : Ces agents prennent en compte à la fois le monde physique et la réalité sociale qui compose le monde dans lequel ils évoluent. L'objectif de ces agents est alors d'inférer de nouvelles connaissances à partir de celles déjà existantes, et d'utiliser ces nouvelles connaissances dans un processus d'aide à la décision.

Les travaux proposés dans cette thèse portent sur la modélisation des objets de l'environnement observés et de la réalité sociale inscrits dans une dynamique. En tentant d'offrir un support riche de représentation des connaissances, le modèle proposé permet **une représentation détaillée du contexte de l'environnement géospatial** et sera présenté au chapitre 3. En outre, le modèle a pour objectif de contribuer au développement d'agents cognitifs en utilisant les informations contextuelles définies aux étapes précédentes. Un tel modèle nécessite de **fournir des outils capables d'extraire des motifs génériques** et d'**offrir une interprétation de ces motifs pour aboutir à la découverte de connaissance** au sein du jeu de données. Cet aspect sera traité au chapitre 6.

1.1.1/ RÉALITÉ PHYSIQUE ET RÉALITÉ OBSERVABLE

Le point de départ de toute application SIG est le monde réel dont la définition peut être associée à celle de la réalité physique. La réalité physique ne peut être illustrée en tant que telle, car toute illustration découle d'une perception et altère la nature de cette réalité. La perception du monde réel par un capteur ou un humain donne lieu à la réalité observable. La figure 1.1 illustre un exemple de cette réalité observable.

Bien que cette figure semble représenter le monde réel, elle ne l'est pas. Cette figure est un cliché qui se résume en un ensemble de pixel qui discrétise obligatoirement le monde réel. Dès lors, cette image est une approximation de la réalité physique réalisée à l'aide d'un capteur. Les perceptions humaines souffrent des mêmes contraintes du simple fait que chaque être humain est unique.

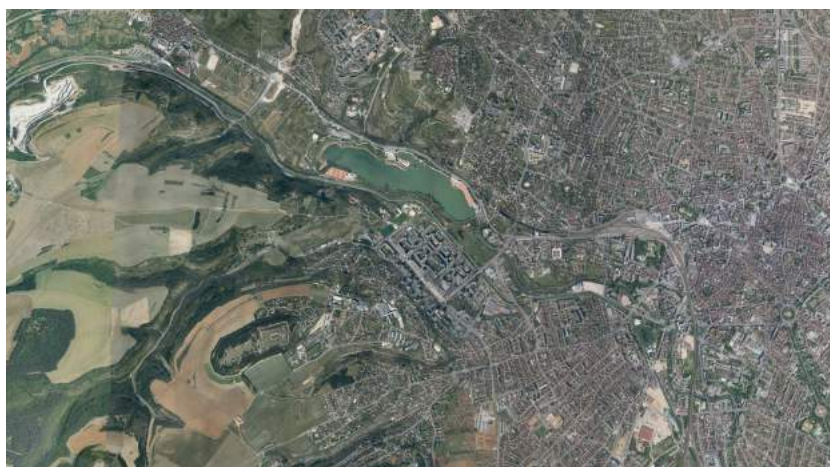


FIGURE 1.1 – La réalité observable

1.1.2/ UNE COMPRÉHENSION PARTAGÉE POUR LA DÉFINITION DES JEU DE DONNÉES

Cependant, bien que les perceptions de tout à chacun soit différentes, il est possible de partager et de s'accorder sur l'interprétation de ces perceptions. Ainsi, bien qu'il existe un spectre infini de longueurs d'ondes pour constituer la lumière visible au travers de différentes couleurs, l'homme a depuis des siècles réussi à définir un vocabulaire commun afin de partager des informations sur les couleurs. Ainsi, chaque être humain possédant la maîtrise de ce vocabulaire est capable de se faire une représentation de ce qu'est le *rouge*, le *vert*, le *bleu*, etc. Toutefois, il est intéressant de noter que plus le nombre de termes pour représenter les couleurs est élevé, plus il existe des risques de confusion d'une personne à l'autre. Par exemple, les termes *blanc cassé*, *blanc de zinc*, *blanc de lait* possède un fort potentiel de confusion.

Le même principe s'applique pour la conception de jeu de données de façon plus élaborée, car il est nécessaire de s'accorder sur plusieurs caractéristiques de la réalité observable. Un jeu de données est un groupement de données par sujet, par thème ou par type. Ainsi, la conception d'un jeu de données, requiert la sélection des caractéristiques prépondérantes de la réalité observable pour un domaine d'étude particulier. Les objets de la réalité observable sont alors approximés et ramenés à des figures géométriques accompagnées éventuellement d'attributs descriptifs.

La figure 1.2 illustre les objets de l'environnement observé à partir de la réalité observable décrite par la figure 1.1. De nombreux objets non-pertinents de la réalité observable ne sont plus représentés, les routes deviennent des lignes, un ensemble d'arbre devient un polygone vert, un lac devient un polygone bleu ou encore un ensemble de bâtiment devient un polygone gris. Ainsi, les nombreuses nuances sont ramenées à un nombre minimal et explicite de couleur et les objets complexes sont résumés par de simples géométries. Cette représentation permet de discriminer et mettre en avant chaque objet de manière la plus compréhensible pour l'homme. En parallèle, ce travail de simplification du monde réel permet d'établir des jeux de données utilisables par la machine pour le stockage et l'analyse.

Dans cet exemple, seuls les objets existants physiquement dans le monde réel sont



FIGURE 1.2 – Les objets de l’environnement observé ainsi que leur réalité sociale

représentés. Cependant, l’exploitation de ces données spatiales dépourvue de données contextuelles ne permet pas d’obtenir de résultats pertinents. Par conséquent, il est nécessaire d’ajouter des attributs ainsi que des géométries afin de modéliser la réalité sociale de l’environnement géographique. Sur la figure 1.2, cette réalité sociale est représentée par : les noms de route, les noms de ville et la géométrie définissant les frontières de la ville de Dijon. En somme, cette réalité sociale définie par l’auteur dans [Frank, 2003] correspond à ce que nous qualifierons de ”contexte” de l’application.

Dans la littérature de l’ingénierie des connaissances, les ontologies se sont imposées comme un moyen efficace de définir un vocabulaire commun pour les chercheurs qui ont besoin de partager l’information dans un domaine. Une ontologie contient des définitions lisibles en machine des concepts de base de ce domaine et de leurs relations. Dans ces travaux, nous prenons le parti d’utiliser l’ontologie comme composante centrale du modèle de représentation d’entité spatio-temporelles proposé dans le cadre de cette thèse.

1.1.3/ DÉFINITION DU MODÈLE POUR LA DÉCOUVERTE DE CONNAISSANCE : DE LA DONNÉE À LA DÉCISION

La définition des objets de l’environnement observé ainsi que la réalité sociale sont guidés par une perception et une compréhension partagée de la réalité observée de sorte que chaque polygone, ligne ou couleur puisse être associée à la sémantique d’un objet ou une entité de cette réalité, et ce, quelque soit l’utilisateur du jeu de données. A ce stade, le jeu de données n’est qu’une structure permettant d’accueillir les données provenant des capteurs. Un enjeu majeur des systèmes de gestion de données consiste à rendre la donnée utile pour des experts ou de simples consommateurs afin d’aider à prendre une décision sur une action à mener.

Cependant, la donnée brute en tant que telle, n’est pas compréhensible par l’humain et doit être contextualisée. Pour cela, le modèle DIKW (Data-Information-Knowledge-Wisdom) [Rowley, 2007], [Cleveland, 1982], [Ackoff, 1989] a fait son apparition au sein

de la littérature de l'information et de la connaissance. Le modèle DIKW consiste généralement en une hiérarchie visant à donner une définition des données, de l'information, de la connaissance et de la sagesse. Ainsi, il décrit l'articulation entre ces différentes strates, c'est-à-dire décrire les processus de transformation pour passer de la donnée à l'information, de l'information à la connaissance et de la connaissance à la sagesse.

Cette pyramide est connue sous plusieurs noms parmi lesquels nous citerons "hiérarchie de la connaissance", "hiérarchie de l'information", "hiérarchie de la compréhension", "pyramide de la connaissance". Bien que dans la littérature Ackoff [Ackoff, 1989] soit généralement cité comme étant à l'origine de cette hiérarchie, d'autres travaux antérieurs comme ceux de Cleveland en 1982 [Cleveland, 1982] définissaient d'ores et déjà les bases du domaine.

Par la suite, de nombreux travaux suivirent proposant chacun des évolutions au niveau de la hiérarchie, des définitions des termes "donnée", "information", "connaissance" et "sagesse", ou encore au niveau des processus de transformation. Si la majorité des travaux s'accorde à définir l'information en fonction des données, la connaissance en fonction des informations, et la sagesse en fonction des connaissances, il n'existe, en revanche, aucun consensus réel pour décrire les processus de transformations. D'autre part, bien que les couches "données-information-connaissance-sagesse" soient présentes dans les modèles proposés de manière quasi constante, d'autres couches sont régulièrement définies en sus. Enfin, les définitions de chacune de ces couches varient fréquemment au sein des travaux proposés par la communauté. En somme, le modèle DIKW fait l'objet d'une littérature abondante, pour autant, il semble n'y avoir qu'un consensus faible quant à sa définition [Rowley, 2007].

Dans le cadre de cette thèse, nous utiliserons le modèle DIKW pour mettre en évidence l'apport de l'ontologie par rapport aux bases de données traditionnelles et pour expliciter le processus depuis la création des données jusqu'à leur exploitation dans l'aide à la décision. Les systèmes d'information traditionnels de la modélisation spatio-temporelle que sont les bases de données relationnelles sont limités au stockage et au traitement de données et d'informations. Les ontologies, quant à elles, repousse cette limitation pour permettre d'atteindre le stade de la gestion des connaissances offrant ainsi un support intéressant pour l'aide à la décision.

Face à l'hétérogénéité des propositions et des définitions du modèle DIKW de la littérature, nous sélectionnerons les différentes couches en s'appuyant sur les travaux classiques et fixerons nos propres définitions de chacune de ces couches afin de lever les ambiguïtés tout en veillant à refléter les définitions habituelles de la littérature :

- **Les données** : la donnée est un élément brut et ne possède aucune signification. Elle existe sous différentes formes utilisables ou non et peut être quantitative ou bien qualitative. Au stade de la donnée, le contenu est dit "non-structuré".
- **L'information** : A ce stade, la donnée est mise en lien avec d'autres données permettant de leur donner un sens et leur conférer le statut d'"information". L'information peut être utile, mais n'a pas vocation à l'être. Typiquement, les bases de données relationnelles établissent des informations à partir des données stockées permettant de répondre à des questions telles que "qui", "quoi", "où" et "quand". Au stade de l'information, le contenu est dit "structuré".
- **La connaissance** : la connaissance est le résultat de l'appropriation de l'infor-

mation. Contrairement à l'information, la connaissance a pour vocation d'être utile. Pour cela, la connaissance se base sur les données et les informations pour tenter d'en extraire un motif, c'est-à-dire répondre à la question : "comment". Le contenu est alors dit "sémantique". La connaissance est en quelque sorte une mémorisation de l'information visant à en extraire une règle. En ce sens, la connaissance possède généralement des capacités de prédiction à propos de "ce qui est décrit" ou "ce qu'il va se passer ensuite".

- La sagesse** : Cette étape est parfois répertoriée sous le terme "savoir". La sagesse prend en compte des codes spécifiques de l'humain tels que la morale, l'éthique, etc. La sagesse tente de comprendre ce qui n'a pu l'être au travers des étapes précédentes. Dès lors, elle se rapproche de la réflexion faisant appel à des processus tels que le discernement ou encore le jugement entre ce qui est vrai et ce qui est faux ou encore ce qui est bien, ce qui est mal. Par conséquent, contrairement aux étapes précédentes, il ne peut y avoir de réponses tranchées aux questions. En définitive, la sagesse tente de mettre en œuvre l'action la plus appropriée en tenant compte de "ce qui est connue" (connaissance) ainsi que de "ce qui est le mieux" en se basant sur des considérations éthiques et sociales.

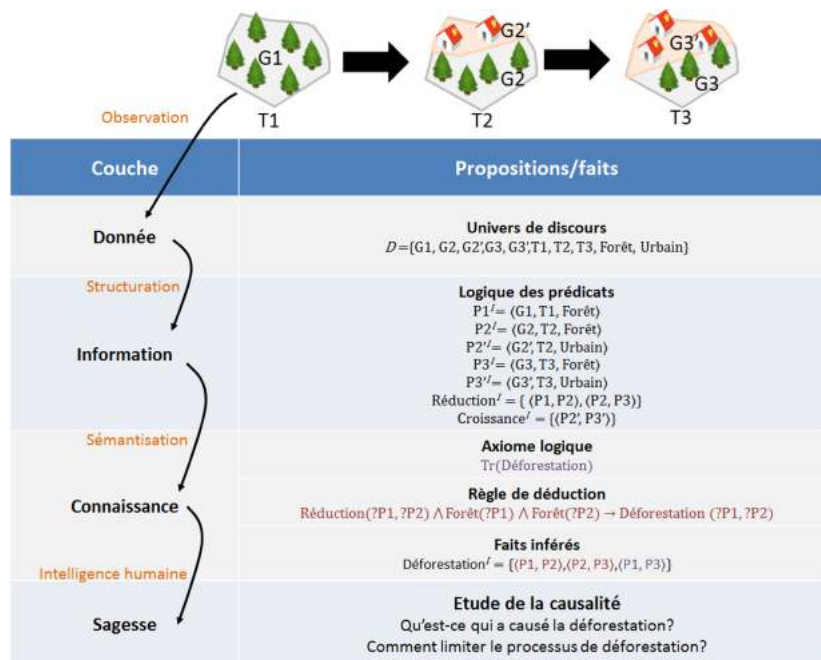


FIGURE 1.3 – Exemple d'évolution spatio-temporelle décrite au travers des différentes couches de la hiérarchie de la compréhension

Chacune des couches abordées précédemment est illustrée par un exemple d'évolution spatio-temporelle au sein de la figure 1.3

La couche "donnée" établit la liste des éléments bruts appelée univers de discours et permettant de représenter la situation à partir de l'observation de celle-ci. L'univers de discours se définit comme un modèle tel que $\langle D, .^I \rangle$ avec D un ensemble non vide et $.^I$ une interprétation. Dans l'exemple, les éléments bruts sont les géométries $G1, G2, G2', G3$

et $G3'$ les estampilles temporelles $T1$, $T2$ et $T3$, et les types *Forêt* et *Urbain*.

Dans la couche "information", les données sont structurées. Les éléments sont mis en corrélation pour tenter de leur donner un sens. Par exemple, la géométrie $G1$, l'estampille $T1$ et le type *forêt* sont mis en corrélation afin de définir l'élément $P1$. Le même principe est appliqué pour les autres parcelles. Enfin, plusieurs parcelles sont également connectées par les relations *Réduction* ou *Croissance*.

La couche "connaissance" permet de sémantiser l'information à l'aide d'axiomes logiques et de règles de déduction capable d'inférer des faits à partir de ceux déjà existants. L'axiome logique donné dans l'exemple définit la relation *Déforestation* comme transitive permettant d'inférer que s'il existe une relation de déforestation entre une parcelle $P1$ et une parcelle $P2$ et entre une parcelle $P2$ et une parcelle $P3$ alors il existe également une relation de déforestation entre $P1$ et $P3$. Il existe également de nombreux autres axiomes logiques qui seront abordés à la section 2.3.3 et pour lesquels plusieurs exemples d'inférences sont disponibles en annexe D. La couche connaissance permet également d'établir des règles de déduction. L'exemple donné définit que toute relation de *réduction* intervenant sur des parcelles de type *Forêt* permet d'établir une relation de déforestation entre ces deux parcelles. Pour cet exemple, la règle s'appuie sur un motif de réduction, dans la suite de ces travaux, nous définissons et formalisons un ensemble de motifs génériques plus complexe dans le chapitre 3 à la section 3.2. Ces motifs plus complexes permettent par exemple de mettre en corrélation la croissance du territoire urbain en parallèle de la diminution des espaces forestiers. L'étude des phénomènes et de leurs causalités sur la base de ces motifs est abordée au chapitre 6.

La couche "sagesse" est un jugement humain qui utilise les connaissances pour analyser la situation, comprendre les causalités et tirer des conclusions sur ce qui est bien ou mal et ce qu'il faut faire ensuite. Par exemple, il peut s'agir de reconnaître que la déforestation est un phénomène néfaste et d'observer que celle-ci est fortement liée à l'urbanisation du territoire. Dès lors, il apparaît nécessaire de mettre en place des moyens afin de réguler le phénomène d'urbanisation et limiter, par extension, le phénomène de déforestation.

La figure 1.4 met en avant les outils classiques permettant de définir ou interroger des systèmes d'informations. Les bases de données traditionnelles sont limitées à la représentation d'information, c'est pourquoi il apparaît pertinent d'établir un comparatif avec les technologies du Web Sémantique réputées pour permettre la gestion et l'analyse de connaissance. Il est intéressant de noter que les technologies du Web Sémantique sont agencés en plusieurs couches de langages allant de la simple identification des objets à la définition détaillée du contexte et de la sémantique de ces objets grâce à la séparation du schéma et des données. Cette répartition en couche est connue sous le nom de "pile du Web Sémantique" dont les couches seront explicitées dans la section 2.3.3. Les bases de données classiques, quant-à-elles, ne possèdent qu'une seule syntaxe dont l'expressivité est limitée pour définir la structure. D'autres différences majeures distinguent les technologies du Web Sémantique des approches classiques et seront abordées à la section 2.4.

Outre le stockage des données, les systèmes d'informations sont également évalués en fonction de leurs capacités de traitement et d'analyse. Dans la littérature des bases de données relationnelles, c'est le langage SQL qui fait autorité. Plusieurs dérivés ou extensions ont été proposées pour satisfaire aux besoins des modèles objets, objets-relationnel ou encore de domaine particulier tels que l'analyse spatiale. Dans le cas du Web Sémantique, c'est le langage SPARQL qui domine. SQL et SPARQL ont tous deux

Couche	Analyse de la dynamique		Syntaxe de représentation		Modèles de la littérature spatio-temporelle
	BD	WS	BD	WS	
Donnée				URI	Modèle snapshot ¹ Modèle TMS ² Modèle STC ³ Modèle ST-Object ⁴
Information	SQL	SPARQL	SQL	Graphe RDF	Modèle identité ⁵ Modèle topologie ⁶ Modèle ESTDM ⁷ Modèle composite ⁸ Modèle Topo-Event ⁹
Connaissance		SPARQL + règle de déduction + Axiome logique		Ontologie RDFS OWL + RIF SWRL	Modèle Continuum ¹⁰ • Entité spatio-temporelles • Relation spatio-temporelles

¹ : [Armstrong, 1988]

² : [Beller et al., 1991]

³ : [Langran et al., 1988]

⁴ : [Worboys, 1994]

⁵ : [Hornsby et al., 2000]

⁶ : [Egenhofer et al., 1992]

⁷ : [Peuquet et al., 1995]

⁸ : [Claramunt et al., 1996]

⁹ : [Jiang et al., 2009]

¹⁰ : [Harbelot et al., 2014b], [Harbelot et al., 2014a], [Harbelot et al., 2013c]

FIGURE 1.4 – Comparatif des outils de définition et d'analyse des bases de données (BD) classiques et du Web Sémantique(WS)

vocations à fournir des informations en mettant en corrélation les données stockées dans le système d'information. Toutefois, il est intéressant de noter que SPARQL est conçu pour interroger des données RDF, c'est-à-dire un modèle sous forme de graphe pour lequel les relations entre les données sont explicitement décrites et accessibles directement. En ce sens, SPARQL permet d'établir des requêtes s'accordant plus facilement avec la représentation mentale de l'utilisateur à propos du domaine d'étude envisagé. En outre, le Web Sémantique se distingue des bases de données relationnelles grâce aux outils de raisonnements et de déduction permettant d'inférer des connaissances explicites à partir de connaissances implicites.

La grande majorité des modèles spatio-temporels fournis pour la gestion d'entités géographiques spatio-temporelles sont implémentés dans des bases de données de type relationnel, relationnel étendu, objet, objet-relationnel et seront étudiés dans la section 2.2. Tous les modèles étudiés utilisent des supports de stockage dont l'expressivité est limitée. Toutefois, les modèles les plus récents et les plus aboutis ont opté pour une modélisation sous forme de graphe révélant la nécessité d'étudier les relations qu'entretiennent les entités au sein de l'environnement géospatial dans le cadre de la modélisation spatio-temporelle. Les limites de ces approches seront abordées dans la section suivante.

1.2/ LIMITES

La dynamique d'un environnement se caractérise par une succession de deux éléments : les états et les transitions. Un état fait référence à une représentation d'une entité à un instant T tandis qu'une transition décrit le processus induisant le passage d'un état vers un autre état. Dans la littérature des ontologies, les états peuvent être assimilés à des objets dits *endurants* (aussi connus sous le nom de *continuant*) et les transitions peuvent être assimilées à des objets dits *perdurants* (aussi connus sous le nom de *occurrent*). Les *endurants* sont définis comme des entités pouvant être perçues comme des concepts complets quelque soit l'instant de temps. Par exemple, une zone urbaine, une voiture, un humain sont des *endurants*. Les *perdurants* sont des entités pour lesquelles seule une partie existe quelque soit l'instant de temps d'observation. Les *perdurants* sont souvent assimilés à des processus tels que l'expansion urbaine, le fait de rouler ou courir. Dès lors, il est intéressant de noter que ce qui est décrit lors de la définition d'un objet *perdurant* est toujours décrit en référence à un *endurant*. Et, que chaque *perdurant* est similaire ou différent d'un autre en fonction du *continuant* concerné. Par exemple, l'expansion urbaine et l'expansion forestière sont des *perdurants* différents car, ils font références à des *continuant*s différents que sont les zones urbaines et les espaces forestiers.

Ainsi, un environnement géospatial peut être vu comme un ensemble de *continuant*s représentés à plusieurs instants ou intervalles de temps et inscrits dans une dynamique par un ensemble de *perdurants*. Dans le cadre de l'étude du changement des couvertures des sols, les *continuant*s sont assimilés à des parcelles tandis que les *perdurants* sont assimilés à des processus ou des phénomènes. Ces derniers sont d'un intérêt capital pour permettre de comprendre la dynamique et les causalités d'évolution de parcelles. Dès lors, plusieurs modèles ont été proposés dans la littérature pour tenter de représenter des entités spatio-temporelles. Les premiers modèles apparût furent les modèles par "snapshots" permettant de représenter un territoire à plusieurs instants de temps. Toutefois, ce type de modèle **ne permet pas de capturer la temporalité des attributs et les propriétés des entités**. Pour pallier cette faiblesse, des modèles basés sur les entités ont vu le jour et permettent de représenter les entités individuellement au sein de l'environnement géospatial. Cependant, ce type de modèle **ne permet pas de lier les différents états au cours du temps ce qui limite fortement la capacité du modèle à mettre en avant les causalités de l'évolution**. Les modèles basés sur les événements sont alors apparus. Ces modèles reposent sur le principe de stocker uniquement les changements. Malgré cela, il n'existe **pas de relations explicites stockées au sein du modèle et le suivi de l'évolution reste une tâche difficile**. D'autres approches ont été proposée pour étudier la dynamique d'environnement au travers des relations topologiques ou encore au travers de l'identité. Bien que ces modèles furent précurseurs dans la prise en compte des relations au sein d'un modèle, **l'étude de la dynamique n'étaient pas explicitement représentée**. Les modèles basés sur les graphes ont alors été proposé pour permettre la détection de motifs spatiaux d'évolution et de processus.

Les approches graphes semblent s'être imposées comme des supports adaptés à la modélisation spatio-temporelle et permettent de représenter la dynamique d'entités spatio-temporelles. Cependant, les motifs et processus permettant de décrire la dynamique **ne s'intéressent qu'à la dynamique spatiale et ne peuvent représenter explicitement des connaissances liées aux entités**. C'est pourquoi, il est nécessaire de fournir des outils de raisonnement afin de permettre de donner une interprétation des motifs spatiaux plus proche d'un phénomène du monde réel en renforçant les capacités

sémantiques du modèle. Les SIG de nouvelle génération requièrent une modélisation et une analyse toujours plus proche de la réalité afin de rapprocher l'humain et la machine. En effet, afin de permettre d'assister l'humain dans sa gestion de l'information, il est intéressant de permettre à la machine d'accéder à la compréhension de l'humain sur un domaine particulier. Dès lors, le **manque d'expressivité des supports actuels** est une forte limitation aux futures défis de la modélisation spatio-temporelle. Dans ce contexte, il est fondamental d'envisager le développement de systèmes capables de prendre en compte des structures sémantiques de plus haut niveau ainsi que de fournir des capacités d'analyses qualitatives pour l'étude de dynamiques géospatiales.

Le Web Sémantique et les technologies associées proposent des outils puissants permettant la représentation des connaissances ainsi que des mécanismes de raisonnement. En outre, les technologies du Web Sémantique dissocient le schéma et les données du système d'information offrant ainsi une structure plus flexible et plus facile à appréhender pour un humain. Enfin, ces outils sont soumis à l'hypothèse du monde ouvert. L'hypothèse du monde ouvert s'oppose à l'hypothèse du monde fermé, valable dans les bases de données traditionnelles, en autorisant la représentation de connaissances incomplètes. Sous l'hypothèse du monde fermé, toute connaissance non explicitement déduite est considérée comme fautive, alors que dans le monde ouvert ce qui est omis reste inconnu. Le manque de connaissance est une caractéristique récurrente du domaine spatio-temporel, c'est pourquoi l'hypothèse du monde ouvert semble la plus adaptée. Toutefois, la représentation de données incomplètes peut également nuire à l'exploitation des données. En effet, l'hypothèse du monde ouvert n'impose pas que toutes les propriétés d'une instance d'un concept soient représentées. Par exemple, dans une application spatio-temporelle, les parcelles d'un territoire possèdent un attribut ou une propriété permettant de stocker et associer une géométrie. Dans le cadre du monde ouvert, il est possible de représenter des parcelles sans géométries. Dès lors, toute analyse spatiale réalisée sur les données fournira des résultats partiellement exploitables. Ainsi, bien que le monde ouvert montre des aptitudes évidentes pour la découverte de connaissances implicites, il est, en revanche, nécessaire **d'établir des limites sur l'incomplétude des données pour définir les nécessités techniques du modèle afin de garantir la consistance du système.**

Les technologies du Web Sémantique ne proposent **pas de standards pour la modélisation spatio-temporelle**. Cependant, des modèles de représentation de données spatiales, d'une part, et temporelles, d'autre part, ont été proposés. Si des modèles tels que GeoSPARQL se sont imposés dans la représentation de données spatiales, la littérature des modèles temporels **souffrent de redondance et ne permettent pas de décrire les transitions entre les différents états**. D'autres approches de la communauté ont proposé des modèles pour la gestion et l'interrogation de données spatio-temporelles. Cependant, ces approches se basent sur **l'introduction de nouvelles syntaxes pour lesquelles ils n'existent pas de supports implémentés.**

Tous ces points seront discutés dans le chapitre 2 : Etat de l'art.

1.3/ VERROUS

La section précédente présente les limites des supports actuels de modélisation spatio-temporelle. La principale limite des modèles proposés dans la littérature concerne le

manque de capacité à représenter explicitement la connaissance liée à l'environnement géospatial ainsi que sa dynamique. Les technologies du Web Sémantique, et plus particulièrement leur application au sein des ontologies, comblent cette limitation. Toutefois, la littérature du Web Sémantique ne propose pas d'outils pour la modélisation spatio-temporelle. Dès lors, le premier verrou consiste à proposer un modèle de représentation de connaissances spatio-temporelles à l'aide des technologies du Web Sémantique.

Les technologies du Web Sémantique sont définies en monde ouvert en assumant que les informations représentées sont incomplètes. Cependant, l'analyse de données spatio-temporelles nécessite d'identifier les propriétés utiles pour celle-ci. Afin d'obtenir des résultats complets, il est nécessaire que ces propriétés ou objets soient définis en monde fermé. Le deuxième verrou consiste à définir un modèle et une méthodologie hybride permettant la coexistence de données établies sous l'hypothèse du monde fermé et du monde ouvert.

1.4/ CONTRIBUTION

Dans le cadre de la conception de bases de données traditionnelles, le modèle conceptuel est considéré comme une étape intermédiaire révélant la vision d'un concepteur pour répondre aux besoins d'un domaine particulier. Dès lors, le modèle conceptuel devient très dépendant du concepteur et le passage au modèle logique induit généralement une structure très difficile à appréhender pour les utilisateurs et nuit à l'exploitation du système final.

1.4.1/ L'ONTOLOGIE : UNE MODÉLISATION FLEXIBLE, ACCESSIBLE ET COMPRÉHENSIBLE SANS PERTE SÉMANTIQUE

La modélisation des données est un outil puissant permettant l'analyse de la sémantique d'une organisation dans le respect de la structure et du flux d'information utilisés dans le cadre des activités de cette organisation. Depuis 1975, l'ANSI (American National Standards Institute) distingue les modèles selon trois couches :

- le modèle **conceptuel** : il décrit la sémantique du domaine au travers de classes d'entités représentant différentes choses ayant une signification pour le domaine, et de relations à propos du lien entre deux classes d'entités. Un modèle conceptuel définit les différents faits ou propositions qui peuvent être exprimés en utilisant le modèle. En ce sens, il fournit les expressions autorisées au sein d'un langage abstrait dont la portée se limite à celle du modèle. L'exemple de modèle conceptuel le plus connu est certainement UML.
- le modèle **logique** : il représente un sous-ensemble du modèle conceptuel qui peut être exprimé en utilisant une technologie particulière. Il décrit, par exemple, des tables, des colonnes, des classes orientées objet, des tags XML, etc.
- le modèle **physique** : il décrit le support physique utilisé pour stocker les données. Dans la méthode Merise, le modèle physique consiste à implanter une base de données dans un système de gestion de bases de données à l'aide du langage SQL.

La grande majorité des modèles spatio-temporels fournis pour la gestion d'entités géographiques spatio-temporelles sont implémentés dans des bases de données de type relationnel, relationnel étendu, objet, objet-relationnel. Chacun de ces modèles est définis au niveau logique nécessitant généralement un modèle conceptuel avant d'être implémenté dans un support physique. Cette méthode de transcription depuis le modèle conceptuel des données vers le modèle physique souffre cependant d'une forte limitation. En effet, lorsque la modélisation de l'information est faite pour créer une base de données, **le modèle conceptuel est obligatoirement différent du modèle logique**, car il n'y a pas de place dans une structure de base de données classique pour capturer, par exemple, les règles métier, créer des relations de subsomption et décrire d'autres aspects clés du modèle conceptuel. **Ces informations sémantiques collectées et documentées dans le cadre de la modélisation initiale sont laissées de côté lors du passage à la modélisation logique des données.** Ces informations laissées de côté doivent alors être implémentées par des développeurs au sein de programmes spécifiques. Ainsi, bien que ces modèles parviennent à gérer les informations géographiques spatio-temporelles, il est, en revanche, impossible de représenter explicitement des connaissances géographiques spatio-temporelles ni d'effectuer des raisonnements pouvant servir de base à la prise de décision.

Depuis la définition des couches de modélisation conceptuelle, logique et physique, il existe certaines technologies pour lesquelles cette architecture n'est plus valable. Les ontologies formelles en sont un bon exemple. En effet, étant admis qu'une ontologie constitue un modèle d'un domaine décrivant les objets évoluant au sein de celui-ci, les trois modèles de données peuvent être chacun pensés comme une seule et même ontologie.

Contrairement aux bases de données classiques, une ontologie peut être vue comme un modèle conceptuel pouvant être implémenté directement sans subir une quelconque perte sémantique.

1.4.2/ L'ONTOLOGIE COMME SUPPORT DE L'INFORMATION SPATIO-TEMPORELLE

Les ontologies constituent un support riche pour la représentation des connaissances qui satisfait aux exigences de la modélisation spatio-temporelle. D'autre part, elles donnent le moyen de raisonner sur les informations de contexte via des règles et des moteurs d'inférences.

La figure 1.5 reprend le modèle DIKW afin d'illustrer l'apport de l'ontologie face aux bases de données classiques. Les bases de données classiques permettent de gérer efficacement l'information spatio-temporelle mais ne peuvent représenter ces informations dans une inférence ou un raisonnement, condition nécessaire pour accéder à l'appellation de connaissance. Les raisonnements impliquent le recours à l'interprétation et témoignent d'outils capables de modéliser certains raisonnements humains. La connaissance est donc une interprétation de l'information basée sur l'expérience, en somme, elle assimile des informations afin d'établir des règles permettant de déduire de nouvelles données, informations ou connaissances. La définition d'une règle témoigne de la compréhension du "comment" fonctionne tel ou tel domaine grâce à la condition "SI" et le conséquent "ALORS". Exemple : SI le nombre de centres commerciaux et d'habitants augmentent dans telle région ALORS il y a un phénomène d'intensification urbaine.

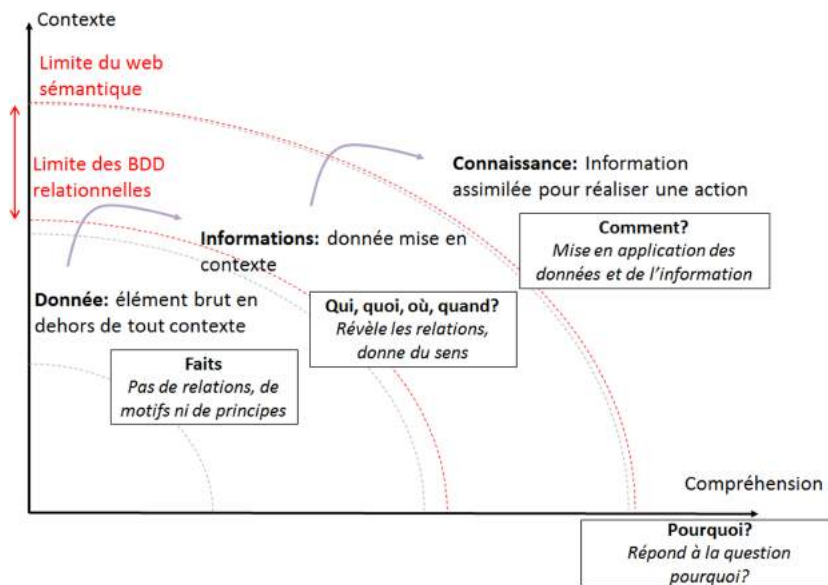


FIGURE 1.5 – Représentation du contexte applicatif : Base de données relationnelles versus ontologie

Afin de permettre la mise en oeuvre des connaissances, il est également nécessaire que celles-ci puissent être assimilées par des humains afin de déterminer les causes (en répondant à la question "pourquoi") et décider des futures actions à mettre en place. En ce sens, la connaissance est intrinséquement destinée à servir l'action, c'est pourquoi les ontologies sont fréquemment employées au sein de systèmes d'aide à la décision permettant aux usagers de s'appropriier plus rapidement et plus facilement les connaissances d'un domaine. Dans ces travaux, nous explorons l'ontologie comme support de modélisation spatio-temporelle afin de fournir :

un modèle générique de représentation et de gestion des connaissances géographiques spatio-temporelles pour l'analyse de phénomènes dynamiques géospaciaux.

1.4.3/ ANALYSE QUALITATIVE ET CAUSALITÉ

Il existe deux approches pour permettre l'étude de phénomènes dynamiques d'un territoire donné : l'approche quantitative et l'approche qualitative. La première approche se base sur des données numériques pour offrir des statistiques à propos des évolutions de la couverture terrestre. Toutefois, cette approche ne fournit des résultats qu'à une échelle globale et ne peut établir clairement les corrélations entre les évolutions observées. L'approche qualitative, quant-à-elle, se base sur la sémantique des données représentées sous la forme d'attributs ou relations. La mise en corrélation de ces données au travers de règles permet de définir des connaissances explicites à partir de données implicites.

Dans ces travaux, le modèle proposé s'oriente principalement sur l'approche qualitative pour représenter les phénomènes dynamiques en se basant sur la relation de filiation. La relation de filiation est au coeur du modèle et permet d'établir un lien spatio-temporel entre les différents états d'une entité au cours du temps. Dans ces travaux, la relation de

filiation est déclinée au travers de plusieurs couches d'expressivité illustrées sur la figure 1.6.

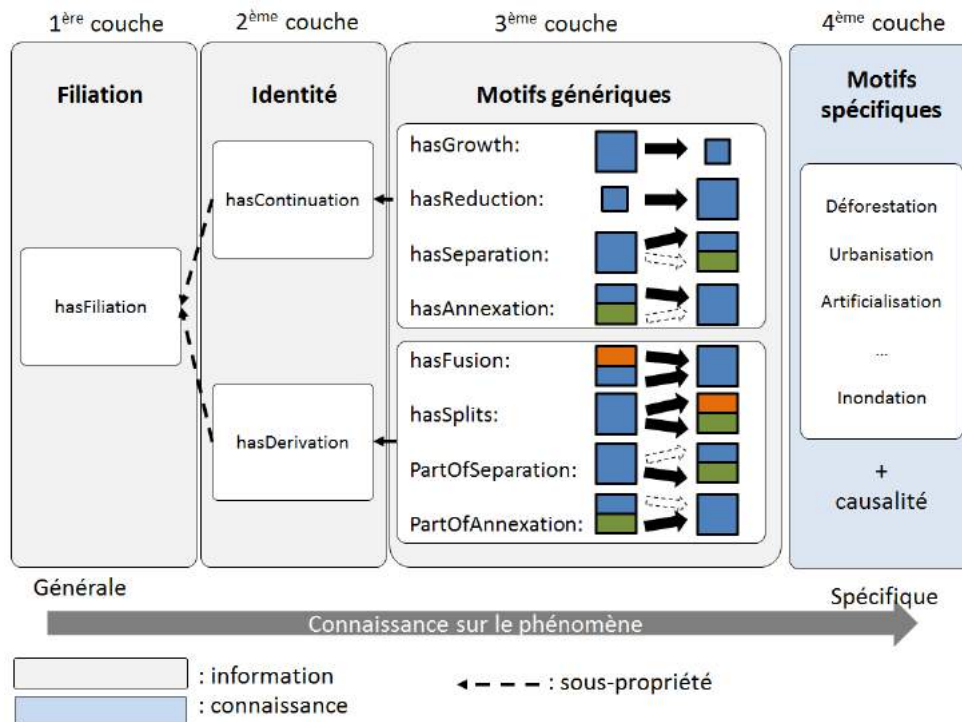


FIGURE 1.6 – Liste des couches et motifs génériques décrits dans le modèle

La première couche du modèle possède une expressivité limitée mais permet de décrire la généalogie des entités d'un territoire donné. La deuxième couche se fonde sur la notion d'identité pour permettre de s'intéresser au cycle de vie des entités. La troisième couche décrit des motifs spatiaux génériques et permet de définir la dynamique d'évolution spatiale des entités. Enfin, la quatrième couche s'appuie sur les motifs génériques de la troisième couche et tente de leur donner une interprétation à l'aide de la sémantique du contexte de l'environnement géospatial définie dans l'ontologie. Le résultat de cette interprétation donne lieu à des motifs spécifiques pouvant être assimilés à des phénomènes. Contrairement à l'approche quantitative, les phénomènes dynamiques sont explicitement décrits entre les entités et peuvent être localisés précisément. D'autre part, cette approche facilite la compréhension des causalités d'évolution entre les entités.

L'approche qualitative exploitée au sein d'une ontologie par notre modèle constitue une contribution majeure et un outil performant pour permettre :

- **l'étude de la profondeur du changement** : la profondeur du changement utilise la sémantique pour répondre à la question : "à quel point le changement est-il significatif?". En effet, dans le cadre de l'étude de la dynamique d'entités spatio-temporelles, tous les changements ne sont pas équivalents. Par exemple, la transformation d'une forêt de conifère en une forêt de feuillus est généralement moins significative, pour l'utilisateur, qu'une zone urbaine transformée en lac. Le modèle proposé s'appuie sur la hiérarchie de concept autorisée par les ontologies pour permettre de distinguer différents degrés de changement.

- **l'étude des motifs de tendance** : les phénomènes détectés sur la base de la relation de filiation ne peuvent être établis qu'entre deux instants de temps consécutifs. Les motifs de tendance se définissent par l'étude de dynamiques progressives se produisant sur plusieurs instants de temps consécutifs.
- **l'étude des évolutions connexes** : la relation de filiation telle qu'étudiée jusqu'alors ne peut expliquer à elle seule toutes les causalités d'évolution des entités d'un territoire. Par exemple, la création de centre commerciaux ou de réseaux routiers à proximité de territoires agricoles est généralement le témoin d'une dynamique d'intensification urbaine pouvant impliquer des répercussions sur le paysage agricole environnant. Dès lors, il est nécessaire de fournir des outils capable de mettre en relation les dynamiques d'entités voisines.

Le modèle présenté dans ces travaux permet l'étude de la profondeur du changement, l'étude des motifs de tendance ainsi que l'étude des phénomènes connexes.

1.4.4/ AIDE À LA DÉCISION

Le modèle décrit dans ces travaux, appelé *Continuum*, est un outil d'analyse capable de modéliser le contexte d'un environnement géospatial dynamique dans l'optique d'assister les experts et aider les chargés des territoires dans leur politique de gestion. La figure 1.7 illustre le fonctionnement global de notre système en s'appuyant sur la décomposition ontologique définie par [Frank, 2003] ainsi que le modèle DIKW.

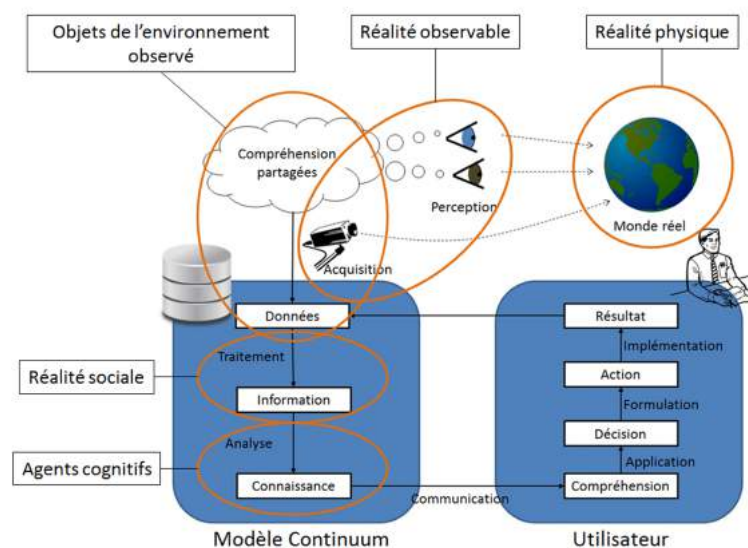


FIGURE 1.7 – Illustration du processus de traitement des données au sein du modèle Continuum

Les connaissances inférées par le modèle offre une meilleure compréhension de la dynamique de l'environnement géospatial. De fait, l'utilisateur peut ainsi mettre à profit ces connaissances afin de prendre une décision sur l'aménagement du territoire, cette décision implique généralement la mise en place d'un plan d'action. Les modifications du

territoire résultantes du plan d'action peuvent alors être injectée dans le modèle afin de servir de support à de nouveaux traitements et analyses permettant de juger l'efficacité des moyens mis en œuvre.

1.5/ PLAN DE LA THÈSE

Cette section présente une brève vue d'ensemble de chacun des chapitres composant cette thèse.

1.5.1/ CHAPITRE 2 : ÉTAT DE L'ART

Ce chapitre a pour objectif d'introduire le domaine de la modélisation spatio-temporelle ainsi que celui de l'ingénierie des connaissances. Il présente des approches et solutions existantes et met en avant les avantages et les limites dans l'optique de concevoir un système d'information géographique de nouvelle génération en se basant sur une ontologie et les technologies afférentes. Dans un premier temps, nous introduirons le domaine de la modélisation spatio-temporelle ainsi que des propositions majeures du domaine. Puis nous introduirons la famille des langages ontologiques à l'aide des logiques de description qui constitue un pilier majeur des travaux actuels du domaine des ontologies. Les logiques de description sont mise en œuvre au sein de systèmes grâce à des langages de représentation d'ontologie. Le langage OWL et plus particulièrement OWL-DL (DL pour Description Logic), conçu pour le Web Sémantique, est aujourd'hui le représentant par excellence des langages de représentation d'ontologie. Ce langage ainsi que les technologies du Web Sémantique seront étudiés en troisième partie de notre état de l'art. La grande majorité des modèles spatio-temporels fournis pour la gestion d'entités géographiques spatio-temporelles sont implémentés dans des bases de données de type relationnel, relationnel étendu, objet, objet-relationnel. Toutes ces méthodes de conception de bases de données possèdent une limitation majeure : il existe en général un écart entre les modèles conceptuels et les modèles logiques qui s'accroît avec la divergence des formalismes. Les modèles logiques peuvent rapidement devenir incompréhensible pour un utilisateur. Les ontologies permettent de lever cette difficulté en offrant une représentation du modèle conceptuel directement au sein de la base de données pour qu'il puisse être accessible. La quatrième section de l'état de l'art établit un comparatif entre l'approche relationnelle et l'approche sémantique pour justifier l'apport de l'ontologie comme support de la modélisation spatio-temporelle. Ce premier chapitre se conclut par l'étude des différents travaux de modélisation sémantique de données spatiales ou temporelles.

1.5.2/ CHAPITRE 3 : LE MODÈLE CONTINUUM

Ce chapitre propose un modèle permettant de représenter des connaissances explicites à propos d'entités géographiques spatio-temporelles ainsi que de fournir des raisonnements pour comprendre la dynamique de ces entités. Le modèle décrit les entités au sein de couches spatio-temporelles et de transitions spatio-temporelles. Les couches permettent de représenter les différents états du cycle de vie des entités tandis que les transitions permettent de lier ces différents états ainsi que de décrire la connaissance

liée à la dynamique de ces entités. Les contraintes de chaque élément composants les couches et les transitions sont explicitées dans ce chapitre.

1.5.3/ CHAPITRE 4 : IMPLÉMENTATION ET CALCUL DES COUCHES 1 ET 2

Ce chapitre présente l'implémentation de notre modèle. L'un des objectifs majeurs de notre modèle consiste à fournir un support pour l'étude de la dynamique des phénomènes géospatiaux. Pour cela, la relation de filiation est au coeur de notre modèle. Les jeux de données de la littérature ne fournissent pas de relation de filiation. Dans ce chapitre, nous proposons une méthode pour permettre de détecter et intégrer les relations de filiation au sein du modèle afin de servir de base d'étude pour la découverte de connaissance. Dans un deuxième temps, l'ajout de contrainte d'identité permet de décrire la couche 2 du modèle pour l'étude du cycle de vie des entités.

1.5.4/ CHAPITRE 5 : CALCUL DE LA COUCHE 3 : LES MOTIFS GÉNÉRIQUES

Ce chapitre décrit la couche 3 du modèle basée sur la définition de motifs génériques permettant de décrire la dynamique spatiale des entités du modèle. Cette couche spécialise la couche 2 en intégrant des contraintes permettant d'accroître son expressivité. Les motifs sont dits "génériques" car ils sont indépendants du domaine d'étude au sens où ils n'intègrent pas de connaissances liées au contexte.

1.5.5/ CHAPITRE 6 : CALCUL DE LA COUCHE 4 ET DÉCOUVERTE DES CONNAISSANCES

Ce chapitre met en avant les capacités d'analyse qualitative de notre modèle. Ce type d'analyse permet d'établir précisément les relations spatio-temporelles entre les différentes entités mettant en évidence les liens de causalité à l'origine de ces évolutions. Dans cet optique, ce chapitre présente une méthodologie permettant d'étudier la profondeur du changement, des outils permettant de rechercher des motifs de tendance ainsi qu'étudier les interactions entre des entités dynamiques voisines au sein du graphe.

1.5.6/ CHAPITRE 7 : EXPÉRIMENTATION

Ce chapitre présente un cas d'application du modèle Continuum sur le jeu de données Corine Land Cover. Il montre un exemple d'évolution au sein des quatre couches du modèle ainsi que des chiffres illustrant le nombre de motifs détectés. L'architecture de l'application implémentée est également abordée.

1.5.7/ CHAPITRE 8 : CONCLUSION

Ce chapitre conclut ce document. Il dresse un bilan des contributions apportées par les travaux présentés dans cette thèse et propose également de futurs travaux afin d'étendre les capacités du modèle.

ETAT DE L'ART

Ce chapitre présente le domaine de la modélisation spatio-temporelle. La modélisation spatio-temporelle est depuis longtemps dominée par l'usage de bases de données relationnelles étendues avec diverses fonctionnalités. L'approche se distingue par l'usage d'une ontologie définie à l'aide des technologies du Web Sémantique, elles-mêmes basées sur les logiques de description. Ainsi, nous introduirons les logiques de description permettant traditionnellement de décrire un base de connaissance. Le langage OWL, basé sur ces logiques de description, sera également introduit en tant que langage de représentation des connaissances permettant la définition d'ontologie. Dans la suite de l'état de l'art, nous établirons un comparatif des différences majeures existantes entre l'approche relationnelle et l'approche sémantique. Enfin, nous étudierons les différents travaux et outils consacrés à la modélisation spatiale et temporelle au sein de la littérature du Web Sémantique.

Sommaire

2.1	Modélisation spatio-temporelle	22
2.1.1	Notion de temps	22
2.1.2	Notion d'espace	26
2.1.3	Notions spatio-temporelles	28
2.2	Étude des approches de modélisation spatio-temporelle	32
2.2.1	Approches basées sur les "snapshots"	32
2.2.2	Approches basées sur les entités	33
2.2.3	Approches basées sur les évènements	35
2.2.4	Approches basées sur les graphes	36
2.2.5	Discussion	37
2.3	Logiques de description et ontologies	39
2.3.1	Base de connaissances	39
2.3.2	Définition et usage d'ontologie en informatique	41
2.3.3	Le langage de représentation des ontologies : OWL	42
2.4	L'approche Web Sémantique et l'approche classique	47
2.4.1	Comparatif des caractéristiques intrinsèques des systèmes en monde ouvert et en monde fermé	47
2.4.2	Interprétation des contraintes : de la contrainte d'intégrité à l'axiome logique	50
2.5	Modélisation spatio-temporelle dans le Web Sémantique	53
2.5.1	Modélisation spatiale dans le Web Sémantique	53
2.5.2	Modélisation du temps dans le Web Sémantique	60
2.5.3	Ontologie de Fluent : Modélisation de la dynamique d'un système	61

2.6 Discussion	66
2.6.1 Apport pour le domaine de la modélisation spatio-temporelle . .	67
2.6.2 Apport pour le domaine de la dynamique d'utilisation des terres et des couvertures terrestres	69

2.1/ MODÉLISATION SPATIO-TEMPORELLE

Au sein de la communauté, plusieurs modèles ont été proposés afin de répondre aux besoins de la modélisation spatio-temporelle. Dans cette section, nous introduirons plusieurs notions temporelles, spatiales et spatio-temporelles impliquées dans la mise en place d'un modèle conceptuel de données spatio-temporelles. Dans un second temps, nous étudierons différentes approches de modélisation proposées dans la littérature.

2.1.1/ NOTION DE TEMPS

L'introduction du temps dans les modèles employés par les SIG induit la nécessité de lier simultanément les dimensions spatiales et temporelles. L'intégration du temps conduit à la modification des modèles spatiaux. La représentation du temps est influencée par le phénomène à modéliser. Conceptuellement, l'objectif fondamental de toutes bases de données temporelles est d'enregistrer ou de dépeindre les changements au fil du temps. Le changement est normalement décrit comme un événement ou une succession d'événements. La définition la plus générale pour décrire un évènement est « quelque chose d'important se passe ». Dans le cadre de la modélisation spatio-temporelle une meilleure définition pourrait être « un changement d'état sur la position, la sémantique de l'entité, ou les deux ». Par exemple, un changement dans les espèces dominantes au sein d'une forêt, un incendie de forêt, le changement de propriétaire d'une terre, ou la construction d'une route. Les changements, et donc aussi les événements, peuvent être distingués en fonction de leur profil temporel en quatre types [Peuquet, 1999] :

- continu : se produit sur un certain intervalle de temps
- majeur : se produit régulièrement
- sporadique : se produit parfois
- unique : se produit une seule fois

La durée et la fréquence sont des caractéristiques importantes pour décrire un modèle temporel. Les changements relatifs aux entités ou aux emplacements peuvent être soudains ou progressifs. Les entités apparaissent, évoluent à travers divers changements, puis disparaissent au fil du temps. Elles peuvent également changer de façon complexe. Une question essentielle est de maintenir ou de changer l'identité des entités : comment suivre les différents états d'une même entité, comme l'implantation d'une ville lorsqu'elle change à travers le temps ? Quels genres d'évolution représentent un changement dans l'identité d'une entité ? Spatialement, une entité peut se déplacer, s'agrandir, rétrécir, changer de forme, se scinder en deux, ou fusionner avec une entité voisine. Bien que l'espace et le temps soient continus, ils sont classiquement divisés en unités discrètes de longueur uniforme ou variable. Le temps est divisé en unités qui sont nécessairement différentes de celles de l'espace (nous ne pouvons pas mesurer le temps en mètres ou en pieds). Les unités temporelles peuvent être des secondes, des minutes, des jours, des

saisons, ou d'autres unités qui peuvent être pratique. Que l'on utilise une échelle temporelle unique ou une hiérarchie d'échelles, la plus petite unité temps d'enregistrement est appelé un chronon [Jensen et al., 1993].

Densité du temps :

Le temps propose trois types de modèles de structuration [André, 2011], continu, dense et discret. La distinction entre le modèle dense et continu ne présente que peu d'intérêt car les nécessités techniques pour la modélisation sont les mêmes. Nous retiendrons que dans le cas d'un temps dense ou continu, le système peut changer à n'importe quel moment. Dans le cas d'un temps discret, la progression du temps est assurée par l'occurrence de tics d'horloge et le système ne peut alors évoluer qu'à des instants précis. En général, c'est la nature du phénomène géographique qui dicte si on prend un temps discret ou dense. Ainsi, le phénomène géographique, ou plutôt les changements intervenant sur une entité spatiale, peut être continu comme pour l'étude du mouvement d'un glacier ; ou bien il peut être discret comme dans le cas du changement des frontières administratives d'un état. Quel que soit le type de changement, [Pelekis et al., 2004] distingue les changements uniformes, par exemple lors du suivi d'un avion ou d'un bateau, et les changements irréguliers, par exemple un taxi circulant dans un centre-ville. Enfin, on notera que certaines entités sont statiques dans le temps et ne changent jamais.

Représentation du temps

L'approche de la modélisation du temps dépend de son usage. Les données susceptibles de porter des références temporelles au sein d'une application spatio-temporelle sont nombreuses. Il peut s'agir d'une carte (une couche géographique), des objets composants cette carte ou encore les événements induisant un changement au sein des objets. Dès lors, l'évolution peut être modélisée suivant plusieurs niveaux de détails soullevant ainsi la question : « A quel niveau de détail associer les références temporelles ? ».

Dans [Worboys, 1994], l'auteur dénote deux approches pour ajouter des références temporelles dans un système. Dans la première option, les références temporelles sont ajoutées sur chaque couche du SIG. Cette option nécessite moins de capacité de stockage puisque l'évolution se limite à un suivi couche par couche. Cependant l'expression de la temporalité et du changement est limitée au cours de sa période d'existence. La seconde option, quant à elle, fusionne l'espace et le temps au niveau de la primitive spatiale de l'objet (point ou polygone). Dans ce second cas, on obtient une expressivité plus fine du changement au sein des objets eux-mêmes. Cependant, cette option nécessite de grande capacité de stockage puisque chaque objet est modélisé individuellement afin de suivre l'évolution de chacun d'entre eux. Dans [Worboys, 2005], l'auteur va plus loin pour présenter les étapes possibles dans l'introduction de nouvelles capacités temporelles dans les systèmes d'information géographique. Ces étapes sont au nombre de quatre.

- **Etape 0** : SIG statiques. Dans cette étape, la situation géographique représentée est statique. Aucune de ses propriétés ne peut varier. Cependant, la couche affichée peut représenter une situation du passé, du présent ou bien une prévision. (une seule couche)
- **Etape 1** : captures instantanées temporelles. A cette étape, plusieurs couches sont disponibles dans le SIG, chacune datée et représentant un espace géographique à différentes époques. Cette approche est jusqu'ici la plus utilisée dans les bases de données spatio-temporelles. Elle permet de répondre à des questions temporelles

comme : « Quel était l'état de cet objet à cet instant ? » ou « A quel moment cet objet avait cet état ? ». Dans le cas d'informations spatio-temporelles, on pourra répondre à des questions comme : « Où était cet objet à cet instant ? » ou « A quel instant cet objet était à cet endroit ? ». (multicouche temporelle)

- **Etape 2** : les objets changent. A cette étape, on passe d'une séquence temporelle d'espace géographique au changement qui peut intervenir sur les objets qui y sont représentés. Dès lors, l'évolution globale d'ensemble géographique laisse place à l'évolution individuelle de chaque objet qui le compose. Pour chaque objet, les primitives de changement sont : création, destruction, apparition, disparition, dérivation, fusion, séparation. En outre, le mouvement est considéré comme un changement particulier. (attributs/rerelations temporels)
- **Etape 3** : les évènements et les actions. Cette dernière étape s'attaque plus en détail au traitement du changement en termes d'évènements et d'actions. (Évènements temporelles)

Tout d'abord, il faut distinguer deux types d'entités du monde réel :

- **Les continuant/endurant** : ceux qui durent dans le temps (ex : tables, maisons, personnes).
- **Les occurrents/perdurant** : ceux qui se produisent puis se terminent (ex : repas, réparation d'une maison, vie d'une personne).

Les continuants font généralement référence à des objets tandis que les occurrents sont davantage assimilés à des processus (évènements ou actions). Par exemple, il y a une différence entre une ville (continuant), dont les caractéristiques sont enregistrées par un recensement et une enquête une fois tous les dix ans, et les évènements (occurrents) impliqués dans la croissance urbaine, la migration et le développement, qui représentent la dynamique de la ville. On constate alors que les occurrents sont des processus qui s'appliquent sur des objets (continuants) afin de les inscrire dans une dynamique. Ainsi l'étape 3 tente de modéliser des occurrents complexes et de définir comment les objets sont impliqués dans cette modélisation en étudiant leur dynamique.

Topologie temporelle

Le modèle d'Allen [Allen, 1983] est le plus connu et le plus utilisé pour modéliser la topologie temporelle. Sa théorie de l'action et du temps propose un formalisme basé sur la logique temporelle qui est utilisée pour représenter et raisonner à propos d'évènements, actions, croyances, intentions ou causalité. La primitive de temps utilisée par Allen est l'intervalle, cependant la prise en compte de point de temps fût rapidement intégrée dans le modèle. Sur cette base, Allen définit un ensemble de 13 relations binaires mutuellement exclusives entre les intervalles. Le temps est supposé linéaire, dense et infini dans le passé et le futur. Les relations définies sont les suivantes :

- **Equals(i,j)** : i et j représentent le même intervalle de temps
- **Meets(i,j)** ou **MetBy(j,i)** : j démarre immédiatement après i
- **Before(i,j)** ou **After(j,i)** : i démarre et finit avant j avec un peu de temps qui s'écoule entre la fin de i et le début de j
- **Overlaps(i,j)** ou **OverlappedBy(j, i)** : i démarre strictement avant j, ils ont quelques points en commun et i termine strictement avant j
- **Contains(i,j)** ou **During(j,i)** : j démarre strictement avant i et termine strictement après i
- **Starts(i, j)** ou **StartBy(j,i)** : i et j démarrent en même temps mais j continu (stricte-

ment) après que i ait terminé

- finishes(i,j) ou finishedBy(j,i) : i et j terminent ensemble mais j démarre strictement avant i

La figure 2.1 illustre les relations topologiques entre les intervalles. La topologie temporelle est également donnée entre des points de temps et entre un intervalle et un point de temps.

	Point-Point	Point-Intervalle	Intervalle-Intervalle
A before B B after A			
A meets B B metBy A			
A overlaps B B overlappedBy A			
A finishedBy B B finishes A			
A contains B B during A			
A starts B B startedBy A			
A equals B B equals A			

FIGURE 2.1 – Illustration des relations topologiques temporelles définies par Allen

Quelques notions supplémentaires

La représentation du temps pour des objets spatiaux apparaît sous diverses formes qui prennent leur importance lors de la modélisation.

Granularité temporelle : Une question qui se pose lorsqu'il s'agit d'informations temporelles concerne la granularité des valeurs. Dans certaines applications, la granularité utilisée [Wiederhold et al., 1991] est le jour, dans d'autres, ce sont les secondes ou bien des années. La plupart des travaux de la littérature modélisent le temps comme un ensemble discret totalement ordonné. Le temps discret implique qu'il existe une certaine granularité (années, jours, secondes, etc) qui est valide. Dans le cas contraire, des décalages sémantiques se créent lorsque des opérations combinent les informations temporelles.

Transaction/temps valide [Jensen et al., 1992] : Les changements ou processus spatiaux sont modélisés selon deux approches distinctes. Le temps transactionnel qui indique le temps durant lequel un événement est effectivement enregistré dans la base de données et le temps valide qui décrit le temps durant lequel un événement intervient dans le monde réel. Certains modèles spatio-temporelles supportent ces deux approches et sont qualifiés de bi-temporels [Clifford et al., 1992].

Durée de vie [Clifford et al., 1992],[Jensen et al., 1992] : Dans les publications citées ci-avant, la durée de vie est définie comme suit : "La durée de vie d'un objet contenu dans une base de données est le temps pendant lequel cet objet est défini. La durée de vie valide d'un objet d'une base de données se réfère au moment où l'objet correspondant existe dans la réalité modélisée, alors que la durée de vie transactionnelle renvoie au

moment où l'objet de la base de données est présent dans la base de données. Si l'objet (attribut, tuple, relation) dispose d'une estampille temporelle associée alors la durée de vie de l'objet est la valeur de l'estampille. Si les composants d'un objet sont estampillés, alors la durée de vie de l'objet est déterminée par le modèle de données utilisé".

2.1.2/ NOTION D'ESPACE

L'espace, bien que largement étudié depuis la création des premiers SIG, se doit d'être bien défini afin d'anticiper les futures contraintes liées à l'ajout d'une dimension temporelle.

- Structure de l'espace : il existe deux approches pour stocker de l'information géographique dans les SIG. Ces approches sont les modèles de données spatiaux raster et vectoriel. Les données raster sont organisées en tableaux de cellules, qui peuvent être des pixels ou des voxels selon que l'on souhaite une représentation 2D ou 3D. Dans ce modèle, l'espace est partitionné en grille où chaque cellule est représentée par sa position sur la grille. L'approche vectorielle, quant-à-elle, décrit chaque objet spatial à l'aide d'un point de départ et d'un point d'arrivée. En général, l'approche vectorielle est privilégiée car elle permet un stockage plus efficace en ne stockant que les données pertinentes et non l'espace géographique entier.
- Orientation/direction : cette notion est rarement intégrée aux modèles proposés dans la littérature et fait référence à la capacité d'un modèle à supporter les caractéristiques d'orientation et direction que les objets du monde réel sont susceptibles d'entretenir dans l'espace. Par exemple, on doit pouvoir définir des relations comme : "Sur le côté gauche de", "A droite".
- Mesures spatiales : il s'agit ici de savoir si le modèle permet d'obtenir des informations de taille, périmètre, aire, distance, etc. A propos d'objets spatiaux et également si le modèle supporte des opérations de comparaison telles que : "plus grand", "plus long", "plus large", etc.

Topologie spatiale

Lorsque nous observons un paysage, des objets tels que des forêts, des bâtiments, des rivières, etc, sont "vus" ou "perçus" dans leur contexte. La notion de voisinage est implicite. Par exemple, une rivière "traverse" une parcelle, un bâtiment "est situé dans" une agglomération, une agglomération est "juxtaposée" à une autre agglomération. Avec ce type d'observation de l'espace géographique naît l'étude de la topologie.

Du point de vue de l'espace géographique, la topologie est l'ensemble des relations perçues qui nous permettent de situer les objets les uns par rapport aux autres. Dans la littérature, plusieurs modèles ont été proposés afin d'étudier la topologie d'un espace géographique. La topologie dans l'espace est majoritairement dominée par deux modèles que sont le modèle 9-IM [Strobl, 2008] et le modèle Region Connection Calculus (RCC) [Randell et al., 1992]. Le premier se fonde sur l'étude des intersections entre les primitives topologies des objets spatiaux (points, lignes ou polygone). L'autre permet de décrire uniquement des régions par les possibles relations qu'elles ont entre elles.

Les deux modèles possèdent 6 prédicats topologiques en commun (la dénomination des relations topologiques varie quelque peu selon les modèles de la littérature, dans ces travaux, nous nous accordons sur la dénomination définie par l'OGC (Open Geospatial Consortium) pour laquelle nous proposons une traduction française) :

- **Disjonction** : l'intersection des deux géométries est un ensemble vide
- **Adjacence** : les géométries possèdent des points en commun, mais aucun d'entre eux n'est situé à l'intérieur de la géométrie de l'autre.
- **Contenance** : la seconde géométrie se situe entièrement dans la première géométrie. La contenance est l'exact opposée de l'inclusion.
- **Inclusion** : la première géométrie se situe entièrement dans la seconde géométrie. L'inclusion est l'exact opposée de la contenance.
- **Égalité** : les deux géométries possèdent les mêmes coordonnées.
- **Chevauchement** : les deux géométries doivent être du même type de données (point, ligne ou polygone). Les deux géométries sont différentes et/ou situées à des endroits différents mais possèdent une intersection.

Le modèle RCC 8 ne s'applique que sur des polygones et possède son propre vocabulaire pour définir les relations, ainsi dans le modèle RCC8, les relations topologiques sont définies comme suit :

- Disjonction = **DC** (Disconnected)
- Adjacence = **EC** (Externally Connected)
- Egalité = **EQ** (Equal)
- Chevauchement = **PO** (Partially Overlapping)

RCC8 possède la particularité de distinguer deux types de contenance et d'inclusion selon que les géométries possèdent ou non une relation d'adjacence en supplément. Les relations sont les suivantes :

- Contenance = **TPPi** (Tangential Proper Part inverse) : la seconde géométrie se situe entièrement dans la première géométrie. Le bord des géométries se touche.
- Inclusion = **TPP** (Tangential Proper Part) : la première géométrie se situe entièrement dans la seconde géométrie. Le bord des géométries se touche.
- Contenance = **nTTPi** (non-Tangential Proper Part inverse) : la seconde géométrie se situe entièrement dans la première géométrie. Le bord des géométries ne se touche pas.
- Inclusion = **nTTP** (non-Tangential Proper Part) : la première géométrie se situe entièrement dans la seconde géométrie. Le bord des géométries ne se touche pas.

La figure 2.2 illustre les prédicats définis pour le raisonnement spatial qualitatif dans le modèle RCC8.

Le modèle 9-IM s'appliquant entre plusieurs types de données géométriques, deux relations sont également présentes en sus de celles citées précédemment :

- **L'intersection** : l'intersection des deux géométries, quelque soit leur type, n'est pas un ensemble vide. L'intersection est l'exact inverse de la disjonction.
- **Le croisement** : ce prédicat ne s'applique qu'entre deux lignes ou un polygone et une ligne (ou des types plus complexes de données comme multipoint/polygone, multipoint/ligne ou ligne/multipolygone). L'ensemble d'intersection des deux géométries se situe à l'intérieur des deux géométries.

La figure 2.3 illustre les différentes relations topologiques pouvant exister entre des points, des lignes ou des polygones dans le modèle DE-9IM. A noter que ces prédicats spatiaux existent également pour des types de données plus complexes comme les multipoints, les multilignes ou les multipolygones. Dans un souci de clarté, nous avons volontairement laissé de côté ces autres cas pour illustrer nos propos.

Prédicats	Polygone	Prédicats	Polygone
A equal B EQ(A,B)		A contains B TPPi(A,B)	
A disjoint B DC(A,B)		A within B TPP(A,B)	
A overlap B PO(A,B)		A contains B nTPPi(A,B)	
A touch B EC(A,B)		A within B nTPP(A,B)	

FIGURE 2.2 – Illustration des relations topologiques spatiales du modèle RCC8

Prédicats	Point – Ligne – Polygone	Prédicats	Point – Ligne – Polygone
A equal B	 Point/Point Ligne/Ligne Polygone/Polygone	A cross B	 Ligne/Ligne Polygone/Ligne
A disjoint B	 Point/Point Ligne/Ligne Polygone/Polygone Point/Polygone Point/Ligne Polygone/Ligne	A within B	 Point/Polygone Point/Ligne Polygone/Ligne Ligne/Ligne
A intersects B	<p>Le prédicat d'intersection est l'opposé de la disjonction ainsi n'importe quel cas de contenance, adjacence ou chevauchement entre n'importe quel type de géométries satisfait le prédicat, c'est pourquoi nous n'illustrerons pas tous les cas possibles.</p>	A contains B	 Point/Polygone Point/Ligne Polygone/Ligne Ligne/Ligne
A touch B	 Point/Ligne Ligne/Ligne Point/Polygone Polygone/Ligne Polygone/Polygone	A overlap B	 Ligne/Ligne Polygone/Polygone

FIGURE 2.3 – Illustration des relations topologiques spatiales du modèle DE-9IM

2.1.3/ NOTIONS SPATIO-TEMPORELLES

Les SIG spatio-temporels possèdent cette caractéristique de modéliser l'espace, le temps et la sémantique de manière indépendante ou combinée. Dès lors, nous tenterons d'obtenir un aperçu de toutes les implications liées à cette caractéristique autant au niveau de la représentation que de l'interrogation.

2.1.3.1/ PRIMITIVES DE MODÉLISATION

La modélisation spatio-temporelle met en jeu différentes primitives selon que l'on souhaite représenter la spatialité ou la temporalité d'un objet. Parmi les primitives les plus utilisées pour représenter une entité, on trouve les points situés à l'aide de coordonnées de longitude ou latitude, ou encore les lignes et les régions constituées d'un ensemble de points. Le temps est difficile à formaliser, plus d'un type de conceptualisation du temps est nécessaire pour les SIG. En réalité, il s'agit plutôt de représenter des entités liées au temps. En général, on retiendra trois options pour représenter la structure du temps :

- Des points de temps (instant)
- Des segments de temps (intervalle)
- Des occurrences de temps (événement)

Les primitives permettent de définir le niveau d'abstraction du monde réel. En général, c'est l'observation de la réalité à modéliser qui définit la primitive. Par exemple, une ville peut être représentée par un point ou par une région. Une foule de gens peut être représentée par une multitude de points ou bien par un polygone. On note que le choix est conditionné par l'usage de l'application.

2.1.3.2/ LE CHANGEMENT

Les changements peuvent intervenir sur la position, la forme et sur la description (propriétés ou attributs) des objets. Ces changements spatiaux ou sémantiques peuvent intervenir simultanément ou indépendamment les uns des autres. Ainsi, la morphologie (propriété géométrique), la topologie (relation spatiale) et les attributs (propriétés et relations sémantiques) d'un objet spatio-temporel peuvent changer ou ne pas changer dans le temps, donnant lieu à 8 scénarios différents [Armstrong, 1988] (voir figure 2.4).

En outre, le type de changement peut être continu ou discret, c'est ce critère qui définit si un modèle prend en charge les objets spatio-temporels du monde réel qui changent de manière continue ou seulement les objets qui subissent des modifications ponctuelles.

Un critère supplémentaire qui concerne davantage les approches continues traite de la gestion du mouvement des objets spatiaux au fil du temps [Parent et al., 2013]. Ceci est un facteur décisif supplémentaire qui différencie les modèles qui prennent en charge les changements de la position et / ou de morphologie des objets dans le continuum espace-temps.

2.1.3.3/ REPRÉSENTATION D'UNE ENTITÉ

L'identité est une caractéristique unique qui distingue un objet d'un autre, indépendamment de ses propriétés, de ses valeurs ou de sa structure. Les objets peuvent se déplacer, les villes et les régions peuvent modifier leurs frontières tout en conservant leur identité. De nouvelles entités peuvent se former à partir d'anciennes entités comme par exemple dans le cas d'une redistribution des parcelles d'un territoire. On constate alors que des objets peuvent changer leurs attributs spatiaux ou non tout en conservant leur identité. Une question majeure survient alors : "Jusqu'à quel point une entité peut-elle varier avant de perdre son identité ?".

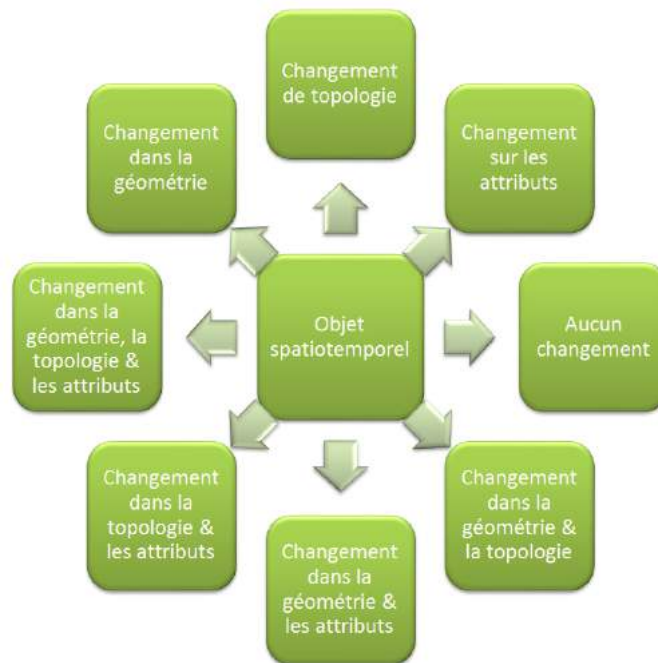


FIGURE 2.4 – Illustration des 8 scénarios de changements

Parfois, il peut être plus approprié de détruire un objet et d'en recréer un à cause d'un trop grand nombre de changements. L'identité est un point clé pour déterminer l'existence ou la non-existence d'un objet autant que pour répertorier les similitudes ou les différences entre les objets.

L'existence fait référence à la présence physique d'un objet. Pour les objets conceptuels, l'existence correspond à la perception d'un objet. On répertorie généralement trois états d'identité [Stell et al., 2011], la création, la continuation, l'élimination. Certains travaux de la littérature prévoient davantage d'états d'identité comme dans [Hornsby et al., 2000] qui définit neuf états d'identité. Cependant des problèmes connus de représentation peuvent survenir lorsqu'il s'agit de conserver l'identité d'objet à travers le temps. D'après la littérature, deux exemples montrent ce problème de conservation de l'identité.

- Un a priori naturel, à propos d'objets dans l'espace, tend à considérer que deux objets occupant exactement le même espace au même instant ont la même identité. Toutefois, cette proposition peut poser quelques problèmes : supposons des entités occupant un espace physique et possédant une identité donnée. Prenons l'exemple d'un objet "maison" possédant l'identité H à l'instant t et d'un autre objet "maison sans cheminée" possédant l'identité H^- . Supposons maintenant qu'entre l'instant t et t' , un orage arrache la cheminée de la maison. A l'instant t' , l'objet "maison" continue d'avoir l'identité H et l'objet maison sans cheminée continue d'avoir l'identité H^- . Cependant, on constate qu'à présent ces objets ayant l'identité H et H^- possèdent exactement les mêmes attributs et occupent le même espace tout en ayant des identités différentes. Ce type de problème a été étudié par Heller [Heller, 1990].
- Un autre exemple de problème lié à l'identité des objets est mis en évidence dans

[Worboys, 2005] avec le célèbre paradoxe du bateau de Thésée. Thésée, d'après la mythologie grecque, a tué le Minotaure dans le labyrinthe de l'île de Crète. Le paradoxe prend l'exemple du scénario du voyage jusqu'en Crète pour rencontrer le Minotaure. Le bateau de Thésée commence à fuir, parce que le bois doit être remplacé. Thésée remplace donc, planche par planche, chaque partie de son navire et jette par-dessus bord le vieux matériel. Il serait naturel de penser que le navire dans lequel Thésée revient devrait être le même que celui dans lequel il est parti. Mais supposons maintenant que d'autres personnes ont suivi Thésée et ramassés toutes les planches qu'il a jeté par-dessus bord, et remontés toutes les parties dans un nouveau bateau, identique à la constitution physique de l'original. S'agit-il de la reconstruction du navire de Thésée, ou s'agit-il d'autre chose ?

Ces exemples montrent que des choix doivent être faits afin de déterminer les caractéristiques qui donnent son identité propre à un objet.

2.1.3.4/ DIFFÉRENTS TYPES D'ENTITÉS SPATIO-TEMPORELLES

Parmi les nombreux systèmes d'information géographique du domaine, quatre types d'entités spatio-temporelles (illustrées sur la figure 2.5) semblent recouvrir les différentes situations étudiées par la littérature du domaine de la modélisation spatio-temporelle [Cheylan, 2003] :

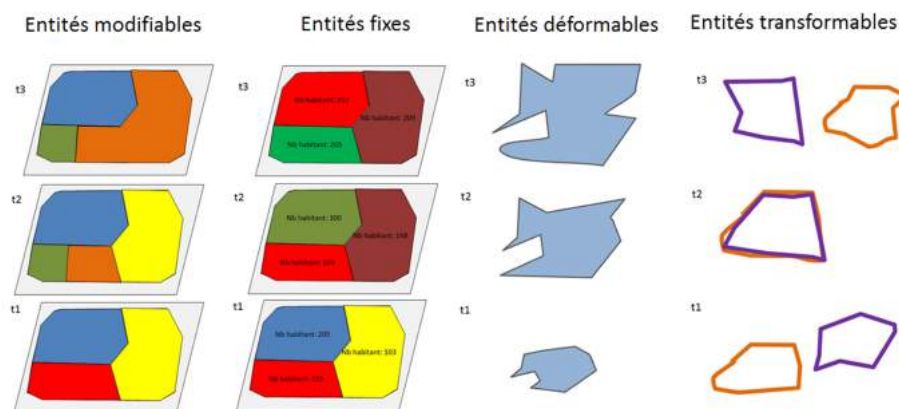


FIGURE 2.5 – Illustration des quatres types d'entités spatio-temporelles

- **Les entités fixes** : dans cette situation, les entités possèdent une représentation spatiale, une identité et des caractéristiques sémantiques. Toutefois, ni la géométrie ni la topologie de ces entités ne varient dans le temps. Seules les valeurs d'identité et des attributs éventuels associés à l'entité sont amenées à varier dans le temps.
- **Les entités génératives, ou modifiables** : Ce type d'entité existe au sein d'un espace dit "dynamique". Un espace dynamique est un espace partitionné au sein duquel des entités se divisent et se regroupent. Sur ce type d'entité, il est nécessaire d'associer les entités à l'aide d'une relation de filiation pour permettre de recomposer l'histoire d'un lieu. Les relations de filiation entre les différentes parcelles parent-enfants et les motifs spatiaux décrits par ces relations représentent une part importante de la sémantique du phénomène. Cependant, ces relations

sont souvent omises du fait de leur explicitation. Les entités génératives sont fréquemment utilisées sur des cadastres historicisés dont l'objectif est l'analyse des liens générationnels. Pour cela, la présence d'un graphe temporel liant les entités entre les générations est indispensable.

- **Les entités déformables** : ce type d'entité ne possède pas de composante d'identité. Dans ce cas, les entités de l'espace se déforment et se déplacent ainsi seule la composante spatiale, géométrique est amenée à varier.
- **Les entités transformables** : il s'agit ici du type d'entité le plus complexe, c'est pourquoi il est extrêmement rare de construire ce type d'entité faute d'outils de modélisation et de traitement. Les entités transformables sont à la croisée entre les entités déformables et les entités modifiables. Plus précisément, il s'agit d'entités qui se déplacent, se déforment mais peuvent également fusionner et se regrouper.

Dans ces travaux, l'approche proposée traite de la modélisation d'entités génératives ou modifiables.

Cette section a introduit un ensemble de notions liées à la modélisation spatio-temporelle. La section suivante présente plusieurs approches de modélisation d'entités spatio-temporelles.

2.2/ ÉTUDE DES APPROCHES DE MODÉLISATION SPATIO-TEMPORELLE

Un territoire dynamique est composé d'un ensemble de couches spatio-temporelles représentant l'état de ce territoire à différents instants. La notion de dynamique révèle le besoin d'étudier les différentes relations et causalités existantes entre ces différentes couches. Au sein d'un territoire, une couche spatio-temporelle peut être vue comme un objet uniforme appelée "snapshot". Toutefois, les approches "snapshot" nécessitent beaucoup d'analyses pour étudier la dynamique et offrent des résultats peu fiables. Dès lors, les approches basées sur les entités décomposent le territoire en un ensemble d'objets possédant chacun leur propre cycle de vie. En considérant que les entités évoluent sous l'influence d'éléments externes, les approches basées sur les événements ont vu le jour et s'intéressent aux transitions intervenues entre deux couches spatio-temporelles. Finalement, il apparaît que la modélisation de couches spatio-temporelles et de transitions sont complémentaires et nécessitent de définir et prendre en compte un certain nombre de relations pour mieux appréhender la dynamique spatio-temporelle. C'est pourquoi les graphes se sont progressivement imposés dans le cadre de la modélisation spatio-temporelle.

2.2.1/ APPROCHES BASÉES SUR LES "SNAPSHOTS"

La figure 2.6 présente le modèle connu sous le nom de "snapshot model" [Armstrong, 1988]. Dans ce modèle, chaque couche est une collection d'unités temporelles homogènes correspondantes à une couverture. Il montre l'état d'une répartition géographique à différents moments sans expliciter les relations temporelles entre les

couches. L'intervalle de temps entre deux couches peut varier sans qu'il n'y ait d'incidence si des changements se produisent dans le laps de temps d'une couche à l'autre.

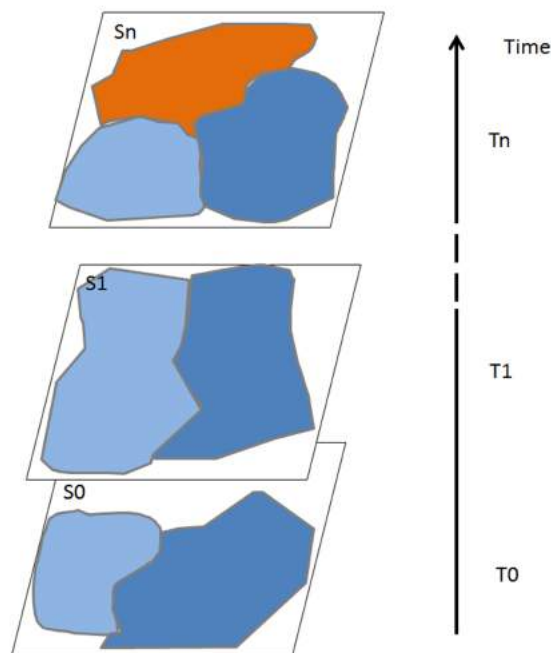


FIGURE 2.6 – Illustration du modèle snapshot

Dans la même lignée de modèle, il est possible de citer le modèle "Temporal Map Set" (TMS) [Beller et al., 1991] qui peut être vu comme une extension du "snapshot model". La conception du modèle TMS vise à modéliser les événements géographiques dans une zone définie. Les événements sont définis comme TMS binaires, c'est-à-dire en spécifiant si chaque cellule est dans ou hors de l'événement. Les approches par "snapshot" donnent toujours lieu à une grande quantité de duplication des données sur des propriétés inchangées dans l'espace et le temps. L'inconvénient majeur est la redondance des données ce qui augmente considérablement le volume de données à stocker. D'autre part, le changement spatial d'une entité est stocké de manière implicite dans cette approche. Par conséquent, l'analyse du changement des entités nécessite la comparaison de deux cartes consécutives impliquant de longs temps de calcul sans garantie de résultats conformes à la réalité car les changements impliquant de petites entités seraient très difficiles à analyser à cause de l'incertitude. Le risque d'incohérence des données analysées est alors un second problème des approches par "snapshot".

2.2.2/ APPROCHES BASÉES SUR LES ENTITÉS

La figure 2.7 illustre le modèle "Space-Time Composites" (STC) [Langran et al., 1988]. Dans ce modèle, le monde est représenté comme un ensemble d'objets spatialement homogènes et temporellement uniformes dans un espace 2D. En somme, chaque couverture existe au sein d'une couche uniforme. Il ne peut exister d'entités distinctes représentant la même couverture. Chaque couche uniforme est appelée "Composite".

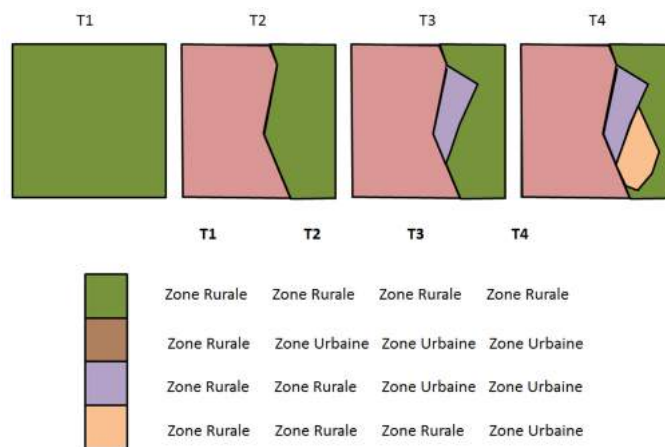


FIGURE 2.7 – Illustration du modèle Space-Time Composite

Chaque composite espace-temps décrit un parcours unique lors des changements temporels des attributs. Les composites espace-temps peuvent être obtenus par des superpositions temporelles de couches estampillées (snapshots). Un composite espace-temps décrit conceptuellement le changement d'un objet spatial à l'aide d'une période de temps. Les changements d'attributs sont enregistrés à des moments discrets, bien que leurs résolutions temporelles n'aient pas besoin d'être exactes. Le modèle STC permet d'enregistrer la temporalité sur les plus grandes unités communes de l'attribut, l'espace et le temps, mais il ne parvient pas à capturer la temporalité chez les attributs dans l'espace (mouvements). En outre, la mise à jour d'une base de données avec le modèle STC exige la reconstruction des unités STC. Par conséquent, les relations géométriques et topologiques entre les changements sur les unités STC et la base de données entière, intervenant à la fois sur les objets spatiaux et les tables d'attributs, doivent être réorganisées.

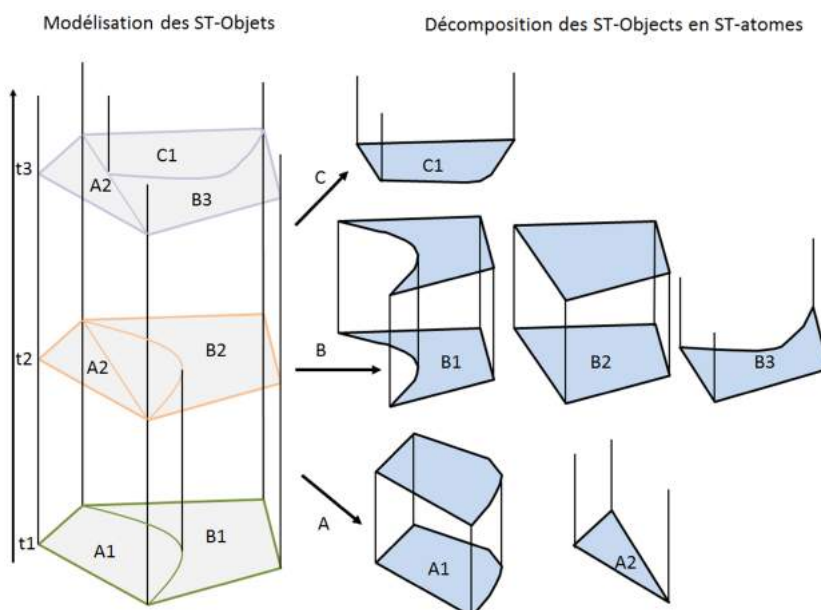


FIGURE 2.8 – Illustration du modèle ST-Objet [Worboys, 1994]

La figure 2.8 illustre le "spatio-temporal object model" (ST-Object model) [Worboys, 1994]. Ce modèle représente le monde comme un ensemble d'objets discrets constitués d'atomes spatio-temporels en incorporant une dimension temporelle orthogonale à l'espace 2D. Les atomes spatio-temporels (ST-atoms) sont les plus grandes unités homogènes pour lesquelles certaines propriétés sont maintenues à la fois dans l'espace et dans le temps. Un objet spatio-temporel peut changer tant dans l'espace que dans le temps, bien qu'il n'y ait aucun changement survenant dans chacun de ses atomes spatio-temporels. Par conséquent, le modèle ST-Object est capable d'enregistrer des changements dans les attributs d'un ST-objet à la fois dans les dimensions spatiales et temporelles, ensemble ou séparément, en projetant ses ST-atomes sur l'espace spatial et/ou temporel. Cependant, des changements graduels dans l'espace à travers le temps sont incapables d'être représentés dans ce modèle ST-Object puisque ses ST-atomes sont discrets. Bien que le modèle ST-Object soit semblable au modèle "snapshot" et au modèle STC, il ne représente que des changements soudains sur une période de temps discrète, et linéaire.

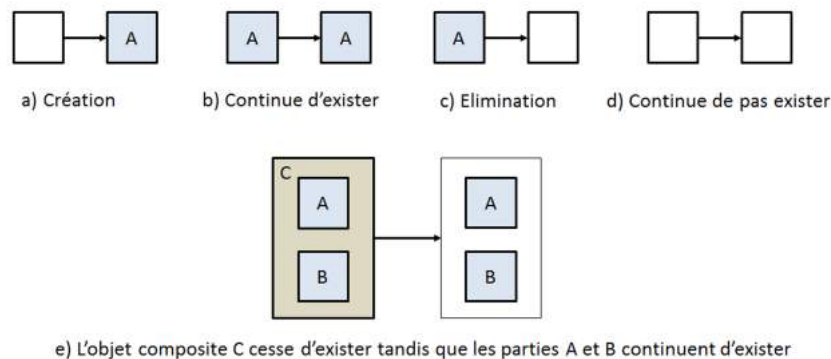


FIGURE 2.9 – Illustration du modèle basé sur l'identité

2.2.3/ APPROCHES BASÉES SUR LES ÉVÈNEMENTS

Le modèle ESTDM [Peuquet et al., 1995] permet d'organiser des données spatio-temporelles portant sur des changements de localisation. Comme le modèle TMS, le modèle ESTDM groupe des couches horodatées pour mettre en évidence des observations temporelles d'un événement unique dans une séquence temporelle. Cependant, ESTDM surpasse le modèle TMS en termes d'efficacité des données et de soutien pour l'analyse des schémas temporels et des relations, car ESTDM stocke les changements par rapport à un état antérieur, plutôt que de stocker une nouvelle instance complète à chaque changement et nécessiter des comparaisons entre elles.

La figure 2.10 illustre le modèle ESTDM. Un fichier d'entête contient des informations sur une thématique particulière (usage des terres d'une région géographique par exemple). Ce fichier pointe d'une part vers une carte de référence et d'autre part vers le premier et le dernier événement d'une liste d'événements qui se sont produits successivement sur cette carte. La carte de référence représente une capture instantanée d'un thème unique dans une région géographique. Grâce à une série d'événement, il est possible d'étudier la dynamique spatio-temporelle de cette thématique. Chaque événement est horodaté et

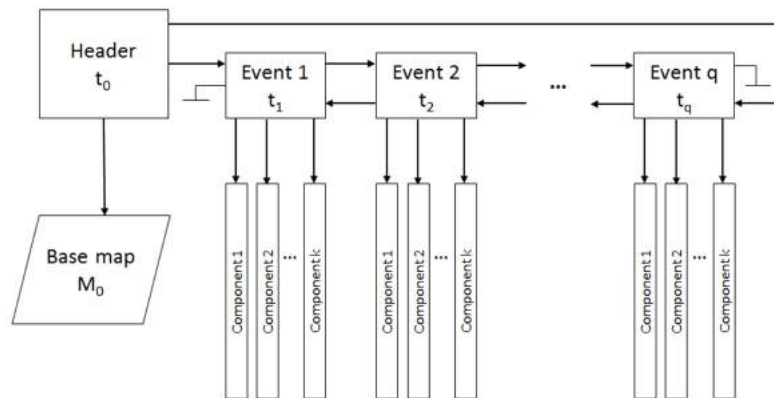


FIGURE 2.10 – Illustration du modèle ESTDM

associé à une liste de composant pour indiquer où les changements ont lieu. Un composant montre les changements intervenants sur des zones précises de la carte (des cellules du raster) et à un instant de temps particulier. Le modèle ESTDM est capable de supporter à la fois des requêtes spatiales et temporelles. Cependant, l'application du modèle ESTDM à un système vectoriel nécessite des modifications profondes notamment au niveau des composants. Les informations sur l'historique ou les transitions d'une entité ou d'un processus seront fragmentées si un changement intervient sur des objets spatiaux ou leur topologie. Des mécanismes sont donc nécessaires pour permettre aux composants de garder une trace des entités et des zones auxquelles ils font référence.

2.2.4/ APPROCHES BASÉES SUR LES GRAPHES

Le modèle ESTDM décrit un phénomène au travers d'une liste d'évènements, un nouvel évènement est créé en bout de liste à chaque fois qu'un changement est détecté. Toutefois, ce modèle prend en compte des données de type raster et les liens de causalité entre les évènements sont difficilement mis en évidence dans ce modèle car les objets ne sont pas clairement identifiés et définis comme dans l'approche vectorielle. Pour répondre à ce problème, le modèle des processus composites (figure 2.11) a pour objectif de représenter les liens entre les évènements ainsi que leurs conséquences. En outre, l'auteur avance que le modèle de données doit différencier ce qui est spatial, temporel et thématique. Le modèle des processus composites se base sur un langage permettant de décrire la sémantique liée à un phénomène du monde réel. Ce phénomène est assimilé à un processus composite, c'est-à-dire une suite de processus qui décrivent la dynamique du phénomène. Un processus composite est, par exemple, la trajectoire d'un bateau et peut se décomposer en 3 processus : stabilité, déplacement, rotation.

Enfin, le modèle de changement topologique basé sur les évènements a été proposé dans l'optique de représenter et raisonner à propos de la dynamique spatiale des phénomènes. Le modèle s'applique sur des objets planaires. Chaque objet est composé de plusieurs surfaces se superposant. Chaque surface peut-être incluse dans une autre ou contenir d'autres surfaces, en revanche les chevauchements ne sont pas admis. Le modèle utilise une structure en arbre pour représenter les relations topologiques entre

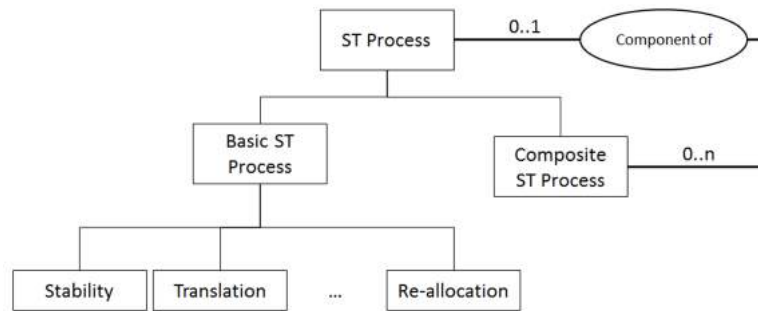


FIGURE 2.11 – Illustration du modèle des processus composites [Claramunt et al., 1996]

les objets. Chaque objet est associé à un arbre représentant l'imbrication spatiale des surfaces de l'objet à un instant T . Chaque arbre est ensuite lié à son prédécesseur et son successeur en liant chacune des surfaces entre les arbres formant ainsi un graphe de relation. Le graphe est ensuite analysé afin de déduire des motifs d'évolution spatiale telles que la division, la fusion, l'ajout ou la suppression d'une surface d'un objet (voir figure 2.12).

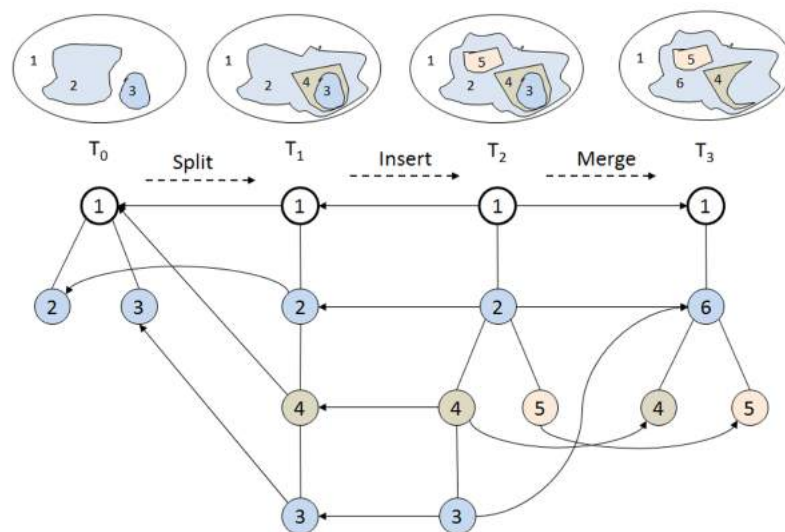


FIGURE 2.12 – Illustration du modèle de changements topologiques [Jiang et al., 2009]

2.2.5/ DISCUSSION

L'évolution d'une entité spatiale dans le temps peut être vue soit comme une succession d'états (ou représentations) de l'entité, soit comme une succession de transitions intervenant sur cette entité au cours du temps. Les modèles de la première proposition se basent sur des approches continues ou discrètes. Des exemples de modèles sont le modèle snapshot [Armstrong, 1988], le modèle Space-Time Composites (STC) [Langran et al., 1988], le modèle Spatio-temporal Object [Worboys, 1994]. L'inconvénient de ces modèles est qu'ils ne représentent que des changements soudains au travers

desquels il est difficile d'identifier des processus tels que le changement ou le mouvement d'une entité de l'environnement géographique. D'autres modèles basés sur les approches continues ou discrètes ont été proposés pour gérer les changements liés à l'identité [Hornsby et al., 2000] ou encore les changements sur les relations topologiques étudiées à l'aide de matrices d'intersection [Egenhofer et al., 1992]. Toutefois, l'analyse des causes du changement nécessaire à l'étude des phénomènes est difficile à déduire en utilisant ce type de modélisation. Par conséquent, cette première approche de modélisation ne permet pas une analyse complète de l'évolution. Pour pallier à ce problème, l'approche de modélisation basée sur les événements et processus a peu à peu vu le jour. Cette approche considère que les entités spatiales évoluent sous l'impulsion d'un événement ou d'un processus et dont l'objectif est d'analyser les causes et les conséquences. Parmi ces modèles, il est possible de citer le modèle Event-Based Spatiotemporal Data Model (ESTDM) [Peuquet et al., 1995], les processus composites [Claramunt et al., 1996] ou encore le modèle de changement topologique basé sur les événements [Jiang et al., 2009]. Ces deux derniers modèles se basent sur les relations temporelles et spatiales afin de déterminer des motifs spatiaux illustrant la dynamique des entités. Toutefois, ces approches ne prennent pas en compte la composante d'identité. Cette limitation est dépassée dans les travaux de [Del Mondo et al., 2013] permettant l'analyse d'entités spatio-temporelles au travers de motifs basés sur des contraintes spatiales et identitaires. En outre, le papier introduit la relation de filiation déclinées en deux catégories :

- Les continuations : qui représente le lien spatio-temporel entre deux états consécutifs d'une entité évoluant sans que l'identité ne soit affectée.
- Les dérivations : qui représente le lien spatio-temporel entre deux états consécutifs d'une entité évoluant et dont l'identité varie entre les deux états.

Malgré une représentation explicite de la dynamique spatiale des entités dans les modèles basés sur les graphes de relation, ces approches souffrent du manque de sémantique pour permettre de qualifier les motifs génériques définis et aboutir à la découverte de phénomènes au sein du jeu de données. Pour cela, il est nécessaire de représenter les données au sein d'un support riche capable de prendre en compte le contexte et la connaissance liée au domaine de l'application ainsi que de fournir des outils de raisonnement pour permettre d'inférer de nouvelles connaissances à partir de celles déjà existantes.

Dans cet optique, nous envisageons les logiques de description comme support de modélisation spatio-temporelle pour répondre aux besoins mis en lumière. Les logiques de description fournissent davantage d'expressivité que l'algèbre relationnel et fournissent également des outils de raisonnement. En outre, les logiques de description sont utilisées pour de nombreuses applications, parmi lesquelles la représentation d'ontologie dont l'objectif est la définition des éléments d'un domaine de connaissance. Les ontologies possèdent l'avantage de fournir un langage lisible en machine pour permettre la représentation des connaissances et appliquer des raisonnements.

2.3/ LOGIQUES DE DESCRIPTION ET ONTOLOGIES

Nous venons d'étudier les approches majeures de la modélisation spatio-temporelle. Toutes ces approches se basent sur l'emploi de bases de données traditionnelles. Toutefois, il est à noter que les approches les plus récentes tendent à représenter l'information spatio-temporelle à l'aide d'un graphe. Dans le cadre de cette thèse, nous adhérons à l'usage de graphe dans le cadre de la modélisation. Cependant, notre approche se distingue par la volonté de sémantiser le graphe afin d'augmenter les capacités de représentation du système ainsi que de fournir des mécanismes d'inférences valides capables de faire émerger des connaissances à partir de celles déjà existantes. Pour cela, nous proposons d'étudier les bases de connaissances comme support de l'information spatio-temporelle. Dans cette section, nous introduirons les bases de connaissances ainsi que les logiques descriptives qui font autorité dans le cadre de la création de bases de connaissances. Enfin, nous présenterons les différents types de raisonnement qui peuvent être appliqués sur une base de connaissances en logique de description.

2.3.1/ BASE DE CONNAISSANCES

Une base de connaissances regroupe des connaissances spécifiques à un domaine spécialisé donné, sous une forme exploitable par un ordinateur grâce à sa définition basée sur les logiques de description [Staab et al., 2013], [Baader et al., 2003]. 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 base de connaissance en logique descriptive se compose d'une TBox représentant la terminologie du domaine et une ABox qui déclare les assertions et instances de ce domaine.

2.3.1.1/ TBOX : TERMINOLOGIE

La terminologie a pour objet l'étude théorique des dénominations des objets ou concepts utilisés pour décrire un domaine particulier. Pour cela, la TBox d'une base de connaissance s'appuie sur le concept d'axiome terminologique qui peut se résumer à deux types de formules décrites par la formule 2.1 :

$$C \sqsubseteq D \text{ ou } C \equiv D \quad (2.1)$$

La première formule définit que toute instance de la classe C appartient également à la classe D tandis que la seconde indique que les concepts C et D sont équivalents, c'est-à-dire que toute instance de la classe C appartient également à la classe D et que toute instance de la classe D appartient également à la classe C . En se basant sur ces deux formules, il est alors possible de construire des définitions pour décrire des concepts complexes. Par exemple, le concept de *Tissus Urbain Discontinu* peut se décrire la façon suivante :

$$TissusUrbainDiscontinu \equiv ZoneUbaine \sqcap Forêt \quad (2.2)$$

Ainsi, une parcelle de tissu urbain discontinu représente un territoire sur lequel coexiste des zones urbaines ainsi que des forêts. La terminologie, ou TBox peut ainsi être décrite comme un ensemble de définitions tel qu'un concept n'apparaît jamais deux fois du côté gauche. En d'autres termes, chaque concept n'est défini qu'une seule fois dans la base de connaissances.

2.3.1.2/ ABox : DESCRIPTION DU MONDE

La deuxième composante d'une base de connaissances est la ABox et permet une *description du monde*. La ABox permet de définir des individus en spécifiant leurs noms, les concepts auxquels ils appartiennent, et leurs relations avec d'autres individus. L'instanciation d'un concept se définit sous la forme $C(a)$ avec C le concept et a l'instance de ce concept. Par exemple :

$$\text{Région}(\text{Bourgogne}) \quad (2.3)$$

La formule ci-dessus définit que la *Bourgogne* est une instance de la classe *Région*

De façon similaire, l'instanciation d'une relation se définit sous la forme $R(b, c)$ avec R la relation et b et c les instances liées par cette relation. Par exemple :

$$\text{sontAdjacentes}(\text{Bourgogne}, \text{Champagne} - \text{Ardenne}) \quad (2.4)$$

La formule définit à présent que les instances *Bourgogne* et *Champagne – Ardenne* sont adjacentes. Les deux exemples cités précédemment sont appelés des faits. L'ensemble des faits d'une base de connaissances offre une description du monde modélisé au sein du système.

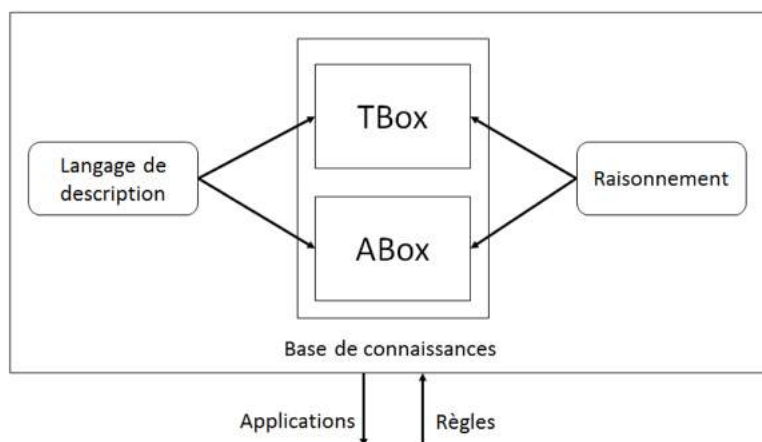


FIGURE 2.13 – Illustration d'une base de connaissances [Baader et al., 2003]

Nous venons de voir qu'une base de connaissances (\mathcal{K}) se compose d'une terminologie, ou TBox (\mathcal{T}_{box}) ainsi qu'un ensemble d'individus (ABox : (\mathcal{A}_{box})) définis dans le respect de cette terminologie tel que $\mathcal{K} = (\mathcal{T}_{box}, \mathcal{A}_{box})$. Comme illustré par la figure 2.13, une base de connaissances se base sur un langage de description capable de

représenter les classes et les relations aussi bien pour la TBox que la ABox. D'autre part, une base de connaissances se caractérise par sa capacité à supporter des tâches de raisonnement telles que le test de subsomption, d'instanciation ou de satisfiabilité. Ces opérations d'inférence s'appliquent aussi bien sur la TBox que sur le ABox. L'annexe B présente d'une part, les familles des logiques de description comme langage formel pour représenter les connaissances et, d'autre part, les raisonnements applicables sur ces connaissances qui constituent un apport majeurs des logiques de description par rapport aux autres langages de représentation des connaissances [Baader et al., 2003].

OWL est un langage de représentation des connaissances permettant la définition d'ontologie. Depuis 2004, OWL est une recommandation du consortium W3C [McGuinness et al., 2004]. Il est basé sur les travaux ayant eu lieu dans le domaine des logiques de description que nous avons étudié à la section précédente. OWL hérite des logiques de description la possibilité de mettre en place des processus de vérification de la consistance et de la cohérence ainsi que de déduction de connaissances à partir des connaissances déjà existantes. Les ontologies sont des outils largement utilisés pour permettre de réaliser des bases de connaissances.

2.3.2/ DÉFINITION ET USAGE D'ONTOLOGIE EN INFORMATIQUE

Au sens informatique, une ontologie représente un graphe structuré de concept pour lequel les relations peuvent être des relations sémantiques ou des relations de subsomption. L'objectif premier d'une ontologie est de modéliser un ensemble de connaissance dans un domaine donné, c'est pourquoi l'ontologie est souvent assimilée à une base de connaissances. Pourtant, il existe une différence fondamentale entre une ontologie et une base de connaissance. Une ontologie fournit la structure de base autour de laquelle une base de connaissances peut être construite. Une ontologie fournit un ensemble de concepts et de termes pour décrire un certain domaine, tandis qu'une base de connaissances utilise ces termes pour représenter ce qui est vrai sur un monde réel ou hypothétique [Swartout et al., 1999].

Il existe de nombreuses définitions de l'ontologie, toutefois une des définitions faisant autorité dans le domaine est celle de Gruber [Gruber, 1995] :

Une ontologie est la spécification explicite d'une conceptualisation d'un domaine de connaissance.

Cette définition fait intervenir les notions de *spécification* et de *conceptualisation* qui nécessitent d'être explicitées davantage pour une meilleure compréhension. Une conceptualisation est une vue simplifiée et abstraite du monde que l'on souhaite représenter sous certains aspects. En somme, une conceptualisation est un choix quant à la manière de décrire un domaine particulier. La spécification, quant à elle, représente la description formelle de cette conceptualisation pour la rendre utilisable par un ordinateur [Studer et al., 1998]. Ainsi, de façon plus détaillée, une ontologie peut se définir comme un vocabulaire commun pour le partage de l'information d'un domaine, incluant des définitions lisibles en machine des concepts de base de ce domaine et des relations entre ces concepts. Sur les bases de cette définition, il existe une confusion fréquente entre vocabulaire contrôlé, taxonomie, thésaurus et ontologie [Gilchrist, 2003] du fait de définitions partiellement identiques. En effet, chacun d'entre eux permet de structurer, classifier, modéliser et représenter des concepts et relations d'un domaine particulier. Également, chacune de ces approches tente de parvenir à un accord pour que les mêmes

termes soient employés pour les mêmes usages pour une communauté donnée. Toutefois, il existe des différences majeures entre ces approches en fonction du degré de signification des termes spécifiés, de la notation ou du langage utilisé pour spécifier la signification des termes, et en fonction des finalités d'usage de chacun d'eux.

Le **vocabulaire contrôlé** peut n'avoir aucune signification spécifiée, par exemple, il peut se résumer en une liste de termes d'usage sur lesquels les gens d'un domaine se sont mis d'accord et pour lesquels la signification est compréhensible. Ou bien un vocabulaire contrôlé peut offrir des définitions très détaillées pour chacun des termes.

La **taxonomie** permet de donner plus de sens aux termes spécifiés grâce à la relation de subsomption (ou relation hiérarchique). Historiquement, la signification d'une relation de subsomption peut se décrire comme *est un type de*, par exemple : *un chien est un type d'animal*. Cependant, de nos jours, le concept de taxonomie réfère à d'autres types de hiérarchie avec d'autres significations pour la relation, comme la relation *fait partie de*, par exemple : *la nièvre fait partie de la Bourgogne*. Les taxonomies définies de manière vague ne permettent pas d'identifier clairement le type de relation de la hiérarchie. Cependant, les taxonomies possédant une définition très claire des différents types de lien de la hiérarchie se rapprochent des ontologies.

Le **thésaurus** possède les mêmes capacités que la taxonomie pour définir des hiérarchies à l'aide de relations, mais se distingue par sa capacité à fournir d'autres types de relations que les relations hiérarchiques. Cependant, ces relations peuvent n'avoir aucune signification autre que : *il existe une relation entre ces deux termes*.

Le mot **ontologie** est souvent utilisé pour faire référence à des choses diverses et variées comme des glossaires, des thésaurus, des taxonomies, des bases de connaissances, etc. Une ontologie formelle est un vocabulaire contrôlé exprimé dans un langage de représentation d'ontologie. Ce langage possède une grammaire permettant d'utiliser les termes de vocabulaire pour exprimer des choses possédant une signification au sein d'un domaine particulier. La grammaire contient des contraintes formelles permettant de spécifier ce qu'est une assertion, une déclaration correctement formée, une requête, etc. Ainsi cette grammaire définit comment les termes du vocabulaire contrôlé peuvent être utilisés ensembles. Les taxonomies et les thésaurus peuvent être vu comme des langages simples pouvant donner lieu à une grammaire. Cependant, de manière générale, ce n'est pas le cas.

En somme, ce sont la combinaison de l'ensemble riche de relations et de la grammaire du langage de représentation qui confère à l'ontologie son caractère unique.

2.3.3/ LE LANGAGE DE REPRÉSENTATION DES ONTOLOGIES : OWL

Nous venons d'étudier la notion d'ontologie. Les ontologies sont utilisées au sein de plusieurs communautés informatiques comme l'intelligence artificielle, le génie logiciel, l'informatique biomédicale, etc. mais c'est au sein du Web Sémantique que naîtra le langage OWL qui fait aujourd'hui autorité dans le domaine de la représentation d'ontologie. Le Web Sémantique est un mouvement collaboratif guidé par le W3C pour favoriser l'échange des données et aider à l'émergence de connaissance en s'appuyant sur celles déjà présentes sur internet.

Tim Berners-Lee décrit le Web Sémantique comme "un modèle qui permet aux données d'être partagées et réutilisées entre plusieurs applications, entreprises et groupes d'uti-

lisseurs” [Van Rijmenam, 2014]. A l’origine, le Web Sémantique fut créé dans l’optique d’orienter l’évolution du web actuel pour permettre aux utilisateurs de trouver, partager et combiner l’information plus facilement et sans intermédiaire. Bien que cet objectif soit resté intact, le terme de “Web Sémantique” est aujourd’hui également souvent utilisé pour désigner les formats et technologies qui ont été conçus pour permettre son émergence. La liste des technologies du Web Sémantique est fréquemment illustrée par une architecture appelée “pile du Web Sémantique” (voir figure 2.14). Dans ces travaux, nous ne donnerons pas une description détaillée de chacune de ces technologies et nous contenterons des brèves définitions données ci-après :

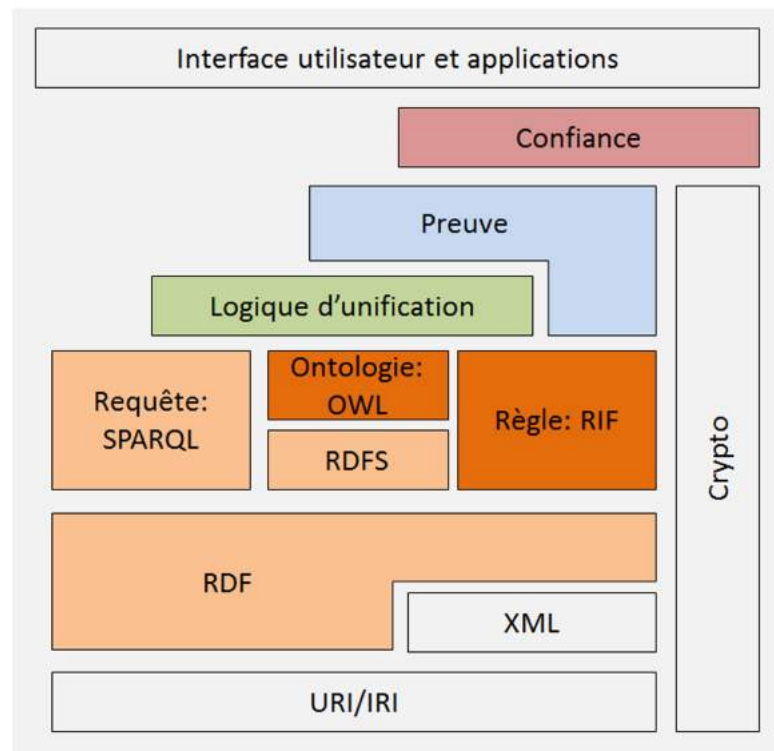


FIGURE 2.14 – Pile du Web Sémantique

- Les **URI/IRI** permettent d’identifier de manière unique les ressources du Web Sémantique.
- Le **XML** fournit une syntaxe élémentaire, pour la structure du contenu dans les documents, mais il ne décrit pas la sémantique du document. XML n’est pas à l’heure actuelle une composante nécessaire des technologies du Web Sémantique dans la plupart des cas.
- Le **RDF** est un langage simple destiné à décrire de façon formelle les ressources web et leurs métadonnées sous forme de triplets de la forme “sujet - prédicat - objet”. Un modèle basé sur RDF peut être représenté à travers plusieurs syntaxes d’échanges, par exemple, RDF/XML, N3, Turtle, et RDFa. RDF est une norme fondamentale considérée comme le langage de base du Web Sémantique. Il permet la mise en oeuvre du web de données qui consiste à lier et structurer l’information sur internet pour accéder simplement à la connaissance.
- **RDF Schema** étend le RDF et son vocabulaire pour permettre une représentation

de structures conceptuelles et d'ontologies simples avec description hiérarchique des concepts et des propriétés.

- **OWL** ajoute plus de vocabulaire pour décrire les propriétés et les classes : comme avec les relations entre les classes, la cardinalité, l'égalité, le typage des propriétés, les caractéristiques de propriétés (par exemple la symétrie), etc. En outre, OWL permet d'effectuer des raisonnements.
- **SPARQL** est un langage de requête et un protocole qui permettra de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles dans le Web à travers l'Internet.
- **RIF** est le langage d'expression de règles et décrit les relations qui ne peuvent l'être en OWL.
- La **logique d'unification** représente les raisonnements logiques permettant d'inférer de nouveaux faits ou de vérifier la consistance de l'ontologie.
- La **preuve** donne l'explication des raisonnements logiques fournis par la couche "logique d'unification".
- La **cryptographie** protège les données RDF grâce à un chiffrement et valide la source des faits par une signature digitale des données RDF.
- La **confiance** assure l'authenticité des sources et la fiabilité des faits dérivés.

Si l'évolution du web actuel vers le Web Sémantique semble toujours attendue bien que n'ayant pas encore réellement eu lieu [Berners-Lee et al., 2001] et restant une idée largement inexploitée [Shadbolt et al., 2006], les technologies du Web Sémantique ont toutefois réussi à s'imposer dans le domaine de l'ingénierie des connaissances grâce notamment au standard OWL. Dans ces travaux, les technologies du Web Sémantique sont utilisées afin de définir une ontologie comme modèle de données spatio-temporelles. Ainsi, les objectifs de ces travaux ne sont que partiellement conjoints à ceux du Web Sémantique, notre intérêt pour le Web Sémantique étant guidé principalement par l'usage des technologies qui y sont associées. La figure 2.15 illustre la relation entre les ontologies, le Web Sémantique et les métadonnées.

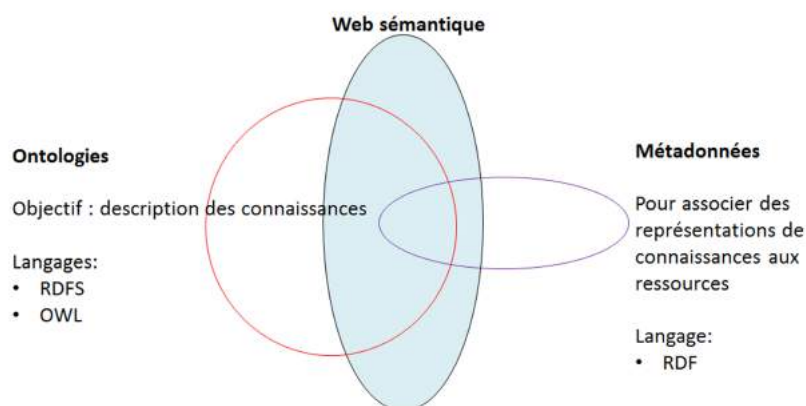


FIGURE 2.15 – Relation entre le Web Sémantique, les ontologies et les métadonnées

Parmi les technologies du Web Sémantique, notre attention sera portée essentiellement sur le langage de représentation des connaissances OWL, basé sur le modèle de données de RDF. En outre, OWL est un standard informatique mettant en oeuvre certaines logiques de description permettant ainsi aux outils comprenant OWL de travailler avec ces données ainsi que de fournir des raisonnements tels que la vérification de la

consistance de l'ontologie ainsi que l'inférence de nouvelles connaissances à partir de celles déjà existantes. Le langage OWL est composé de trois sous-langages : OWL-Lite, OWL-DL et OWL-Full. OWL-Lite correspond essentiellement à la famille *SHIF*, alors que OWL-DL (DL signifie "Description Logic") correspond essentiellement à la famille *SHOIN*. Plus précisément, il s'agit des familles *SHIF(D)* et *SHOIN(D)* car OWL distingue deux types de rôles. Les rôles qui lient deux individus (appelés *Object Property*), comme ceux que nous avons vu dans tous les exemples jusqu'à présent, et les rôles qui associent un individu avec un littéral (appelés *Datatype Property*, d'où le *(D)*). Un littéral est une valeur possédant un type donné comme *Integer*, *String*, *Boolean*, etc. Les sous-langages OWL-Lite et OWL-DL ne sont pas des extensions de RDF, dans le sens qu'un triplet RDF n'est pas nécessairement valide dans ces deux sous-langages. C'est pour cette raison que le sous-langage OWL-Full a été ajouté, et comprend tout OWL-DL, avec en plus tout RDF. En revanche, OWL-Lite est un sous-langage de OWL-DL. Enfin, OWL-DL est la version décidable du langage OWL. Il permet donc la réalisation de raisonnements, contrairement à OWL-Full qui est beaucoup trop expressif.

Depuis 2009, le consortium a officiellement lancé OWL 2, qui se distingue de la première version par un pouvoir expressif augmenté et une élimination de la décomposition en trois sous-langages. On parle plutôt de profils, en éliminant de OWL 2 certains éléments du langage, limitant ainsi son pouvoir expressif. Ainsi, OWL-Full n'existe plus, alors que OWL-Lite et OWL-DL peuvent être considérés comme des profils de OWL 2. Depuis 2012, OWL 2 est recommandé par le W3C et permet l'expressivité de la logique *SROIQ(D)* [Staab et al., 2013], [Baader et al., 2003]. L'évolution de l'expressivité depuis les logiques descriptives les moins expressives jusqu'aux logiques les plus complexes utilisées dans le domaine du Web Sémantique en lien avec les sous-langages d'OWL a été présentée sur la figure B.1.

Constructeurs OWL	Syntaxe	Exemple
intersectionOf	$C1 \sqcap C2$	<i>Humain</i> \sqcap <i>Homme</i>
unionOf	$C1 \sqcup c2$	<i>Docteur</i> \sqcup <i>Avocat</i>
complementOf	$\neg C$	\neg <i>Homme</i>
oneOf	$\{a_1 \dots a_n\}$	$\{, \textit{Georges}, \textit{Patrick}, \textit{Sarah}\}$
allValueFrom	$\forall R.C$	\forall <i>possèdeEnfant.Docteur</i>
someValueFrom	$\exists R.C$	\exists <i>possèdeEnfant.Avocat</i>
hasValue	$\exists R.a$	\exists <i>citoyenDe.France</i>
minCardinality	$\geq nR$	≥ 2 <i>possèdeEnfant</i>
maxCardinality	$\leq nR$	≤ 1 <i>possèdeEnfant</i>
inverseOf	R^-	<i>possèdeEnfant</i> ⁻

TABLE 2.1 – Les constructeurs OWL

Comme vu précédemment, OWL-DL et OWL-Lite se basent respectivement sur les logiques *SHOIN* et *SHIF* qui sont une augmentation de l'expressivité de la logique *SH*, elle-même basée sur *S* à laquelle la gestion des hiérarchies de rôles a été ajoutée. Les tableaux 2.1 et 2.2 présentent les constructeurs du langage OWL et les axiomes du langage OWL 2 ainsi que leur correspondance en logique de description.

Sur la base de ces axiomes et constructeurs plusieurs raisonnements d'inférence sont possibles en OWL et dérivent des principes de raisonnement énoncés en logique de description. L'annexes D détaillent toutes les règles d'implications de RDFS et OWL, et donne plusieurs exemples d'inférence en OWL sur la TBox ainsi que sur la ABox. Outre ces implications propres au langage OWL, il est également possible de définir ses propres

Axiome OWL	Syntaxe	Exemple
subClassOf	$C1 \sqsubseteq C2$	<i>Humain</i> \sqsubseteq <i>Animal</i> \sqcap <i>Bipède</i>
equivalentClass	$C1 \equiv C2$	<i>Homme</i> \equiv <i>Humain</i> \sqcap <i>Mâle</i>
subPropertyOf	$R1 \sqsubseteq R2$	<i>possèdeFille</i> \sqsubseteq <i>possèdeEnfant</i>
equivalentProperty	$R1 \equiv R2$	<i>Coût</i> \equiv <i>Prix</i>
disjointWith	$C1 \sqsubseteq \neg C2$	<i>Homme</i> $\sqsubseteq \neg$ <i>Femme</i>
sameAs	$a_1 \equiv a_2$	<i>Corse</i> \equiv <i>île de Beauté</i>
differentFrom	$a_1 \sqsubseteq \neg a_2$	<i>Roméo</i> $\sqsubseteq \neg$ <i>Juliette</i>
transitiveProperty	$Tr(R)$	$Tr(Contient)$
functionalProperty	$\top \sqsubseteq (\leq 1R)$	$\top \sqsubseteq (\leq 1estMèreDe)$
inverseFunctionalProperty	$\top \sqsubseteq (\leq 1R^-)$	$\top \sqsubseteq (\leq 1aPourMère^-)$
symmetricProperty	$R \equiv R^-$	<i>estMariéAvec</i> \equiv <i>estMariéAvec</i> ⁻

TABLE 2.2 – Les axiomes OWL 2

règles de la forme :

$$\text{antécédent} \Rightarrow \text{conséquent} \quad (2.5)$$

L'antécédent établit une liste de conditions. Si toutes les conditions sont vraies alors le conséquent l'est aussi. Les règles sont un moyen simple de représenter une connaissance, et sont une simplification radicale de la logique du premier ordre pour laquelle il est relativement facile de mettre en œuvre des moteurs d'inférence qui peuvent traiter les conditions et déduire de bonnes conclusions. Plusieurs outils existent afin de définir des règles sur une ontologie. Les plus connus sont Rule Interchange Format (RIF) et Semantic Web Rule Language (SWRL). RIF est une recommandation du W3C depuis 2010 et a été conçue pour fournir un format d'échange de règles entre des systèmes de règles [Kifer et al., 2010]. SWRL, quant à lui, est une extension de OWL. Bien que n'étant pas admis comme un standard, il est cependant très réputé et utilisé pour sa simplicité. D'autre part, SWRL [Horrocks et al., 2004] possède quelques implémentations au sein des triplestores qui ont émergé, contrairement à RIF qui ne possède pas d'implémentation au sein de triplestores supportant OWL. Outre ces deux outils, SPARQL Update [Seaborne et al., 2008] au travers de la fonction *INSERT* permet également d'établir une forme de règle. La fonction *INSERT* se base sur SPIN (SPARQL Inferencing Notation) [Knublauch et al., 2009] pour permettre l'expression de règles en SPARQL. SPARQL [Prud'Hommeaux et al., 2008] est un standard du W3C qui s'est largement imposé et est implémenté dans de nombreuses technologies industrielles supportant RDF et OWL ainsi que de nombreuses API. De ce fait, les règles établies en SPARQL bénéficient de la puissance du langage qui offre les meilleurs temps de réponses pour interroger une ontologie. D'autre part, il n'est pas nécessaire de maîtriser d'autres langages de règles. Toutefois, il est intéressant de noter que contrairement à RIF et SWRL, les règles SPARQL s'établissent sous la forme :

$$INSERT \{ \text{conséquent} \} [WHERE \{ \text{antécédent} \}] \quad (2.6)$$

Avec le *conséquent* et l'*antécédent* constitués d'un ou plusieurs triplets.

Pour toutes les raisons exposées précédemment, dans ces travaux, nous opterons pour l'utilisation de SPARQL Update comme support de règle sur une ontologie OWL.

2.4/ L'APPROCHE WEB SÉMANTIQUE ET L'APPROCHE CLASSIQUE

Dans la littérature, nombreuses sont les bases de données relationnelles proposant une extension spatiale. Cependant, il n'existe aucune base de données relationnelles proposant une extension spatio-temporelle à cause de la structure et des comportements complexes de ce type de données. Par conséquent, plusieurs modèles conceptuels spatio-temporels ont été proposés dans la littérature afin de servir d'intermédiaire au stockage des données dans des bases de données relationnelles. Les plus connus d'entre eux ont été étudiés à la section précédente bien qu'aucun ne soit jamais devenu un standard ni un modèle officiel de stockage pour l'information et l'interrogation de données spatio-temporelles. Dans le cadre de nos travaux, nous avons choisi de baser notre modèle spatio-temporel sur les technologie du Web Sémantique impliquant de nombreuses différences concernant le mode de stockage, l'interprétation des données, la flexibilité du modèle, etc. Dans cette section, nous étudierons plus en détails les différences de modélisation selon l'approche graphe sémantique ou relationnelle.

2.4.1/ COMPARATIF DES CARACTÉRISTIQUES INTRINSÈQUES DES SYSTÈMES EN MONDE OUVERT ET EN MONDE FERMÉ

Plus particulièrement, l'approche relationnelle et l'approche sémantique se distinguent par deux visions distinctes des données stockées dans le système. L'approche relationnelle est soumise à l'hypothèse du monde fermé, par conséquent, tous les faits qui ne sont pas inclus dans la base de données sont considérés comme faux à moins de démontrer le contraire en un temps fini. En somme, en l'absence d'information sur une affirmation, celle-ci est considérée comme fautive. L'approche sémantique, quant à elle, est soumise à l'hypothèse du monde ouvert qui définit qu'il n'est pas possible de déduire qu'une affirmation est fautive du simple fait que celle-ci n'a pas été explicitement définie comme vraie dans le système d'information. Ces hypothèses engendrent un certain nombre de différences dans la façon de concevoir les données du système d'information. Dans cette partie, un comparatif des deux approches est établi dans le cadre de la conception d'un système d'information.

2.4.1.1/ COMPLÉTUDE DES INFORMATIONS

La conception d'un système d'information consiste à stocker des données de manière organisée afin de modéliser un domaine. Les informations stockées sont dites "complètes" ou "incomplètes" et impactent directement les raisonnements pouvant être appliqués au système. L'approche relationnelle est reconnue comme étant complète [Motro, 1989]. Les informations manquantes sont souvent gérées à l'aide de l'assertion de la valeur "null" définie en SQL. Dans la littérature, cet état de fait est connu sous le nom "d'hypothèse du domaine de fermeture" [Hustadt et al., 1994] et définit qu'il ne peut y avoir d'autres objets dans l'univers que ceux désignés par des constantes au sein de la base de données. Par exemple, dans une application de voyage proposant des vols entre deux destinations à une certaine date. Si l'utilisateur cherche un vol entre Paris et New-York le 15 Janvier 2014 et que l'application ne renvoie aucun résultat alors la conclusion est qu'il n'existe aucun vol entre Paris et New-York pour ce jour. Cependant, la complétude du monde fermé n'est pas toujours réaliste en pratique car la plupart des bases de données possèdent

des informations susceptibles d'être incomplètes [Levesque, 1984], [Motro, 1989]. Un des points principaux de l'hypothèse du monde ouvert est de travailler sur des informations incomplètes. Par conséquent, le monde ouvert autorise que les attributs d'un objet spécifique ou d'une instance soient incomplets ou partiellement connus. L'hypothèse du monde ouvert adopte "l'hypothèse du domaine d'ouverture" [Hustadt et al., 1994] qui définit qu'il peut y avoir dans l'univers plus d'objets que ceux définis dans la base de connaissance à moins qu'une contrainte dans la base de données ne l'interdise. Prenons l'exemple d'une application spatio-temporelle où l'utilisateur sélectionne plusieurs parcelles d'un territoire (sur la même couche temporelle ou sur une couche temporelle différente) et le système fournit en retour la relation existante entre les parcelles. Dans le cas où le système ne renvoie aucune relation entre des parcelles cela ne signifie pas nécessairement que l'évolution de ces parcelles n'est pas liée d'une façon ou d'une autre, c'est pourquoi à défaut de résultats, dans un système soumis à l'hypothèse du monde ouvert, la réponse sera "je ne sais pas". La notion de complétude des informations influe directement sur la manière de concevoir le schéma du système de gestion des données qu'il s'agisse d'une base de données relationnelle ou d'une ontologie.

2.4.1.2/ FLEXIBILITÉ DU SCHÉMA

Puisque la connaissance est une représentation de la compréhension d'un domaine donné par les praticiens et les experts, il est donc important que ces deux types d'utilisateurs soient actifs dans sa collecte, son organisation (structure) et son usage. De plus le caractère incomplet des approches en monde ouvert suscite le besoin d'étendre les informations à modéliser dans la suite du cycle de vie de l'application. La flexibilité du schéma devient alors un enjeu face au besoin d'adaptation des modèles. Des modifications sur le schéma d'un système relationnel nécessite de revoir totalement l'architecture de la base de données, par conséquent, l'approche relationnelle n'est pas fondamentalement extensible mais permet de définir précisément les limites d'un domaine à modéliser. L'approche Web Sémantique, quant à elle, a été créée afin de réutiliser des ontologies existantes et pour être extensible. Pour ce faire, l'approche Web Sémantique possède la particularité de dissocier la partie terminologique appelée TBox et la partie assertionnelle appelée ABox. Ainsi le schéma peut évoluer indépendamment des instances offrant une meilleure souplesse pour la conception et la mise à jour de l'application. De plus, les schémas de diverses ontologies peuvent être fusionnés pour étendre les capacités de représentation de l'application. Cependant, la réutilisation et la fusion d'ontologie soulève le problème de l'unicité des entités représentées lorsqu'une même entité est représentée dans deux ontologies avec des noms différents.

2.4.1.3/ GESTION DE L'UNICITÉ DES ENTITÉS

La représentation d'entités du monde réel soulève le besoin d'identification pour permettre aux utilisateurs d'établir le lien entre les données fournies par le système et leurs correspondances dans le monde réel. L'identification se fait via l'utilisation de labels ou de noms. L'hypothèse du nom unique permet de définir que des noms différents au sein du système font référence à des entités différentes dans le monde réel [Hustadt et al., 1994], [Levesque, 1984]. Cette hypothèse est souvent liée à l'hypothèse du monde fermé car les données étant réputées complètes, le système n'est pas supposé être étendu par la suite en fusionnant des données d'une autre base de données relationnelle. Contrairement à

l'approche relationnelle, OWL permet de définir des labels différents pour représenter les mêmes objets, car les informations sont admises comme étant incomplètes dans ce type d'application, la fusion de graphe est, pas conséquent, fréquemment mise en œuvre. Ainsi il est fréquent d'observer des mêmes noms pouvant référer à des entités différentes dans des jeux de données. De même, des noms différents peuvent référer à une entité identique. En OWL, ces assertions sur l'identité doivent être explicitement définies au travers des relations *owl:sameAs* et *owl:differentFrom*.

2.4.1.4/ INTERPRÉTATION DU MONDE REPRÉSENTÉ

La connaissance d'un domaine est contextuelle. Par conséquent, selon le contexte et la perspective, la signification d'une information peut avoir plus ou moins d'importance. Dès lors, l'interprétation autorisée par les différentes hypothèses a un impact direct sur les raisonnements qui seront appliqués sur le système. Dans les bases de données relationnelles, un seul schéma est utilisé pour définir le champ d'application et l'interprétation du monde. Cette unicité du schéma rend le monde fermé particulièrement adapté à la validation des données puisqu'il ne laisse place qu'à une seule interprétation du monde, par conséquent les faits sont vrais ou faux et aucune autre explication ne peut venir justifier la violation d'une contrainte d'intégrité. En revanche, dans le cadre de l'ontologie, le schéma et les données sont définis séparément [Motik et al., 2009]. Ainsi, plusieurs interprétations (monde) pour les mêmes données sont envisageables. Par conséquent, les raisonneurs en monde ouvert auront tendances à générer de la connaissance afin d'expliquer la violation d'une contrainte.

2.4.1.5/ DISCUSSION

L'hypothèse du monde fermé invite à définir "ce qui est possible". A l'inverse, l'hypothèse du monde ouvert permet de statuer "ce qui n'est pas possible". Lorsqu'une ontologie OWL est vide alors tout est possible. Ce n'est que lorsque l'on contraint progressivement l'ontologie qu'elle devient plus restrictive. Dans les domaines bien définis (sièges dans un avion, les livres dans une bibliothèque), le modèle relationnel constitue une approche adaptée car les données sont considérées comme complètes. L'hypothèse du monde fermé est performante pour faciliter la validation des données pendant les opérations de transaction. Le nombre de faits négatifs sur un domaine donné est généralement beaucoup plus grand que le nombre de faits positifs. Ainsi, dans de nombreuses applications, le nombre de faits négatifs est si grand que leur représentation explicite peut devenir pratiquement impossible. Dans de tels cas, il est plus simple et plus rapide de définir tous les faits "vrais" connus que d'énumérer toutes les conditions de "faux". Cependant, le modèle relationnel est un paradigme où l'information doit être complète et décrite par un schéma unique. Les bases de données traditionnelles exigent un accord sur un schéma qui doit être fait avant que les données puissent être stockées et interrogées. Le modèle relationnel suppose que tous les objets et les relations qui existent dans le domaine sont ceux qui sont explicitement représentés dans la base de données, et qui identifient de manière unique les noms d'objets dans ce domaine. Cela rend l'hypothèse du monde fermé et ses hypothèses connexes un très mauvais choix lorsque l'on tente de combiner des informations provenant de sources multiples, ou pour faire face à l'incertitude ou l'incomplétude du monde.

L'avantage principal de l'hypothèse du monde ouvert et du Web Sémantique, généralement associé à cette hypothèse, est de permettre aux informations d'être réutilisables. La réutilisation d'une ontologie permet d'assembler, étendre, spécialiser ou encore adapter les connaissances définies à partir d'autres ontologies. De cette manière, le Web Sémantique offre une bonne flexibilité pour permettre l'intégration de nouvelles connaissances lorsque l'application nécessite des connaissances spécifiques. Lorsqu'une ontologie est étendue, toutes les déclarations définies comme étant vraie le reste. Enfin, l'approche du Web Sémantique permet de fournir des mécanismes d'inférences afin de générer de la connaissance au sein d'une application.

En définitive, l'approche relationnelle s'avère un candidat tout indiqué pour la validation des données. Cependant, les irrégularités et l'incomplétude sont une limite à la conception du modèle relationnel. L'approche Web Sémantique dépasse ces limites en proposant une structure flexible du schéma des données. De plus, la dissociation explicite entre le schéma (TBox) et les données (ABox) offre un environnement propice pour l'interprétation des données représentées. Par conséquent, l'incomplétude du monde ouvert peut être partiellement comblée grâce à des raisonneurs capables d'auto-alimenter le système en se basant sur des contraintes.

Dans beaucoup d'application du Web Sémantique, il est intéressant d'utiliser OWL pour définir des contraintes d'intégrités qui doivent être satisfaites par les données. Cependant, l'hypothèse du monde ouvert et le rejet de l'hypothèse du nom unique sont jusqu'alors un frein au développement des contraintes d'intégrités en OWL [Tao et al., 2010]. En effet, les conditions définies pour lever des violations de contraintes en monde fermé, génèrent de nouvelles connaissances dans les applications de raisonnement basées sur OWL. Dès lors, il apparaît nécessaire de fournir des applications hybrides combinant à la fois les raisonnements en monde ouvert avec les capacités de validation de contraintes du monde fermé.

2.4.2/ INTERPRÉTATION DES CONTRAINTES : DE LA CONTRAINTÉ D'INTÉGRITÉ À L'AXIOME LOGIQUE

Les contraintes sont un enjeu majeur de la modélisation en ce sens qu'elles peuvent servir de support à la vérification et la validation des données ou bien servir de support aux mécanismes d'inférences permettant de générer des connaissances à partir des données existantes. Dans le premier cas, les contraintes seront appelées contraintes d'intégrité et dans le second, les contraintes seront assimilées à des axiomes logique.

2.4.2.1/ LES CONTRAINTES D'INTÉGRITÉ POUR LA DÉTECTION D'INCONSISTANCE

Les contraintes d'intégrité permettent d'anticiper l'insertion de valeurs incorrectes dans le modèle relationnel. Ce type de contrainte trouve son intérêt dans la validation, l'analyse et l'insertion de données. Les contraintes d'intégrités sont intimement liées au modèle unique qui contient les faits. En somme, une contrainte d'intégrité ne peut exister qu'au sein du monde fermé. La vérification des contraintes d'intégrité est similaire à la détection d'inconsistances logiques mais diffère au sens où les contraintes d'intégrité ne causent pas de nouveaux faits. En effet, elle ne peuvent détecter que la violation de l'une de ces contraintes. La violation d'une contrainte d'intégrité peut être causée par un manque d'information. Par exemple, si une contrainte d'intégrité définit que chaque individu de la

classe ville doit posséder un nom via la propriété "possèdeNom", alors cette propriété doit être renseignée pour chaque ville sous peine d'engendrer une violation de la contrainte. Lorsqu'une contrainte d'intégrité est violée, il existe deux moyens de la résoudre :

- Ajouter de l'information :
Exemple : Renseigner la propriété "possèdeNom" lorsque ça n'est pas encore le cas pour chaque ville.
- Supprimer des informations existantes :
Exemple : Supprimer les villes pour lesquels la propriété "possèdeNom" n'a pas été renseignée.

Il existe plusieurs formalismes permettant de définir des contraintes d'intégrités en OWL. Parmi ces formalismes, il est possible de citer les axiomes OWL standard, les règles SWRL ou encore les requêtes SPARQL.

2.4.2.2/ L'AXIOME LOGIQUE COMME SUPPORT D'INFÉRENCE

Les axiomes logiques sont également des contraintes et peuvent se définir exactement de la même manière que les contraintes d'intégrité. Toutefois, elles se distinguent par l'interprétation fournie par le raisonneur. A l'instar des contraintes d'intégrité qui sont indissociables du monde fermé, les axiomes logiques sont totalement dépendant de l'hypothèse du monde ouvert. Dès lors, si un axiome n'est pas satisfait alors tout peut être vrai à moins de prouver le contraire. De plus, plusieurs modèles peuvent satisfaire les axiomes. Les axiomes permettent de meilleures capacités d'inférences en ce sens qu'elles tentent d'inférer des faits pouvant expliquer l'insatisfaction d'une contrainte au sein du modèle. Toutefois, les faits inférés ne sont pas toujours intuitifs.

2.4.2.3/ DISCUSSION

Ces deux vues impactent directement les conclusions sur la vérification de la consistance et par conséquent sur l'inférence qui peut être extraite du système. Voici quelques exemples permettant de distinguer la vérification de la consistance dans l'hypothèse du monde ouvert et dans l'hypothèse du monde fermé :

Exemple 1 : contrainte de valeur

- Déclaration
 $Forêt \sqsubseteq \exists Forêt.Integer$
 $Forêt(ForêtDeMeudon)$
- Hypothèse du monde ouvert
Consistant : **vrai**
Raison : **ForêtDeMeudon** possède bien la propriété **ForêtID** mais on ne connaît pas la valeur exacte de cette propriété.
- Hypothèse du monde fermé
Consistant : **faux**
Raison : **ForêtDeMeudon** n'a pas d'**ForêtID**.

Sur l'exemple 1, une restriction est définie sur la classe *Forêt* impliquant que toute instance de cette classe doit avoir une valeur de type entier pour la propriété *ForêtID*. Cette déclaration peut avoir valeur de contrainte d'intégrité si elle est définie dans

le cadre d'un système en monde fermé ou bien d'axiome logique dans le contexte d'un système en monde ouvert. Dans le cas d'une interprétation en monde fermé, la vérification de la consistance par le raisonneur révélerait une violation de la contrainte d'intégrité, car l'instance *ForêtDeMeudon* ne possède pas de valeur pour la propriété *ForêtID*. En revanche, bien que la contrainte ne soit pas explicitement satisfaite, un raisonneur en monde ouvert ne détectera pas d'erreur considérant que la propriété *ForêtID* ainsi que la valeur existe. Toutefois, cette valeur n'est pas connue du système.

Exemple 2 : contrainte de participation

- Déclaration
 - $\top \sqsubseteq \forall estCapitaleDe.Pays$
 - $estCapitaleDe(Paris, France)$
- Hypothèse du monde ouvert
 - Consistant : **vrai**
 - Inférence : *Pays(France)*
- Hypothèse du monde fermé
 - Consistant : **faux**
 - Raison : *Pays(France)* n'est pas explicitement définie.

La première ligne de l'exemple 2 spécifie que lorsque la propriété *estCapitalDe* est instanciée alors l'instance du co-domaine doit être de type *Pays*. La deuxième ligne propose un exemple d'instantiation de la propriété définissant que Paris est la capitale de la France. En revanche, aucune déclaration ne spécifie que France est de type *Pays*. En conséquence, un raisonneur en monde fermé considérera que la contrainte est violée et détectera une erreur. En monde ouvert, le raisonneur sera, quant à lui, capable d'inférer que l'instance France est de type *Pays* garantissant ainsi la consistance du système.

Exemple 3 : contrainte d'unicité

- Déclaration
 - $\top \sqsubseteq \leq 1 aPourNom$
 - $aPourNom(Entité702, \hat{îleDeBeauté})$
 - $aPourNom(Entité702, Corse)$
- Hypothèse du monde ouvert
 - Consistant : **vrai**
 - Inférence : $\hat{îleDeBeauté} \doteq Corse$
- Hypothèse du monde fermé
 - Consistant : **faux**
 - Raison : **Entité702** a plus d'une valeur pour la propriété **aPourNom** ce qui est interdit car cette dernière est fonctionnelle.

La première ligne de l'exemple 3 définit que la propriété "aPourNom" est fonctionnelle. Par conséquent, toute instance utilisant cette propriété ne peut avoir qu'une seule valeur en co-domaine de cette propriété. La deuxième ligne spécifie que l'instance "Entité702" a pour nom "îleDeBeauté". La troisième ligne spécifie que l'instance "Entité702" a pour nom "Corse". Le raisonneur en monde fermé détecte donc une erreur du fait qu'une même instance possède plus d'une valeur pour la même propriété. Le raisonneur en monde ouvert, quant à lui, considère que "îleDeBeauté" et "Corse" réfèrent à la même entité.

Les quelques exemples exposés précédemment démontrent que l'hypothèse du monde fermé et l'hypothèse du monde ouvert révèlent chacune des fonctionnalités intéressantes selon les besoins de l'utilisateur. L'exemple 1 spécifiait une contrainte de valeur afin de garantir que chaque entité de type "Forêt" possède un identifiant. Les identifiants sont généralement unique pour chaque objet. Dans le cadre des systèmes d'information, les identifiants sont souvent d'une importance capitale pour garantir le bon fonctionnement du système, c'est pourquoi les identifiants d'une base de données relationnelles sont fréquemment définis en tant que contrainte d'intégrité. L'application du monde ouvert dans le cadre de telles contraintes autorisent beaucoup de liberté sur la structuration des données et admet intrinsèquement des jeux de données pouvant être fortement incomplets. Dès lors, il serait possible de stocker toutes sortes d'entités et de données, cependant, le contenu serait difficilement exploitable en raison de l'incomplétude résultante. En résumé, un système trop incomplet en monde ouvert aura tendance à répondre très fréquemment "je ne sais pas" le rendant réaliste mais inexploitable, le monde fermé, en revanche, assume l'hypothèse d'un monde où les informations sont complètes et peut donc aisément répondre "vrai" ou "faux" à n'importe quelle requête. L'hypothèse du monde fermé s'avère très adaptée pour des applications artificiellement complètes comme des systèmes de réservation de billet d'avion (où le nombre de places, les horaires de départ et d'arrivée, et les tarifs sont tous stockés dans le système). Cependant, le monde réel tel que nous le percevons et tel que nous le connaissons révèle généralement beaucoup d'inconnues. Par conséquent, notre connaissance à propos des phénomènes du monde réel est, par nature, incomplète ce qui rend l'hypothèse du monde fermé irréaliste et inadéquate.

2.5/ MODÉLISATION SPATIO-TEMPORELLE DANS LE WEB SÉMANTIQUE

Le domaine de la modélisation spatio-temporelle tente de donner une représentation conceptuelle de la dynamique du monde réel. Généralement, la connaissance de l'environnement représenté est incomplète, c'est pourquoi les systèmes spatio-temporels existants ne sont que des outils permettant d'assister les experts afin de découvrir des connaissances implicites. Dès lors, il existe un besoin de modéliser la sémantique et le contexte de l'environnement géospatial afin de fournir des systèmes d'informations géographiques de nouvelle génération capable de fournir des raisonnements proches de ceux des experts. Dans cette section, nous étudierons toutes les contributions du Web Sémantique permettant de fournir des outils pour la modélisation ou le raisonnement dans le domaine spatial ou temporel.

2.5.1/ MODÉLISATION SPATIALE DANS LE WEB SÉMANTIQUE

Face à la montée en puissance des données et applications spatialisées, de nouveaux travaux visant à fournir un Web Sémantique Géospatial ont vu le jour pour proposer des supports complets pour l'information géographique. La création de ce Web Sémantique géospatial nécessite le développement de multiples ontologies spatiales, chacune avec une sémantique formelle. La représentation de cette sémantique doit permettre à la fois

aux machines de traiter les données, mais également d'offrir à l'utilisateur une meilleure compréhension. Enfin, les outils d'interrogation doivent également évoluer afin d'autoriser l'intégration des fonctions d'analyses spatiales sur ces ontologies [Egenhofer, 2002].

2.5.1.1/ GML (GEOGRAPHY MARKUP LANGUAGE)

GML [Cox et al., 2002] est une grammaire XML définie par l'OGC (Open Geospatial Consortium) permettant de représenter des caractéristiques géographiques. L'objectif de ce langage est de servir de support de modélisation pour des systèmes d'informations géographiques ainsi que de servir de format d'échange d'informations géographiques sur internet. GML distinguent les objets géographiques qui peuvent être des points, des lignes, des polygones ou d'autres types de données plus complexes, et les caractéristiques géographiques qui réfèrent à des 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. Une même caractéristique peut être représentée par plusieurs objets géographiques sous réserve que ces objets soient représentés par des géométries simples.

GML comporte bon nombre de primitives permettant notamment de définir des caractéristiques géographiques, des géométries, un système de coordonnées cartographique, des relations topologiques, des données temporelles ainsi que des caractéristiques géographiques dynamiques, c'est-à-dire des entités évoluant dans le temps au travers de plusieurs états successifs tel qu'un bus se déplaçant au cours du temps.

En définitive, GML constitue un support d'encodage très complet pour l'information géographique. Cependant, GML ne permet pas de décrire le contexte d'un environnement géospatial de manière aussi performante que des langages issus du Web Sémantique. D'autre part, GML a pour seule vocation de fournir un format d'échange. Par conséquent, il n'existe aucun langage de requête permettant d'interroger un fichier GML. Cependant, GML a servi de support à bon nombre de travaux issus du W3C Geospatial Incubator Group [Lieberman et al., 2007] tels que GeoRSS ou encore GeoOWL.

2.5.1.2/ BASIC RDF GEO VOCABULARY

Basic RDF Geo Vocabulary [Brickley, 2003] est un langage créé par le W3C Semantic Web Interest Group¹ en 2003 afin d'offrir un moyen de représenter des points en RDF dans un système de coordonnées WGS84. Chaque point peut ainsi être décrit en RDF/XML avec des coordonnées de latitude, longitude et altitude. Le développement des aspects géospatiaux n'étant pas le coeur du travail du W3C Semantic Web Interest Group, Basic RDF Geo Vocabulary est un vocabulaire minimaliste composé d'une classe "Point" et des propriétés "lat", "long" et "alt" correspondant respectivement aux coordonnées de latitude, longitude et altitude.

2.5.1.3/ GEORSS ET GEOOWL

GeoRSS [Reed et al., 2006] est à l'origine un encodage visant à incorporer de l'information géographique dans des flux web de sorte que des applications puissent interroger,

1. <http://www.w3.org/2001/sw/interest/>

agréger, partager et fusionner des flux géoréférencés. Les efforts fournis sur GeoRSS ont donné lieu à deux encodages principaux que sont GeoRSS GML et GeoRSS Simple. GeoRSS Simple constitue la version légère de GeoRSS en supportant des géométries basiques (point, ligne, boîte englobante et polygone) ainsi que quelques fonctions de bases pour localiser les entités. GeoRSS GML, quant à lui, a bénéficié du soutien de l'Open Geospatial Consortium (OGC) et supporte de nombreuses caractéristiques supplémentaires par rapport à GeoRSS Simple en se basant sur GML. Entre autre GeoRSS GML permet de définir des systèmes de coordonnées autres que le traditionnel WGS84.

Bien que GeoRSS n'ait pas été conçu à destination du Web Sémantique, le W3C Geospatial Incubator Group a largement été influencé par cet encodage pour enrichir Basic RDF Geo Vocabulary afin de développer une ontologie géospatiale et devenir une recommandation du W3C pour la représentation sur le web d'entités géographiques et géo-localisées. Les travaux donneront naissance à l'ontologie GeoOWL². L'ontologie GeoOWL est un modèle riche et flexible basé sur le vocabulaire GeoRSS pour représenter des concepts géospatiaux au travers de propriétés et de classes géographiques. En définitive, malgré la publication de jeux de données et la création de quelques triplestores supportant les ontologies GeoRSS et GeoOWL, ces dernières ne devinrent jamais des recommandations officielles du W3C.

2.5.1.4/ SPARQL-ST, ST-RDF

Outre les différentes directions initiées par le W3C, la communauté géospatiale s'est également intéressée à la définition d'outils capables de représenter et d'interroger des données géospatiales. En se basant sur GeoRSS, une extension de SPARQL nommée SPARQL-ST fût proposée afin de fournir un moyen de définir des requêtes spatiales sur une ontologie. Contrairement à GeoOWL et Basic RDF Geo Vocabulary, SPARQL-ST permet de gérer plusieurs systèmes de coordonnées cartographiques. Toutefois, à cause de la nouvelle syntaxe définie en supplément de SPARQL, ce langage est utilisable uniquement sur des systèmes implémentant cette extension. A notre connaissance, il n'existe qu'un prototype d'implémentation conçu par les auteurs de cette extension [Perry et al., 2011].

Dans [Reed et al., 2006], les auteurs proposent d'enrichir le Web Sémantique à l'aide de données spatiales et temporelles au travers d'une extension de RDF et de SPARQL respectivement appelé stRDF et stSPARQL. L'approche se base sur les travaux de la communauté des bases de données contraintes. Ce domaine tente d'étendre les systèmes de base de données traditionnels en enrichissant à la fois le modèle de données et les primitives de requête avec des contraintes. Ainsi, stRDF est un modèle de données contraint basé sur RDF pour permettre de représenter des données spatiales et temporelles. Le langage de requête, stSPARQL, étend SPARQL en incluant de nouveaux opérateurs pour interroger les relations topologiques et introduit une nouvelle syntaxe pour manipuler des variables spatiales. Les géométries sont définies à l'aide d'un ensemble de points semi-linéaires, par conséquent, stSPARQL n'est pas compatible avec les standards de l'OGC. Les auteurs ont volontairement fait ce choix en avançant qu'un ensemble de points semi-linéaires comme type de donnée unique pouvait permettre de définir aussi bien des points, des lignes et des polygones sans nécessiter d'imposer une

2. http://www.w3.org/2005/Incubator/geo/XGR-geo-20071023/W3C_XGR_Geo_files/geo_2007.owl

hiérarchie de type de données spatiales à l'utilisateur. Cependant, il est à noter qu'il peut s'avérer difficile d'obtenir des informations sur une géométrie sans savoir à l'avance le type de cette géométrie. Par exemple, les propriétés de certaines géométries peuvent être différentes selon le type de données spatiales de celles-ci comme c'est le cas dans GeoOWL.

2.5.1.5/ GeoSPARQL

GeoSPARQL est le nouveau standard du W3C pour la représentation et l'interrogation de données géospatiales liées pour le Web Sémantique. Afin d'exécuter des requêtes GeoSPARQL sur un ensemble de données, ces données doivent avoir une partie spatiale définie dans l'ontologie GeoSPARQL. La partie spatiale des objets de l'ontologie est généralement définie à l'aide du format WKT (Well Known Text). WKT est un format standard en mode texte utilisé pour représenter des objets géométriques vectoriels issus des systèmes d'informations géographiques. En outre, WKT supporte également la définition de système de coordonnées. Toutefois, GeoSPARQL autorise également la définition de géométrie en se basant sur GML. Enfin, pour la définition des relations topologiques, GeoSPARQL permet d'utiliser Simple Features, RCC8 ou DE-9IM. Outre la définition d'opérateurs topologiques, Simple feature spécifie un modèle de stockage commun pour la plupart des données géographiques en deux dimensions (point, ligne, polygone, multi-point, multi-ligne, etc). Pour présenter rapidement les bases de GeoSPARQL, nous allons commencer avec un exemple simple de données et une requête utilisant une boîte englobante. Tout d'abord, nous définissons la tour Eiffel comme appartenant à la classe "Feature", et lui attribuons un point dans l'espace comme géométrie correspondante.

Listing 2.1 – Exemple illustrant l'activation d'une représentation spatiale d'un objet dans GeoSPARQL

```
ex:TourEiffel a geo:Feature ;
  rdfs:label "Tour Eiffel";
  geo:hasGeometry ex:WMPoint .
ex:WMPoint a sf:Point;
  geo:asWKT "POINT(-77.03524 38.889468)"^^sf:wktLiteral.
```

La première ligne indique que la ressource "ex :TourEiffel" est de type "Feature", c'est-à-dire un objet du monde réel qui peut être localisé. La deuxième ligne donne une étiquette à cette nouvelle ressource. La troisième ligne donne à cet objet une localisation spatiale. La quatrième ligne indique que la localisation spatiale est un point, et la cinquième ligne définit les coordonnées de ce point. A présent, nous pouvons exécuter une requête recherchant des objets de type "Feature" dans une zone définie par des coordonnées de latitude et de longitude.

Listing 2.2 – Requête GeoSPARQL renvoyant tous les objets situés spatialement dans la boîte englobante définie

```
SELECT ?f
WHERE {
  ?f geo:hasGeometry ?g .
  ?g geo:asWKT ?gWKT .
  FILTER (geof:sfWithin(
    ?gWKT, "POLYGON((-77.2_38.8,-77_38.8,_-77_39,-77.2_39.9,-77.2_38.8))"^^sf:wktLiteral
  )
)
```

Cette requête recherche les ressources qui ont une géométrie qui se trouve dans un rectangle englobant. La première ligne trouve les géométries associées à une ressource. La deuxième ligne obtient la représentation des géométries. Le filtre compare la représentation avec le rectangle englobant à l'aide d'un opérateur spatial pour voir si la géométrie est située dans la zone englobante. A noter que la zone englobante est composée de cinq points et non quatre correspondants aux quatre coins, car il est nécessaire de répéter la définition du premier point pour fermer le polygone. Notre polygone étant un rectangle, il est nécessaire de fermer le polygone avec un cinquième point. L'ontologie GeoSPARQL se compose de trois classes principales :

- `geo:Feature` : une chose qui peut avoir une localisation spatiale comme un parc ou un monument, etc. En outre, un individu de la classe « feature » représente un objet possédant une identité et des attributs.
- `geo:Geometry` : une localisation spatiale définie par un point ou une empreinte spatiale, c'est-à-dire, un ensemble de coordonnées représentant la géographie d'une entité.
- `geo:SpatialObject` : la superclasse de "Feature" et de "Geometry".

Les relations entre ces classes sont schématisées sur la figure 2.16 :

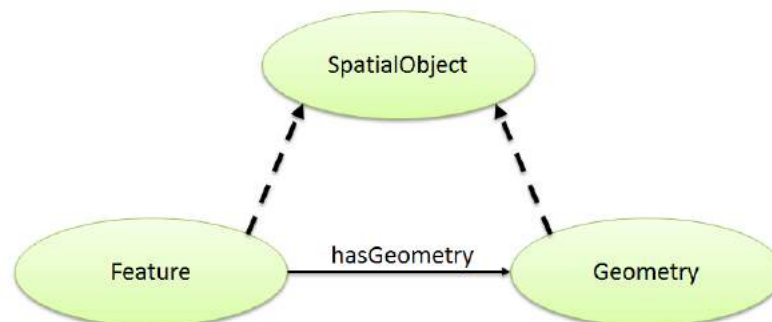


FIGURE 2.16 – Ontologie GeoSPARQL

La propriété `geo:hasGeometry` relie les "Features" à leur géométrie. En séparant les entités réelles et leurs empreintes spatiales, GeoSPARQL permet de lier plusieurs géométries à un même objet. La ressource correspondant à la géométrie a ensuite une représentation littérale en RDF, qui est liée à une propriété nommée par le type de représentation. Par exemple, la propriété de `geo:asWKT` relie la ressource de la géométrie à un `wktLiteral`.

L'ontologie GeoSPARQL est petite et est destinée à être raccordée à l'ontologie d'un domaine particulier. Ceci est facilement réalisable en définissant une classe dans l'ontologie de domaine comme sous-classe de `geo:Feature`, ce qui signifie que les instances de la classe peuvent pointer vers une `geo:Geometry` avec la propriété `geo:hasGeometry`. Ce qui suit est une ontologie de "Points d'intérêt" relativement simple pour montrer comment une ontologie est liée à GeoSPARQL. La dernière ligne en rouge est la ligne clé pour réaliser la connexion à GeoSPARQL.

Listing 2.3 – Activation d'une représentation spatiale dans une ontologie de domaine avec GeoSPARQL

```

ex:Restaurant a owl:Class;
  rdfs:subClassOf ex:Service .
ex:Park a owl:Class;
  rdfs:subClassOf ex:Attraction .
ex:Museum a owl:Class;
  rdfs:subClassOf ex:Attraction .
ex:Monument a owl:Class;
  rdfs:subClassOf ex:Attraction .
ex:Service a owl:Class;
  rdfs:subClassOf ex:PointOfInterest .
ex:Attraction a owl:Class;
  rdfs:subClassOf ex:PointOfInterest .
ex:PointOfInterest a owl:Class;
  rdfs:subClassOf geo:Feature .

```

GeoSPARQL étend le langage SPARQL avec des opérateurs spatiaux permettant d'exécuter des requêtes spatiales sur un triplestore contenant l'ensemble des triplets RDF. La première étape afin de lancer une requête spatiale consiste généralement à accéder à la géométrie des objets impliqués dans l'analyse spatiale.

Listing 2.4 – Requête GeoSPARQL renvoyant la géométrie d'un objet au format WKT

```

SELECT ?wkt
WHERE {
  ex:TourEiffel geo:hasGeometry ?g .
  ?g geo:asWKT ?wkt .
}

```

GeoSPARQL offre la possibilité de faire des comparaisons topologiques entre géométries. Il y a trois façons de le faire :

1. En utilisant les fonctions de filtre de GeoSPARQL.
2. En utilisant les propriétés entre les géométries.
3. En utilisant les propriétés entre les objets (Feature).

Les fonctions de filtre nécessitent d'écrire des requêtes complexes pour effectuer des opérations spatiales sur un objet. En effet, à partir de l'objet, il est nécessaire de trouver la géométrie correspondante via la propriété *geo:hasGeometry* puis, à partir de la géométrie, obtenir la représentation au format WKT ou GML respectivement via les propriétés *geo:asWKT* et *geo:asGML*. Pour cette raison, GeoSPARQL offre des mécanismes de simplification de requêtes en autorisant l'utilisation d'opérateurs directement entre les géométries ou bien encore directement entre les objets spatiaux. Les opérateurs ne sont plus des fonctions de filtre, mais des propriétés entre les géométries ou objets spatiaux. Le code suivant permet d'appliquer un opérateur spatial en utilisant les fonctions de filtre de GeoSPARQL :

Listing 2.5 – Requête GeoSPARQL utilisant un opérateur spatial à l'aide des fonctions de filtre

```

SELECT ?f
WHERE {
  ?f geo:hasGeometry ?g .
  ?g geo:asWKT ?gWKT .
  FILTER (geof:sfWithin(
    ?gWKT, "POLYGON((-77.2_38.8, -77_38.8, -77_39, -77.2_39.9, -77.2_38.8))"^^sf:wktLiteral
  )
)
}

```

Les comparaisons topologiques peuvent également s'effectuer entre deux géométries du triple store par une propriété directe entre les géométries. Par exemple, pour trouver des restaurants qui se situent dans la ville de Strasbourg, nous aurons la requête suivante :

Listing 2.6 – Requête GeoSPARQL utilisant un opérateur spatial à l'aide des propriétés

```
SELECT ?f
WHERE {
  ex:Strasbourg geo:hasGeometry ?g1 .
  ?f a ex:Restaurant;
    geo:hasGeometry ?g2 .
  ?g2 geo:sfWithin ?g1 .
}
```

A noter que le préfixe sur les propriétés topologiques diffèrent de celui des fonctions topologiques. Cependant, le même ensemble d'opérateur est disponible. Dans nos exemples, nous avons opté pour l'utilisation des opérateurs du modèle "Simple Features", cependant GeoSPARQL propose également des opérateurs approximativement équivalents en utilisant les modèles DE-9IM ou encore RCC 8. Tous les opérateurs implémentés dans GeoSPARQL sous forme de fonctions et de propriétés pour chacun des modèles sont disponibles dans le tableau ci-dessous :

Simple Features		Egenhofer (DE-9IM)		RCC 8	
Fonctions	Propriétés	Fonctions	Propriétés	Fonctions	Propriétés
geof:sfEquals	geo:sfEquals	geof:ehEquals	geo:ehEquals	geof:rcc8eq	geo:rcc8eq
geof:sfDisjoint	geo:sfDisjoint	geof:ehDisjoint	geo:ehDisjoint	geof:rcc8dc	geo:rcc8dc
geof:sfIntersects	geo:sfIntersects	geof:ehCovers	geo:ehCovers	geof:rcc8tp	geo:rcc8tp
geof:sfTouches	geo:sfTouches	geof:ehMeet	geo:ehMeet	geof:rcc8ec	geo:rcc8ec
geof:sfWithin	geo:sfWithin	geof:ehInside	geo:ehInside	geof:rcc8ntpp	geo:rcc8ntpp
geof:sfContains	geo:sfContains	geof:ehContains	geo:ehContains	geof:rcc8ntppi	geo:rcc8ntppi
geof:sfOverlaps	geo:sfOverlaps	geof:ehOverlap	geo:ehOverlap	geof:rcc8po	geo:rcc8po
geof:sfCrosses	geo:sfCrosses	geof:ehCoveredBy	geo:ehCoveredBy	geof:rcc8tppi	geo:rcc8tppi

Les opérateurs définis sous forme de propriété permettent également d'appliquer une opération spatiale entre deux objets plutôt que sur leur géométrie pour obtenir des requêtes simplifiées. Par exemple, au lieu de représenter la requête « quels sont les restaurants dans la ville de Strasbourg ? » telle que nous l'avons exprimée précédemment, elle pourra aussi bien être exprimée comme suit :

Listing 2.7 – Requête simplifiée GeoSPARQL utilisant un opérateur spatial à l'aide des propriétés

```
SELECT ?f
WHERE {
  ex:Strasbourg geo:sfWithin ?f .
  ?f a ex:Restaurant;
}
```

Pour utiliser cette méthode d'interrogation, les objets en question nécessitent d'être définis avec une géométrie par défaut. Cela permet au triple store de trouver la géométrie par défaut pour chaque objet et de l'utiliser pour la comparaison entre les objets. Les géométries par défaut sont exprimées avec la propriété `geo:hasDefaultGeometry` au lieu de la propriété `geo:hasGeometry`.

2.5.2/ MODÉLISATION DU TEMPS DANS LE WEB SÉMANTIQUE

Outre la modélisation spatiale, le Web Sémantique a très tôt nécessité l'introduction de données temporelles au sein des langages disponibles. A l'origine, les ontologies de temps avaient pour objectif de décrire le contenu temporel des pages web ainsi que les propriétés temporelles des web services. Dès lors plusieurs ontologies de temps ont fait leur apparition au sein de la communauté. Dans cette section, nous étudierons deux travaux majeurs du domaine.

2.5.2.1/ OWL-TIME

OWL-Time [Hobbs et al., 2004] est une ontologie de temps conçue pour décrire le contenu temporel des pages web ainsi que les propriétés temporelles des services web. OWL-Time permet de représenter à la fois des instants et des intervalles de temps. Des propriétés sont définies pour spécifier qu'un instant débute ou termine un intervalle de temps. En outre, les 13 relations d'Allen sont utilisées pour spécifier les relations entre les intervalles. L'ontologie OWL-Time prend également en compte la durée en permettant de définir des unités temporelles telles que des secondes, minutes, heures, jours, etc. Enfin, OWL-Time est la seule ontologie connue à permettre les agrégats temporels, c'est-à-dire à représenter des expressions telles que « tous les premiers mardis de la semaine ». OWL-Time se compose d'une classe principale nommée *TemporalEntity*, elle-même composée de deux sous-classes *Instant* et *Interval*. Les propriétés *begins* et *ends* ont été définies respectivement afin de spécifier le début et la fin d'une entité temporelle. D'autres propriétés ont été définies afin d'enrichir les relations entre les entités temporelles. par exemple, la relation *inside* s'applique sur les instants et permet de définir que ce dernier fait partie d'un intervalle. La classe *Interval* possède une sous-classe *ProperInterval* qui correspond à un intervalle, dont les instants de début et de fin ne sont pas égaux.

2.5.2.2/ SWRL TEMPORAL ONTOLOGY

SWRL Temporal Ontology [O'Connor et al., 2011] permet de représenter des informations temporelles au format OWL en se basant sur des intervalles de temps. SWRL se présente comme la combinaison de OWL et du langage RuleML utilisé pour la déduction, la réécriture et l'inférence. Une règle SWRL se présente toujours sous la forme suivante :

$$\textit{antecedent} \Rightarrow \textit{consequent} \quad (2.7)$$

Où l'antécédent et le conséquent sont tous deux une conjonction d'atomes tels que $a_1 \wedge \dots \wedge a_n$. Les variables sont définies en les préfixant à l'aide d'un point d'interrogation. En utilisant cette syntaxe, il est possible de définir que la combinaison des propriétés "parent" et "frère" implique une propriété "oncle" :

$$\textit{parent}(?x, ?y) \wedge \textit{brother}(?y, ?z) \Rightarrow \textit{uncle}(?x, ?z) \quad (2.8)$$

SWRL propose une syntaxe facilement compréhensible pour l'utilisateur et possède également l'avantage de proposer des fonctions spécifiques utilisables au sein des règles appelées built-ins. SWRL Temporal Ontology propose une liste de built-ins permettant de vérifier les relations d'Allen. Par exemple, il est possible d'utiliser un built-ins temporel (ex :before) pour définir qu'une personne dont la date de naissance est antérieure à une autre personne est plus âgée que cette dernière :

$$Person(?x) \wedge birth(?x, ?x1) \wedge Person(?y) \wedge birth(?y, ?y1) \wedge before(?x1, ?y1) \quad (2.9)$$

$$\Rightarrow olderThan(?x, ?y) \quad (2.10)$$

Dans ce modèle, les informations (faits ou propositions) peuvent être associées avec des instants ou intervalles de temps qui définissent le ou les moments pendant le(s)quel(s) ces informations peuvent être tenues pour vraies. En dehors de ces instants ou intervalles de temps, aucune conclusion ne peut être tirée. En outre, bien qu'elle ne permette pas d'établir de relation entre les instants ou intervalles de temps, cette ontologie propose des built-ins temporels spécifiques au langage basé sur les règles : SWRL. Ces built-ins permettent de vérifier les relations d'Allen entre les différentes entités temporelles lors de l'interrogation. SWRL Temporal Ontology [O'Connor et al., 2011] se compose d'une classe principale *Entity* composée des sous-classes *Duration*, *Granularity*, *ValidTime* et *Proposition*. La classe *ValidTime* se compose de deux sous-classes *ValidInstant* et *ValidPeriod* permettant de définir le moment durant le lequel un fait ou une proposition est vraie. La propriété *hasTime* a été définie afin d'associer une date à une instance de la classe *ValidInstant* et les propriétés *hasStartTime* et *hasFinishTime* ont été définies afin de spécifier une date de début et une date de fin à une instance de la classe *ValidPeriod*. La classe *Duration* permet de représenter les durées. Pour cela, la propriété fonctionnelle *hasCount* a été définie pour spécifier un entier ainsi qu'une propriété fonctionnelle dont le *range* est la classe *Granularity* comportant toutes granularités. Enfin la classe *Proposition* comporte une sous-classe *ExtendedProposition*. Une propriété *hasValidTime* ayant pour *Domain* la classe *Proposition* et pour *Range* la classe *ValidTime* a été définie. Dès lors on peut ajouter des capacités temporelles à n'importe quelle classe d'une ontologie non-temporelle, il suffit de définir une classe comme sous-classe de *ExtendedProposition*.

2.5.3/ ONTOLOGIE DE FLUENT : MODÉLISATION DE LA DYNAMIQUE D'UN SYSTÈME

OWL-Time et SWRL Temporal Ontology sont des outils performants destinés à la représentation du temps dans une ontologie. Cependant, l'introduction de ces données temporelles ne permet pas, à elle seule, de représenter l'évolution des objets. Pour pallier à ce problème, des ontologies de plus haut niveau, appelée ontologie de fluents ont émergé et considèrent les objets comme pouvant être représentés à divers instants de temps. Ces ontologies intègrent à la fois le modèle RDF, les relations d'Allen et la gestion de l'évolution d'un environnement géographique.

2.5.3.1/ NAISSANCE DES ONTOLOGIES DE FLUENT

Une fois le modèle de temps défini, le besoin de représenter l'évolution d'un concept dans le temps subsiste. Il s'agit alors de dater les propriétés d'un même concept à mesure qu'elles évoluent, c'est-à-dire de convertir des propriétés statiques en propriétés dynamique dans l'ontologie. Dans un modèle cela s'exprime par une classe-association définie entre deux entités. Facilement gérable dans le domaine des bases de données, il n'en est pas de même dans une ontologie car OWL ne permet que des relations binaires entre des individus. Plusieurs modèles, comme Temporal RDF, Réification, Versioning, ou encore 4D-fluent, ont toutefois été proposés pour gérer ce type de relation dans une ontologie.

La réification La Réification est une technique utilisée pour représenter des relations n-aires avec des langages comme OWL autorisant uniquement des relations binaires. Mais cette technique pose de nombreux problèmes. Pour chaque propriété statique à convertir en propriété dynamique, on passe d'une relation binaire à 4 relations binaires. La complexité de l'ontologie est donc considérablement accrue ainsi que les requêtes. De plus, les capacités d'inférences s'en trouvent amoindries. Enfin, la réification est très sujette à la redondance des objets réifiés.

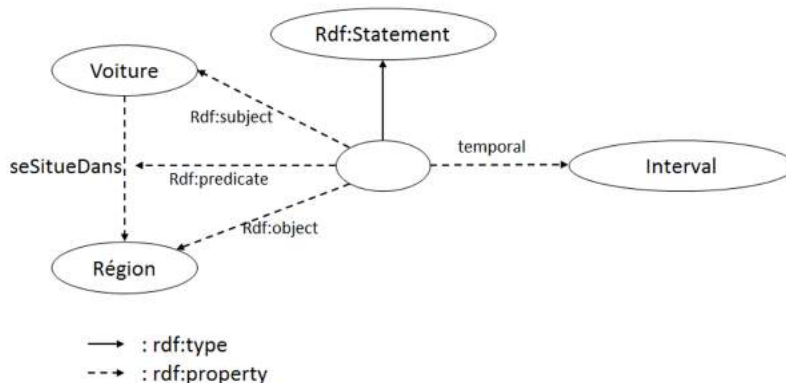


FIGURE 2.17 – Temporal RDF basé sur le principe de la réification

Temporal RDF Temporal RDF [Gutierrez et al., 2007] se base sur la technique de réification et propose une extension du standard RDF permettant de nommer des propriétés avec l'intervalle de temps correspondant. Cependant, il faut noter que RDF ne possède pas toute l'expressivité du langage OWL et ne permet pas l'utilisation de relations qualitatives. La figure 2.17 illustre un exemple de triplet temporel exprimé avec temporal RDF.

Relation n-aire Le modèle des relations n-aires [Noy et al., 2006], à l'instar de la réification, est une technique générale pour représenter des relations n-aires. Cependant, contrairement à la réification, ce modèle ne représente pas la relation dynamique comme l'objet d'une propriété mais utilise deux propriétés chacune liée à un nouvel objet

intermédiaire. Le nouvel objet intermédiaire est lié à un intervalle de temps pour permettre de définir la durée de validité de la relation. Contrairement à la réification qui nécessite l'ajout de deux objets pour représenter une relation dynamique, le modèle des relations n-aires ne nécessite qu'un seul objet supplémentaire pour introduire la temporalité de la relation. En revanche, ce modèle souffre de la redondance dans le cas de relations inverses ou symétriques (voir figure 2.18).

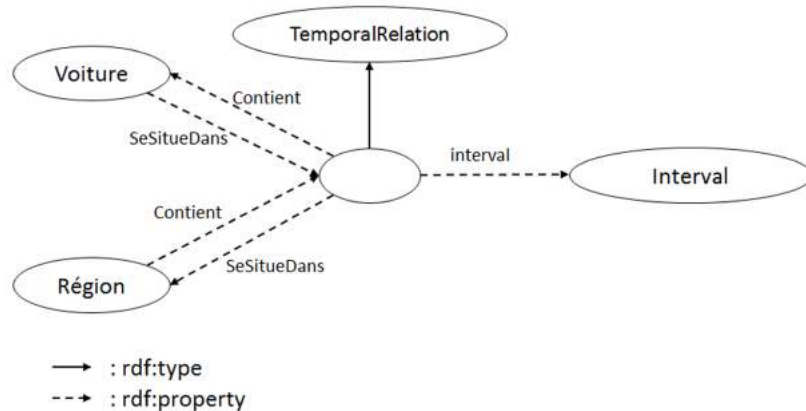


FIGURE 2.18 – Modèle des relations n-aires

Le versioning Le versioning [Klein et al., 2001] consiste à établir différentes versions de l'ontologie à différents moments. L'inconvénient majeur de cette méthode réside dans la redondance résultante du moindre changement sur un attribut. De plus, les requêtes doivent alors s'effectuer sur plusieurs versions de l'ontologie. La figure 2.19 illustre l'exemple d'une voiture changeant de région entre un instant $t1$ et un instant $t2$ en utilisant la méthode du versioning.

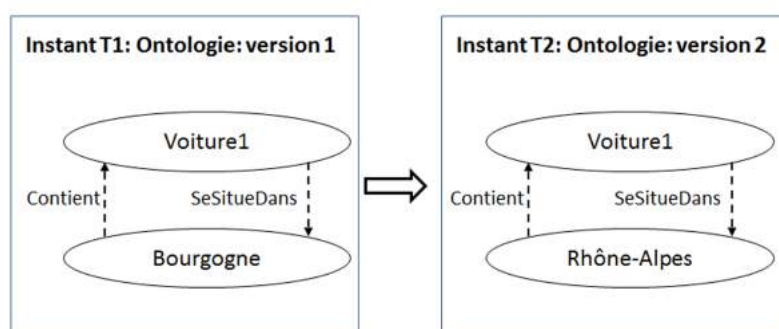


FIGURE 2.19 – Versioning

2.5.3.2/ 4D-FLUENTS

Dans [Welty et al., 2006], les auteurs abordent la notion d'ontologie de fluents permettant de représenter le fait que les propriétés des objets varient au cours du temps. L'approche considère les objets comme possédant des parties temporelles, appelées timeslices, et la représentation du changement des propriétés d'un objet se fait au travers de l'utilisation de fluents qui sont des propriétés valides uniquement durant un certain intervalle de temps. Lorsque la propriété d'un objet varie, un nouveau timeslice est établi, possédant la nouvelle propriété de l'objet. L'ontologie présentée dans cette article a pour but de modéliser des objets perdurants (ou *occurents*), c'est-à-dire des objets possédant plusieurs parties temporelles les représentant durant un certain intervalle de temps. Ces parties temporelles sont appelées des timeslices et une entité est définie par l'ensemble de ses timeslices. On peut alors représenter l'évolution des propriétés d'un objet au cours du temps en utilisant l'ensemble des timeslices. La figure 2.20 illustre un exemple à l'aide du modèle 4D-fluents.

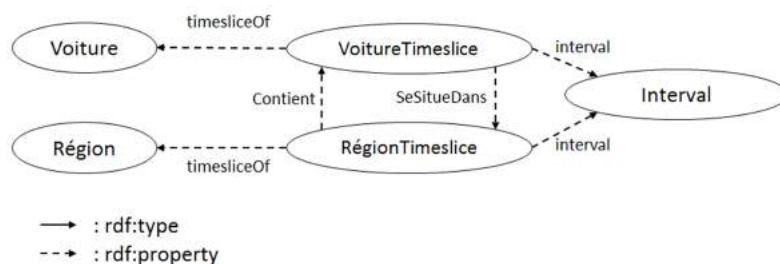


FIGURE 2.20 – Exemple du modèle 4D-fluent

Dans l'approche 4D-Fluent, la classe *TimeInterval* utilisée correspond à la classe *Interval* définie dans OWL-Time [Hobbs et al., 2004]. On considère que l'intervalle de temps associé au timeslice représente la durée pendant laquelle le fluent associé est valide. Si un timeslice est associé à plusieurs *fluents* alors ces *fluents* doivent être valides sur le même intervalle de temps. La figure 2.21 illustre la terminologie du modèle 4D-fluents.

Pour rendre une propriété dynamique, c'est-à-dire pour transformer une propriété statique en un fluent, il suffit de définir une propriété comme sous-propriété de *FluentProperty*. La propriété *tsTimeInterval* permet d'associer un intervalle de temps à un timeslice. Enfin la propriété *tsTimesliceOf* permet d'associer un timeslice à n'importe quelles instances des classes de l'ontologie à l'exception des instances de la classe *TimeInterval*.

2.5.3.3/ TOWL

TOWL (Time-determined Ontology Web Language) [Frasincar et al., 2010], [Milea et al., 2012] est un langage visant à étendre le langage OWL avec une dimension temporelle afin de permettre la représentation du temps, du changement et des transitions d'état. TOWL propose une architecture en couche afin d'étendre les capacités de OWL-DL comme le montrent les 4 couches de la figure ci-dessous :

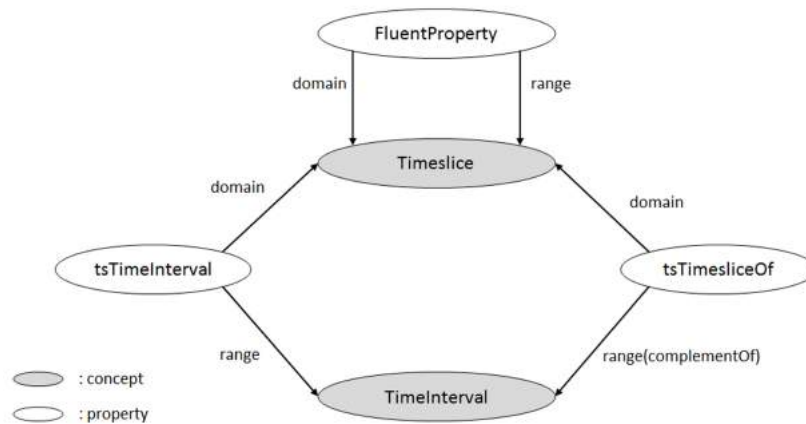


FIGURE 2.21 – Illustration de la terminologie du modèle 4D-fluent

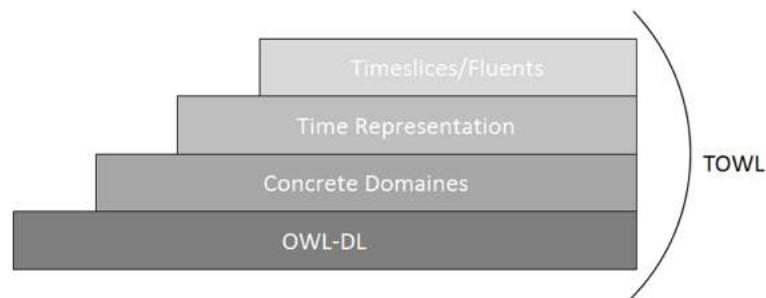


FIGURE 2.22 – les 4 couches du modèle TOWL

OWL-DL est logiquement la première couche et ne nécessite pas d'explication supplémentaire. La deuxième couche concerne l'expressivité du langage d'une manière générale, plutôt que de le restreindre au sens purement temporel. Dès lors, la couche des domaines concrets autorise la représentation d'intervalles et les 13 relations d'Allen entre ces intervalles. La couche suivante permet de représenter le temps de manière quantitative. Enfin, la dernière couche reprend l'idée des objets perdurants avec l'emploi de timeslices et de *fluents* proposée dans l'approche 4D-fluents. Toutefois, certaines différences sont à relever entre l'approche 4D-fluent et TOWL. Tout d'abord, dans TOWL, les timeslices ne sont pas obligatoirement associés à un intervalle de temps mais peuvent également être associées à un instant de temps. Lors de la définition d'un fluent entre deux timeslices, l'approche 4D-fluents obligeait les deux timeslices à être associées au même intervalle de temps. Avec TOWL, les timeslices peuvent avoir leur propre intervalle de temps, cependant, lorsqu'un fluent est défini entre deux timeslices la relation *equal* doit être définie afin de garantir que ces timeslices sont valides sur le même intervalle de temps. Un autre apport de TOWL est de spécifier deux types de *fluents*. Alors qu'avec 4D-fluent, nous avons seulement une propriété *fluentProperty*, TOWL distingue les *fluentObjectProperty* (voir figure 2.24) et les *fluentDatatypeProperty* (voir figure 2.23). Une sous-propriété de *fluentObjectProperty* relie deux timeslices tandis qu'une sous-propriété de *fluentDatatypeProperty* relie un timeslice à un objet de type *rdfs:Literal*.

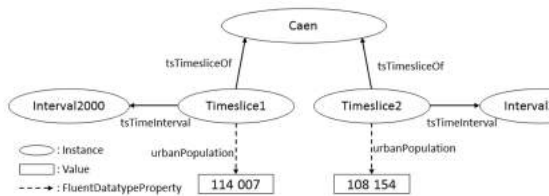


FIGURE 2.23 – Fluent Datatype Property

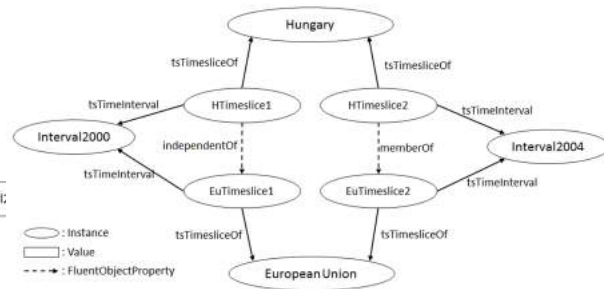
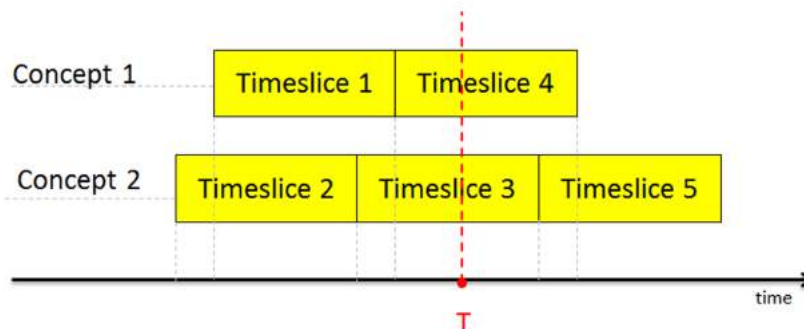


FIGURE 2.24 – Object Datatype Property

ral et indique donc des changements de valeurs. L'emploi d'une propriété de type *fluentObjectproperty* implique que les timeslices reliés par la propriété soient valides sur le même intervalle de temps. La relation *equal* sera donc utilisée entre ces intervalles de temps. Pour cela, les intervalles de temps doivent être bien définis, c'est-à-dire posséder un instant de début et un instant de fin clairement spécifiés. L'obligation de fournir des intervalles bien définis permet de détecter des inconsistances, par exemple, si les intervalles de temps de deux timeslices liés par un *fluent* ne sont pas exactement les mêmes. Mais cette obligation permet en outre de résoudre un problème connu sous le nom de *Frame Problem* [Milea et al., 2008], qui se définit comme la capacité à déterminer à n'importe quel instant ce qui est valide et ce qui ne l'est pas. Dans le cadre d'une application spatio-temporelle, le *Frame Problem* est un enjeu majeur car il permet de visualiser un espace géographique à n'importe quel instant. La figure illustre le *Frame Problem* 2.25.

FIGURE 2.25 – Illustration du *Frame Problem*

2.6/ DISCUSSION

L'émergence d'outils et méthodes en faveur du développement de standards de modélisation de l'information spatio-temporelle au sein des technologies du Web Sémantique est quasi inexistante à l'heure actuelle. Toutefois, la gestion du temps et de l'évolution d'une part, et le support d'information spatiale d'autre part ont bénéficié de quelques propositions dans la littérature du Web Sémantique. Ces propositions ont

été étudiées dans la section précédente. Des modèles tels que SWRL Temporal Ontology [O'Connor et al., 2011] ou OWL-Time [Hobbs et al., 2004] permettent d'introduire des données temporelles mais ne peuvent gérer l'évolution, c'est pourquoi des ontologies de fluents comme 4D-fluents ou TOWL ont fait leur apparition. Toutefois, bien que les ontologies de fluent soient capables de définir plusieurs états pour une même entité au cours du temps, elles ne proposent pas de mécanismes permettant de lier ces états dans le temps afin d'étudier le cycle de vie de ces entités. Les outils de représentation spatiale tels que GeoRSS, GeoOWL et plus récemment GeoSPARQL ont ouvert la voie au développement d'ontologies spatialisées mais ne proposent aucune solution au besoin de modéliser la dynamique d'entité spatialisée. Le modèle proposé dans ces travaux exploite les forces des modèles sémantiques de la littérature afin de répondre aux besoins la modélisation spatio-temporelle et dépasser les limitations des approches classiques en terme de représentation des connaissances. En outre, le modèle constitue un outil adapté pour la mise en relation de l'utilisation des terres et de la couverture des sols et assister les chargés de territoire dans la prise de décision.

2.6.1/ APPORT POUR LE DOMAINE DE LA MODÉLISATION SPATIO-TEMPORELLE

Les systèmes d'informations géographiques sont conçus pour permettre le stockage, la gestion, l'interrogation, le partage et la diffusion d'informations géographiques. Traditionnellement, les informations géographiques gérées par des SIG spatio-temporels sont définies selon trois composantes :

- la composante **descriptives** : qui regroupe l'ensemble des attributs et relations permettant de décrire un objet géographique.
- la composante **spatiales** : qui regroupe les données relatives à la géométrie, la localisation ou les relations topologiques de l'objet.
- la composante **temporelle** : qui permet d'exprimer la dynamique et l'évolution des deux composantes précédentes.

Dans ces travaux, nous introduisons la composante d'**identité** qui permet d'étudier le cycle de vie d'un objet ainsi que ses éventuelles altérations en objets dérivés. En ce sens, l'identité possède également une dynamique qui peut être exprimée à l'aide de la composante temporelle. Chacune des composantes peut évoluer indépendamment ou conjointement. Pour les raisons citées précédemment, dans ces travaux, nous introduisons une ontologie spatio-temporelle sémantique comme modèle pour l'analyse de phénomènes dynamiques géospatiaux.

Notre modèle est une ontologie géographique spatialisée (ou spatio-temporelle). La définition d'une telle ontologie répond à des besoins particuliers de la modélisation spatio-temporelle qui peuvent être identifiés comme suit :

- définir une représentation spatiale des entités géographiques de l'ontologie. Ces représentations spatiales peuvent être des lignes, des points, des polygones ou d'autres types de données spatiaux plus complexes.
- définir des relations spatiales telles que les relations topologiques.

- introduire des données temporelles permettant de localiser dans le temps l'intervalle ou l'instant de validité des entités géographiques.
- définir plusieurs états pour une même entité au cours du temps afin de stocker l'évolution de celle-ci ainsi que ses propriétés et relations.
- lier les différents états d'une entité au cours du temps afin de retracer le cycle de vie de chaque entité.
- fournir un outil capable de raisonner sur la spatialité, la temporalité et la sémantique des entités géographiques.
- autoriser l'extraction de motif générique et spécifique pour découvrir et qualifier les phénomènes.

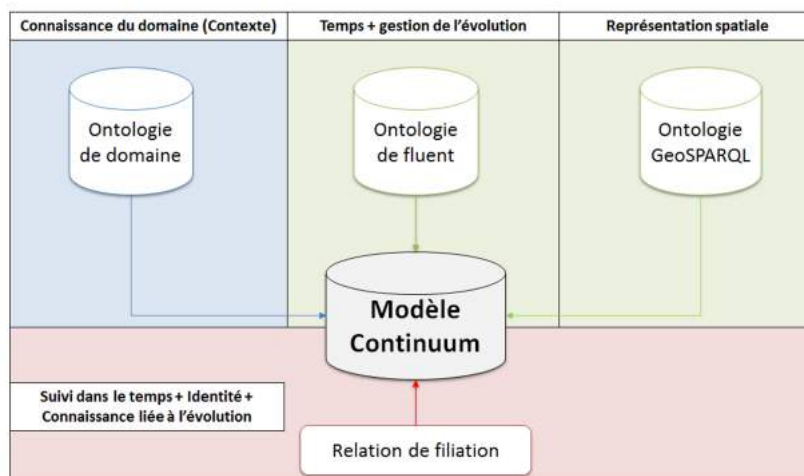


FIGURE 2.26 – Présentation du modèle

La figure 2.26 illustre les grandes composantes du modèle Continuum. Ainsi, notre modèle reprend les fondamentaux des ontologies de fluent ainsi que des modèles de représentation spatiale afin de répondre aux exigences de la modélisation spatio-temporelle. En outre, la connaissance et le contexte lié à l'environnement géospatial sont stockés au sein d'une ontologie de domaine. Cette ontologie de domaine représente la connaissance métier que l'utilisateur pourra utiliser afin de définir des règles d'inférence pour améliorer la compréhension de l'environnement et de sa dynamique. Enfin, en reprenant le concept de relation de filiation introduite dans [Del Mondo et al., 2013], notre modèle permet de lier les différents états d'une entité au cours de son cycle de vie permettant ainsi de combler une des lacunes des ontologies de fluent dans le cadre de la modélisation spatio-temporelle. Outre le simple fait d'obtenir une meilleure visualisation du cycle de vie des entités du modèle, nous déclinons cette relation de filiation en une hiérarchie de subsomption dans laquelle les sous-propriétés sont progressivement enrichies sémantiquement. Les sous-propriétés les plus spécifiques sont définies en fonction

de contraintes de temps, d'espace et d'identité, et correspondent à des motifs d'évolution servant de support à l'inférence de phénomènes géospatiaux.

2.6.2/ APPORT POUR LE DOMAINE DE LA DYNAMIQUE D'UTILISATION DES TERRES ET DES COUVERTURES TERRESTRES

L'étude du changement distingue deux types d'évolutions, celle de l'utilisation des terres et celle de la couverture des sols. La couverture des terres réfère à la couverture physique et biologique d'un territoire, il peut s'agir d'eau, de végétation, ou encore de surfaces artificielles. L'utilisation des terres est un terme plus complexe faisant référence aux activités humaines telles que l'agriculture, l'exploitation forestière, ou la construction de bâtiments. Les notions de contexte économique et social sont parfois inclus pour définir l'utilisation des terres. Les modèles de changement d'utilisation et de couverture des sols ont été développés pour répondre aux problématiques de quand, où et pourquoi ces changements d'utilisation et de couverture sont intervenus. Généralement, les couvertures de terres sont entièrement définies au sein des modèles. En revanche, les données liées à l'usage des terres sont plus complexes et nécessitent une connaissance experte pour être mise en corrélation avec l'évolution des couvertures. C'est pourquoi l'analyse des causes et des conséquences des phénomènes d'un territoire requiert l'intervention d'un utilisateur expert.

Une première façon d'étudier l'évolution du territoire consiste à évaluer les changements sur chaque couverture terrestre. Cette méthode fait aujourd'hui autorité lorsqu'il s'agit de détecter ou d'estimer les changements de couvertures sur un territoire donné [Mas, 1999], [Ramankutty et al., 1999], [Caldas et al., 2015]. Le tableau 2.3 montre le résultat de cette approche sur les classes du niveau 3 du jeu de données Corine Land cover.

Deux types d'informations majeures ressortent de cette analyse. D'une part, nous détectons facilement l'expansion ou la diminution pour chacun des types de couverture. D'autre part, nous pouvons déterminer le taux d'expansion ou de diminution entre deux instants. Mais, cette approche ne permet pas d'expliquer la causalité de l'évolution. Tout d'abord, les résultats obtenus ne décrivent l'évolution qu'à une échelle globale excluant ainsi toute analyse plus détaillée. Deuxièmement, la zone étudiée pouvant être considérée comme un territoire composé de parcelles possédant chacune un type couverture, l'expansion ou la diminution d'une parcelle implique nécessairement la diminution ou l'expansion d'une parcelle voisine donnant lieu un ré-agencement partiel des parcelles du territoire entre deux instants de temps consécutifs. En conséquence, il peut s'avérer difficile de retrouver la causalité et la relation de filiation existant entre deux parcelles évoluant entre deux instants consécutifs. Pour autant, la recherche et l'analyse de ce lien de causalité pour chacune des parcelles est un élément essentiel de notre compréhension de l'évolution. En dernier point, une analyse plus détaillée permettrait de localiser précisément les phénomènes spécifiques dans une région donnée plutôt qu'une tendance globale sur un territoire. Pour passer de l'échelle globale à une échelle détaillée, nous tentons de retrouver l'évolution de chaque parcelle en se basant sur les relations de filiation.

Le développement d'ontologies formelles pour la modélisation et l'analyse de phénomène dynamique offre des capacités accrues pour assister les scientifiques et les chargés de territoire dans les prises de décision. Par exemple, le développement intensif

TABLE 2.3 – Calcul de l'aire pour chaque classe de CORINE land cover au niveau 3.

	Aire 1990 (Km ²)	Aire 2000 (Km ²)	Aire 2006 (Km ²)	% Chan- gement de 1990 à 2000	% Chan- gement de 2000 à 2006
BeachDune	30.8	61.1	60.6	98.4	-0.8
ConiferousForest	7366.4	7862.5	6187.6	6.7	-21.3
Vineyards	1395	1481.2	1558.4	6.2	5.2
NaturalGrassLands	15.3	112.7	20.2	636.6	-82.1
Pastures	597.1	644	658.1	7.9	2
Ports	1.6	1.6	2.9	0	81.3
UrbanGreen	10.3	10	8.3	-2.9	-17
Marshes	49.6	51.2	61.5	3.2	20.1
Salines	4.7	4.7	0	0	-100
IndCommercial	71.5	83.7	87	17.1	3.9
BroadLeavedForest	434.5	639	642.2	47.1	0.5
FruitPlantation	14.8	16	15.7	8.1	-1.9
VegMoors	1.8	4.2	121.7	133.3	2797.6
Estuaries	380.2	380.2	389.2	0	2.4
WaterCourses	250.8	250.8	251	0	0.1
ShrubWoodland	513.7	665.9	1830.5	29.6	174.9
Airports	6.9	9.5	11.4	37.7	20
IntertidalFlats	114.9	136.2	136.7	18.5	0.4
ArableNonIrrigated	2163.6	2492.2	2087.3	15.2	-16.2
UrbanContinuous	25.4	26.2	23.6	3.1	-9.9
RoadsRails	7.3	7.7	8.5	5.5	10.4
Construction	0.6	9.2	0	1433.3	-100
SeasOceans	24042.6	24046.4	24046.7	0	0
Sports	28.5	35.2	36.3	23.5	3.1
Mines	16.2	23.6	24.5	45.7	3.8
UrbanDiscontinuous	397.3	483.8	513.4	21.8	6.1
AgricAndNatural	72.3	109.9	115.5	52	5.1
ComplexCultivation	853.1	1018.9	1126.8	19.4	10.6
InlandWaterBodies	80.6	139.2	143.8	72.7	3.3
MixedForest	432.9	622.1	631.6	43.7	1.5
SaltMarshes	30.8	31.3	36.8	1.6	17.6
Dumps	0	0	1.1	0	0
ArablePermlrrigated	0	0	10.5	0	0

des zones industrielles et commerciales, extensions urbaines ou encore infrastructures routières ont entraîné une perte conséquente de terres agricoles en France. Ce phénomène est connu sous le nom d'artificialisation et entraîne donc la raréfaction des espaces agricoles cultivés. Combiné à l'accroissement de la population mondiale, ce phénomène pourrait poser des problèmes alimentaires à l'avenir [Laroche et al., 2006], [Pointereau et al., 2009]. C'est pourquoi l'artificialisation est devenue une préoccupation de certaines instances de l'ONU ainsi que de nombreux gouvernements.

La figure 2.27 illustre un phénomène d'artificialisation sur un territoire quelconque. Il existe plusieurs sortes de sols artificialisés et également plusieurs sortes de terres agricoles. En logique de description, il est possible de les représenter selon les équations suivantes :

$$SolsArtificialisés \equiv ZoneIndustrielleCommerciale \sqcup ZoneUrbaine \quad (2.11)$$

$$TerresAgricoles \equiv SolsFriche \sqcup SolsCultivés \sqcup SolsBoisés \sqcup Prairies \quad (2.12)$$

Il est intéressant de noter que les langages basés sur la logique de description permettent aisément de regrouper ou dissocier les parcelles grâce à la relation de subsomption. Par exemple, il est possible de récupérer toutes les parcelles de type

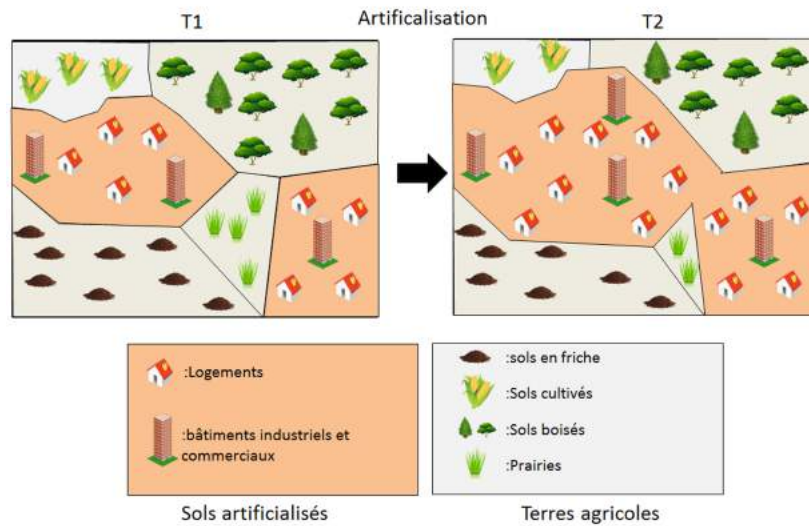


FIGURE 2.27 – Illustration d'un phénomène d'artificialisation

Zone Urbaine et de type *sols artificialisés* sans nécessiter de traitements particuliers. Ceci est particulièrement utile dans notre exemple, car l'étude des phénomènes d'artificialisation nécessite le regroupement de différents types de parcelles au sein de classes plus génériques. Pour inférer le phénomène d'artificialisation, les SIG traditionnels ont recours à l'usage de données statistiques comme illustré précédemment par le tableau 2.3. Ainsi, sur un territoire donné, le système calculera les pourcentages d'augmentation ou de diminution de la surface des *sols artificialisés* et des *terres agricoles*. Dès lors, de manière schématique, il est possible de définir l'artificialisation de la façon suivante :

SI il y a une augmentation des Sols Artificialisés et une diminution des Terres Agricoles
ALORS il y a un phénomène d'Artificialisation

Ce type d'analyse statistique, bien que largement utilisée, ne permet pas d'établir clairement les liens entre chacune des parcelles du territoire. Dès lors, l'analyse n'est pertinente qu'à une échelle globale mais ne permet pas de localiser précisément les lieux où le phénomène se produit. Dans notre modèle, bien que l'analyse statistique puisse être fournie, nous nous focalisons sur l'analyse des liens logiques. Plus particulièrement, nous nous intéresserons à la relation de filiation qui décrit le lien de succession entre des parcelles entre deux instants de temps. Dès lors, l'analyse ne se base, non plus sur les valeurs numériques associées aux parcelles, mais sur la sémantique des liens qu'entretiennent ces parcelles entre elles. Ces liens sont représentés par un graphe de relation. La figure 2.28 illustre le graphe sémantique associé à l'exemple de la figure 2.27.

L'exemple de notre figure décrit trois relations de filiation que sont la diminution, l'augmentation et la dérivation. Une relation de filiation n'est établie qu'entre des parcelles possédant une intersection spatiale entre des instants $T1$ et $T2$. La dérivation, quant à elle, décrit un changement de l'identité. Dès lors, à l'aide d'une règle d'inférence, notre système est capable de détecter un phénomène d'artificialisation dans des régions précises de l'environnement géospatial. L'équation suivante décrit cette règle :

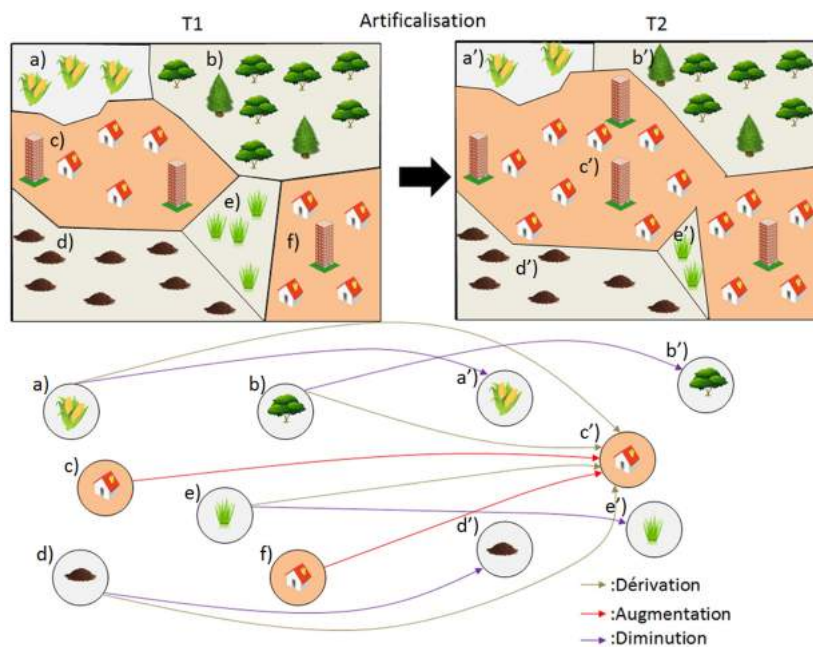


FIGURE 2.28 – Illustration d'un phénomène d'artificialisation représenté au sein d'un graphe sémantique

$$\begin{aligned}
 & \text{TerresAgricoles}(?x) \wedge \text{diminution}(?x, ?x') \wedge \text{SolsArtificialisés}(?y) \wedge \\
 & \text{augmentation}(?y, ?y') \wedge \text{derivation}(?x, ?y') \rightarrow \text{Artificialisation}(?x, ?y')
 \end{aligned}
 \tag{2.13}$$

Littéralement, cette règle signifie que s'il existe une dérivation entre deux parcelles consécutives avec la première parcelle de type *TerresAgricoles* ayant subi une diminution de sa surface et la deuxième parcelle de type *SolsArtificialisés* ayant subi une augmentation de sa surface, alors il existe un phénomène d'artificialisation entre ces deux parcelles. Ainsi l'analyse du phénomène permet de localiser précisément chaque parcelle impliquée. Pour fournir une analyse à une plus large échelle, il est alors pertinent de recourir à l'analyse statistique en se basant sur les phénomènes inférés entre les parcelles ce qui permet, non seulement de déterminer le pourcentage d'augmentation ou de diminution de la surface mais également de déterminer le ou les endroit(s) précis où le phénomène est le plus marqué. Selon l'ampleur du phénomène révélée par notre système, les gérants de territoire peuvent décider ou non de mettre en place des mesures afin de limiter les conséquences de ce dernier. Dès lors, les moyens d'action possibles peuvent être de densifier la construction des infrastructures ou encore maîtriser le canevas de construction afin de préserver au mieux les terres nécessaires à l'agriculture. Une fois les moyens d'action mis en place, le résultat est ensuite réintroduit dans le modèle au travers d'une nouvelle couche temporelle. Ainsi, de nouveaux traitements et analyses peuvent être appliqués et permettre d'évaluer l'efficacité des moyens d'actions utilisés.



CONTRIBUTION

CONTINUUM : UN MODÈLE SÉMANTIQUE SPATIO-TEMPOREL

Ce chapitre présente le modèle Continuum permettant de capturer l'évolution des entités d'un parcellaire afin de découvrir des phénomènes dynamiques. Les données sont stockées dans le modèle au sein de plusieurs couches temporelles. Une couche temporelle représente un territoire composé d'un ensemble de parcelle à un instant t . Les couches temporelles consécutives sont ensuite connectées via une transition spatio-temporelle. La transition spatio-temporelle décrit la façon dont chaque parcelle d'une couche temporelle t_1 est liée à une ou plusieurs parcelles d'une couche temporelle t_2 consécutive. Dans le modèle continuum, ce lien est associé à la relation de filiation et est décliné selon plusieurs couches d'expressivités permettant d'offrir de nouveaux outils pour l'étude de l'évolution spatio-temporelle d'un environnement géographique.

Sommaire

3.1 Définition d'une couche spatio-temporelle	75
3.1.1 Organisation hiérarchique des concepts	76
3.1.2 Représentation du temps : intervalles et instants	76
3.1.3 Les quatre composantes d'une entité	77
3.2 Définition d'une transition spatio-temporelle	78
3.2.1 Expression des contraintes d'intégrité des relations de filiation	80
3.2.2 Contraintes d'intégrité sur les primitives du modèle Continuum	89

3.1/ DÉFINITION D'UNE COUCHE SPATIO-TEMPORELLE

Le modèle Continuum tente de représenter les entités dynamiques évoluant dans le temps. Ces entités seront nommées "*timeslices*" dans le cadre de ces travaux. Chaque timeslice se définit selon quatre composantes que sont : l'identité, l'espace, le temps, et la sémantique intrinsèque de l'entité. L'identité est la composante la plus importante du modèle. Généralement, lorsque la modélisation porte sur des parcelles de terrain, une classe correspond à une couverture du sol spécifique et définit l'identité du *timeslice*. Cependant d'autres critères peuvent être utilisés pour souligner l'unicité d'un *timeslice*. Les ontologies possèdent la particularité de pouvoir organiser des classes sur différents niveaux hiérarchiques en utilisant une taxonomie comme colonne vertébrale. Chaque classe décrit un concept et la taxonomie permet d'associer les *timeslices* à des concepts plus ou moins spécifiques.

$$Timeslice \sqsubseteq \top \quad (3.1)$$

La formule 3.1 présente le concept timeslice en logiques descriptives.

3.1.1/ ORGANISATION HIÉRARCHIQUE DES CONCEPTS

Dans notre modèle, la classe *Timeslice* désigne le concept plus général et peut être spécialisée par une hiérarchie. Dès lors, les concepts spécifiques sont utiles pour distinguer les entités représentées tandis que les concepts généraux permettent au contraire de les regrouper. La formule suivante formalise la hiérarchie.

$$C_n \sqsubseteq \dots \sqsubseteq C_i \sqsubseteq \dots \sqsubseteq C_0 \quad (3.2)$$

où C_0 correspond à la classe *Timeslice* et n représente la profondeur de la hiérarchie.

3.1.2/ REPRÉSENTATION DU TEMPS : INTERVALLES ET INSTANTS

Afin de représenter le temps, le modèle se base sur l'approche suggérée par [Artale et al., 1998] en considérant le domaine temporel comme une structure linéaire composée par un ensemble de points temporels (*TemporalPoint*).

$$TemporalPoint \sqsubseteq \top \quad (3.3)$$

Tous les éléments de type *TemporalPoint* suivent un ordre strict, qui oblige tous les points entre deux instants temporels t_1 et t_2 à être ordonnés dans le temps. En sélectionnant une paire de points temporels $[t_o, t_f]$, il est possible de définir un intervalle fermé de points ordonnés définissant ainsi des intervalles de temps (*Interval*).

$$\begin{aligned} Interval &\sqsubseteq \top \\ Interval &\equiv (= 1hasStartPoint.TemporalPoint) \sqcap \\ & \quad (= 1hasEndPoint.TemporalPoint) \end{aligned} \quad (3.4)$$

Cette formule précise que tout objet *Interval* possède une seule propriété *hasStartPoint* et une seule propriété *hasEndPoint* de type *TemporalPoint*. Les propriétés sont, par nature, fonctionnelles.

Dans certaines applications, la structure du temps peut également être définie selon des instants de temps (*TemporalPoint*). Afin de représenter à la fois des intervalles et des instants de temps, nous définissons le concept de temps (\mathcal{T}).

$$\mathcal{T} \equiv TemporalPoint \sqcup Interval \quad (3.5)$$

La formule 3.5 définit qu'un temps \mathcal{T} est soit un objet de type *Interval* soit un objet de type *TemporalPoint*. La formule 3.6 précise que *Interval* et *TemporalPoint* sont des concepts disjoints.

$$\textit{TemporalPoint} \sqcap \textit{Interval} \equiv \perp \quad (3.6)$$

Dans le modèle, la propriété *hasTime* possède comme *domain* la classe \mathcal{T} , ainsi il est possible d'utiliser les classes *TemporalPoint* et *Interval* selon les besoins de modélisation :

$$\top \sqsubseteq \forall \textit{hasTime}.\mathcal{T} \quad (3.7)$$

Cette sous-section a présenté la modélisation du temps en logique de description. La section suivante présente les quatre composantes d'une entité.

3.1.3/ LES QUATRE COMPOSANTES D'UNE ENTITÉ

Dans ces travaux, les différents états des entités spatio-temporelles sont représentés par un objet de type \mathcal{TS} (timeslice). Cette classe comprend quatre composantes : 1) spatiale, qui est la représentation géométrique de l'entité (\mathcal{G}) ; 2) identité, qui associe chaque état à l'entité (\mathcal{O}) qu'il représente ; 3) temporelle, pour décrire le temps (\mathcal{T}) durant lequel le *timeslice* est valide ; 4) un ensemble de propriétés alphanumériques, qui décrivent les caractéristiques de l'entité durant la période de validité du *timeslice*. La formule 3.8 représente la formalisation de la classe \mathcal{TS} .

$$\begin{aligned} \mathcal{TS} \equiv & (= 1\textit{hasGeometry}.\mathcal{G}) \sqcap \\ & (= 1\textit{isTimesliceOf}.\mathcal{O}) \sqcap \\ & (= 1\textit{hasTime}.\mathcal{T}) \sqcap \\ & (\exists \textit{hasProperties}.\overline{\mathcal{TS}}) \end{aligned} \quad (3.8)$$

La propriété *hasGeometry* permet de lier une géométrie au *timeslice* et de définir sa composante spatiale. La propriété *isTimesliceOf* permet de décrire l'identité du *timeslice*. La propriété *hasTime* décrit la composante du *timeslice*. Enfin, *hasProperties* représente un ensemble de propriétés permettant décrire les caractéristiques thématiques du *timeslice* indépendantes des propriétés définies précédemment. Les propriétés *hasGeometry*, *isTimesliceOf* et *hasTime* doivent impérativement être décrites de manière unique pour chacun des *timeslices*. C'est pourquoi chacune d'entre elles est définie comme fonctionnelle. Les propriétés thématiques, en revanche, sont facultatives dans la description du *timeslice*. La figure 3.1 illustre le concept de *timeslice* ainsi que ses composantes.

Une fois les couches définies, il est possible d'exprimer les transitions spatio-temporelles qui feront l'objet de la section suivante.

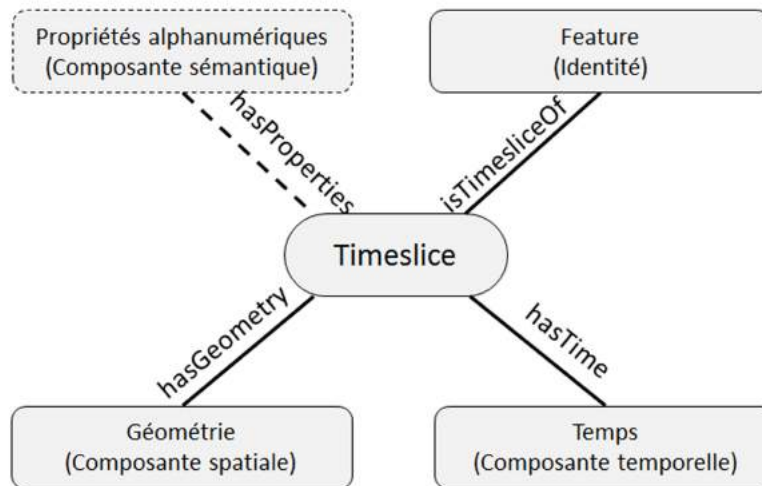


FIGURE 3.1 – Les quatre composantes du timeslice

3.2/ DÉFINITION D'UNE TRANSITION SPATIO-TEMPORELLE

Dans une zone géographique avec une couverture dynamique du territoire, une même région peut être associée à différents timeslices et à différents instants de temps. Afin de représenter ces associations spatiales dans le temps, nous définissons la relation *filiation* dans notre modèle. Cette relation permet de connecter deux timeslices consécutifs dans le temps. Par conséquent, la classe *Timeslice* est définie à la fois comme *domaine* (voir formule 3.9) et comme *co-domaine* (voir formule 3.10).

$$\exists hasFiliation \sqsubseteq \mathcal{T}S \quad (3.9)$$

$$\top \sqsubseteq \forall hasFiliation. \mathcal{T}S \quad (3.10)$$

Cette propriété est essentielle pour établir un lien spatio-temporel entre deux entités. Cependant, ce simple lien ne permet pas d'appréhender correctement le phénomène impliquant les entités connectées. Un traitement supplémentaire s'avère nécessaire pour bien comprendre l'évolution. Dans le modèle Continuum, la relation de filiation est spécialisée au travers de différentes couches connaissances. Ainsi chaque spécialisation de cette relation offre une connaissance plus approfondie de l'évolution d'une entité. A terme, la relation de filiation peut être assimilée à un phénomène du monde réel. La figure 3.2 illustre la notion de transition dans le modèle Continuum. Une transition est décrite par la relation de filiation au sein du modèle. La relation de filiation est inférée et déclinée en plusieurs couches d'expressivité en se basant sur les composantes temporelle, identitaire, spatiale et thématique des timeslices.

Chacune des couches d'expressivité définie sur la relation de filiation sera expliquée dans la suite de cette section.

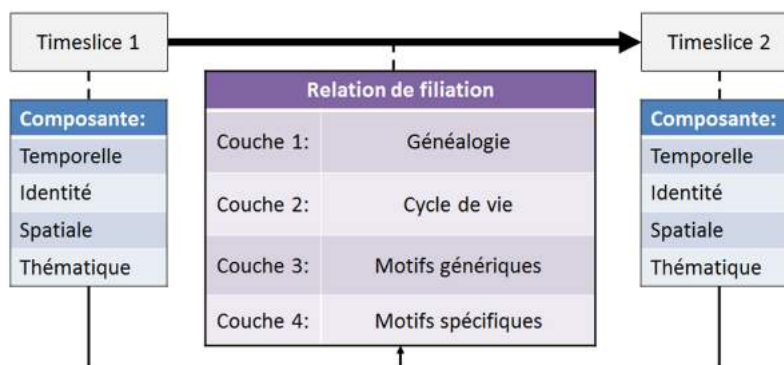


FIGURE 3.2 – Transition entre deux timeslices

Les formules 3.11, 3.12 et 3.13 formalisent cette hiérarchie de connaissance en utilisant les Logiques de Description.

$$\begin{aligned} &hasFiliation \equiv \\ &hasContinuation \sqcup hasDerivation \end{aligned} \quad (3.11)$$

Afin de spécialiser nos relations de filiation, nous intégrons tout d'abord les contraintes d'identité pour distinguer les objets ayant changé leurs natures entre deux instants consécutifs. Ainsi, les relations de filiation sont divisées en relation de continuation ou de dérivation.

$$\begin{aligned} &hasContinuation \equiv \\ &hasEquality \sqcup hasGrowth \sqcup hasReduction \sqcup \\ &hasAnnexion \sqcup hasSeparation \end{aligned} \quad (3.12)$$

$$\begin{aligned} &hasDerivation \equiv \\ &Conversion \sqcup hasSplit \sqcup hasFusion \sqcup \\ &partOfAnnexion \sqcup partOfSeparation \end{aligned} \quad (3.13)$$

L'ajout de contraintes spatiales permet de spécialiser les relations de continuation et de dérivation au sein d'une nouvelle couche. Chacune des relations définies dans cette couche constitue un motif d'évolution spatio-temporelle. Etant applicable sur n'importe quel jeu de données indépendamment du contexte, ces motifs sont dits "**génériques**" et serviront de support afin de qualifier des phénomènes du monde réel en utilisant les connaissances stockées dans l'ontologie de domaine. Ces phénomènes pourront également être trouvés sous l'appellation de motifs "**spécifiques**", c'est à dire des motifs dépendant du domaine d'application. Pour définir nos motifs génériques, nous utiliserons les relations définies dans le modèle DE-9IM (Equal, Within, Contains) pour définir les contraintes spatiales [Strobl, 2008]. De même, nous utiliserons les relations définies

par Allen dans [Allen, 1983] pour définir des contraintes temporelles (Meets, Equal, ...). Afin de distinguer l'égalité spatiale et temporelle, nous avons décidé d'adopter le terme *hasEqual* en cas d'égalité spatiale.

Dans le modèle Continuum, un changement sur la composante spatiale ou sur la composante sémantique génère un nouveau timeslice. D'une part, ce nouveau 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 de temps du timeslice enfant.

Une relation de filiation entre deux timeslices consécutifs ts_1 et ts_2 est défini par les relations entre leurs représentations spatiales (g_p et g_c), leurs définitions sémantiques (s_p et s_c), leurs identités (o_p et o_c) et leurs intervalles de temps (i_p et i_c). Par conséquent, une relation de filiation intervient lorsqu'un changement se produit sur la géométrie, la composante sémantique ou l'identité d'un timeslice. A noter que les lettres p et c ont la signification suivante : p pour "parent" et c pour "child".

3.2.1/ EXPRESSION DES CONTRAINTES D'INTÉGRITÉ DES RELATIONS DE FILIATION

Les relations de filiation sont soumises à des contraintes. Afin de formaliser ces contraintes, nous considérons des interprétations qui consistent en un ensemble non-vide Δ^I (le domaine d'interprétation) et une fonction d'interprétation qui associe à chaque concept atomique A un ensemble $A^I \subseteq \Delta^I$, à chaque rôle atomique R une propriété objet (R) $R^I \subseteq \Delta^I \times \Delta^I$ et à chaque propriété de type de données (U) un ensemble $U^I \subseteq \Delta^I \times \Delta_{\mathcal{D}}^I$ avec \mathcal{D} le co-domaine des données défini par $D^{\mathcal{D}} \subseteq \Delta_{\mathcal{D}}^I$. Dans un souci d'expressivité, nous définirons nos contraintes en logique du premier ordre. La définition des contraintes nécessitent de décrire et utiliser des opérateurs temporels et spatiaux (voir formules 3.14 et 3.15).

$$Meets^I = \{\langle \mathcal{T}_i, \mathcal{T}_{i+1} \rangle \mid i \in \mathbb{N}\} \text{ et } \left\{ \begin{array}{l} \text{si } \mathcal{T}_i, \mathcal{T}_{i+1} \in TemporalPoint \\ \forall \mathcal{T}_i, \mathcal{T}_{i+1} \rightarrow succ(\mathcal{T}_i) = \mathcal{T}_{i+1} \\ \\ \text{si } \forall a, b \in TemporalPoint^I \\ \text{si } \forall I, I' \in Interval^I \\ \text{si } (I, a) \in hasEndPoint \\ \text{et } (I', b) \in hasStartPoint \\ \rightarrow a = b \end{array} \right. \quad (3.14)$$

$$hasEqualGeometry^I = \{\langle g, g' \rangle \mid (\forall x(x \in g \rightarrow x \in g')) \wedge (\forall x(x \in g' \rightarrow x \in g)), g, g' \in \mathcal{G}\}$$

$$Within^I = \{\langle g, g' \rangle \mid (\forall x(x \in g \rightarrow x \in g')) \wedge (\exists x(x \in g' \wedge x \notin g)), g, g' \in \mathcal{G}\}$$

$$Union^I = \{\langle g_1, g_2, g' \rangle \mid (\forall x(x \in g_1 \vee x \in g_2 \rightarrow x \in g')), g_1, g_2, g' \in \mathcal{G}\} \quad (3.15)$$

L'opérateur *Meets* permet de décrire que deux instants ou deux intervalles de temps sont contigus dans le temps. S'il s'agit d'instant alors l'opérateur décrit la succession de deux instants. S'il s'agit d'intervalle alors l'instant de fermeture du premier intervalle doit être

égal à l'instant de départ du deuxième intervalle. L'opérateur *hasEqualGeometry* définit l'égalité entre deux géométries. L'opérateur *Within* décrit l'inclusion d'une géométrie g dans une géométrie g' . Enfin, l'opérateur *Union* définit l'union de deux géométries.

En complément de ces opérateurs, nous définissons la propriété "hasSemanticProperty" représentant toutes les propriétés sémantiques (propriétés objets ou type de données) d'un timeslice à l'exception de "hasGeometry", "hasTime" et "isTimesliceOf". La formule 3.16 formalise ce composant sémantique (\mathcal{S}) :

$$hasSemanticProperty^I = \{ \langle s, s' \rangle \mid \forall s' (\forall s' (\langle s, s' \rangle \in R^I \vee \langle s, s' \rangle \in U^I) \wedge (\neg hasGeometry(s, s') \wedge \neg hasTime(s, s') \wedge \neg isTimesliceOf(s, s'))) \}, s, s' \in (R^I \sqcup U^I) \} \quad (3.16)$$

3.2.1.1/ CONTRAINTES D'INTÉGRITÉ DE LA PREMIÈRE COUCHE DU MODÈLE : LA RELATION DE FILIATION

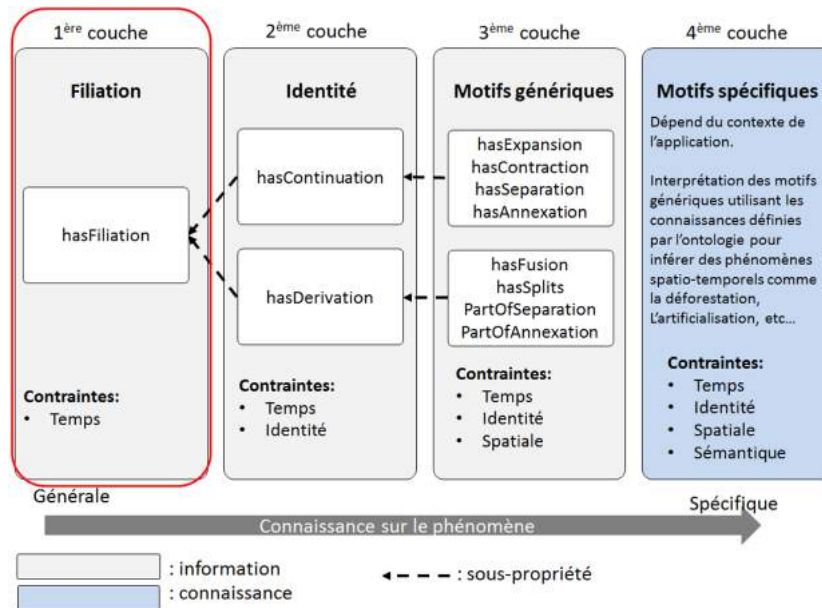


FIGURE 3.3 – Description de la couche filiation

A présent, il est possible de décrire les contraintes de chacune des relations de filiation de notre modèle. La formule 3.17 décrit la plus générale d'entre elles : la relation *hasFiliation*

$$\begin{aligned} & [\forall p (\forall c (hasFiliation(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \rightarrow \\ & ((\exists g_p \exists g_c (hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \rightarrow \neg hasEqual(g_p, g_c))) \\ & \vee (\exists o_p \exists o_c (isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \rightarrow o_p \neq o_c) \vee \\ & (\exists s_p \exists s_c (SemanticProperties(p, s_p) \wedge SemanticProperties(c, s_c) \rightarrow s_p \neq s_c)) \wedge \\ & (\exists i_p \exists i_c (hasTime(p, i_p) \wedge hasTime(c, i_c) \rightarrow meets(i_p, i_c)))))))] \end{aligned} \quad (3.17)$$

La formule 3.17 peut se décomposer en 5 blocs pour faciliter la compréhension :

- **REL_{Filiation}** :

$$\forall p(\forall c(\text{hasFiliation}(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c)) \quad (3.18)$$

Le bloc $REL_{Filiation}$ définit qu'une relation de filiation ne peut être établie qu'entre deux timeslices avec p le timeslice parent et c le timeslice enfant.

- **GEOM_{Filiation}** :

$$\exists g_p \exists g_c(\text{hasGeometry}(p, g_p) \wedge \text{hasGeometry}(c, g_c) \rightarrow \neg \text{hasEqual}(g_p, g_c) \quad (3.19)$$

Le bloc $GEOM_{Filiation}$ définit que les géométries du timeslice parent et du timeslice enfant respectivement g_p et g_c ne sont pas égales.

- **ID_{Filiation}** :

$$\exists o_p \exists o_c(\text{isTimesliceOf}(p, o_p) \wedge \text{isTimesliceOf}(c, o_c) \rightarrow o_p \neq o_c \quad (3.20)$$

Le bloc $ID_{Filiation}$ définit que les identités du timeslice parent et du timeslice enfant respectivement o_p et o_c ne sont pas égales.

- **SEM_{Filiation}** :

$$\exists s_p \exists s_c(\text{SemanticProperties}(p, s_p) \wedge \text{SemanticProperties}(c, s_c) \rightarrow s_p \neq s_c) \quad (3.21)$$

Le bloc $SEM_{Filiation}$ définit que les propriétés sémantiques du timeslice parent et du timeslice enfant respectivement s_p et s_c ne sont pas égales.

- **TEMP_{Filiation}** :

$$\exists i_p \exists i_c(\text{hasTime}(p, i_p) \wedge \text{hasTime}(c, i_c) \rightarrow \text{meets}(i_p, i_c) \quad (3.22)$$

Le bloc $TEMP_{Filiation}$ définit que l'instant ou l'intervalle de temps de validité du timeslice parent et du timeslice enfant respectivement i_p et i_c sont contiguës.

Ainsi, il est possible d'exprimer la formule 3.17 selon le formalisme suivant :

$$\mathbf{REL}_{Filiation} \rightarrow (\mathbf{GEOM}_{Filiation} \vee \mathbf{ID}_{Filiation} \vee \mathbf{SEM}_{Filiation}) \wedge \mathbf{TEMP}_{Filiation}$$

3.2.1.2/ CONTRAINTES D'INTÉGRITÉ DE LA DEUXIÈME COUCHE DU MODÈLE : LA COUCHE D'IDENTITÉ

Une relation de filiation représente alors une évolution impliquant deux timeslices parent et enfant consécutifs dans le temps. Cette évolution peut intervenir sur la composante spatiale, ou encore la composante d'identité ou bien la composante sémantique du timeslice. Cette relation est ensuite spécialisée afin de porter une connaissance plus précise à propos de l'évolution d'une entité. La spécialisation directe de la relation de filiation prend en compte des contraintes sur l'identité des entités constituant la deuxième couche de notre modèle.

hasContinuation : cette relation représente un changement sur la composante sémantique ou géométrique à la condition que l'identité reste inchangée.

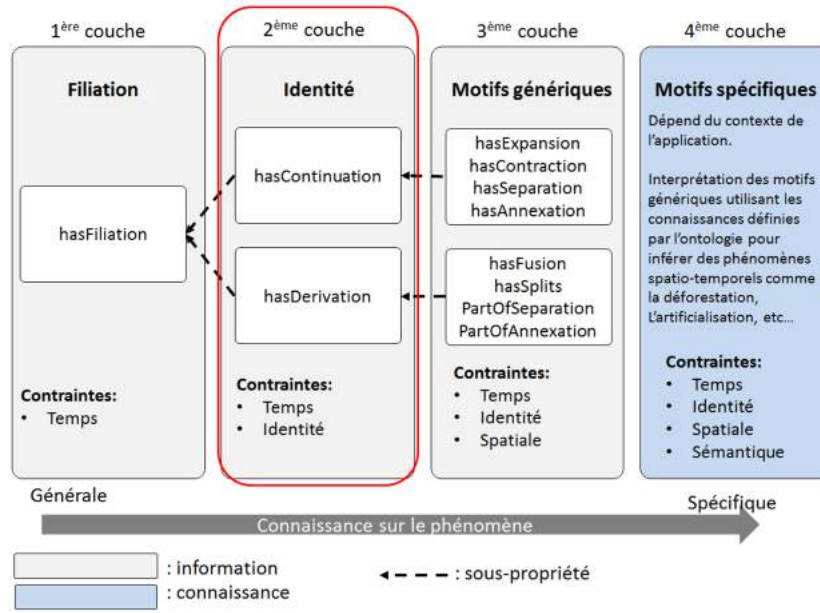


FIGURE 3.4 – Description de la couche d'identité

$$\mathbf{REL}_{\text{Continuation}} : \forall p(\forall c(\text{hasContinuation}(p, c) \wedge \mathcal{T}S(p) \wedge \mathcal{T}S(c))$$

$$\mathbf{GEOM}_{\text{Continuation}} = \mathbf{GEOM}_{\text{Filiation}}$$

$$\mathbf{ID}_{\text{Continuation}} : \exists o_p \exists o_c (\text{isTimesliceOf}(p, o_p) \wedge \text{isTimesliceOf}(c, o_c) \rightarrow o_p = o_c)$$

$$\mathbf{SEM}_{\text{Continuation}} = \mathbf{SEM}_{\text{Filiation}}$$

$$\mathbf{TEMP}_{\text{Continuation}} = \mathbf{TEMP}_{\text{Filiation}}$$

$$\mathbf{REL}_{\text{Continuation}} \rightarrow (\mathbf{GEOM}_{\text{Continuation}} \vee \mathbf{SEM}_{\text{Continuation}}) \wedge \mathbf{ID}_{\text{Continuation}} \wedge \mathbf{TEMP}_{\text{Continuation}} \quad (3.23)$$

La relation de continuation se distingue de la relation de filiation au travers de contraintes sur l'identité. Par conséquent, la seule formule qui diffère de ceux de la relation de filiation est le bloc $\mathbf{ID}_{\text{Continuation}}$. A présent, l'identité du timeslice parent doit être identique à celle du timeslice enfant. Les changements ne peuvent avoir lieu que sur la composante spatiale ou sémantique.

hasDerivation : La relation de "dérivation" est également une spécialisation directe de la relation de filiation. Cette relation représente un changement sur la composante sémantique ou géométrique. Cependant, à l'inverse de la relation de "continuation" l'identité est cette fois affectée.

$$\mathbf{REL}_{\text{Derivation}} : \forall p(\forall c(\text{hasDerivation}(p, c) \wedge \mathcal{T}S(p) \wedge \mathcal{T}S(c))$$

$$\mathbf{GEOM}_{\text{Derivation}} = \mathbf{GEOM}_{\text{Continuation}}$$

$$\mathbf{ID}_{\text{Derivation}} : \exists o_p \exists o_c (\text{isTimesliceOf}(p, o_p) \wedge \text{isTimesliceOf}(c, o_c) \rightarrow o_p \neq o_c)$$

$$\mathbf{SEM}_{\text{Derivation}} = \mathbf{SEM}_{\text{Continuation}}$$

$$\mathbf{TEMP}_{\text{Derivation}} = \mathbf{TEMP}_{\text{Continuation}}$$

(3.24)

$$\mathbf{REL}_{\text{Derivation}} \rightarrow (\mathbf{GEOM}_{\text{Derivation}} \vee \mathbf{SEM}_{\text{Derivation}}) \wedge \mathbf{ID}_{\text{Derivation}} \wedge \mathbf{TEMP}_{\text{Derivation}}$$

A l'instar de la relation de continuation, la relation de dérivation se distingue de la relation de filiation via des contraintes sur l'identité. La seule différence entre la continuation et la dérivation concerne la condition sur la formule $\mathbf{ID}_{\text{Dérivation}}$. Contrairement à la continuation, les identités des timeslices parent et enfant, respectivement o_p et o_c doivent être différentes.

3.2.1.3/ CONTRAINTES D'INTÉGRITÉ DE LA TROISIÈME COUCHE DU MODÈLE : LES MOTIFS GÉNÉRIQUES

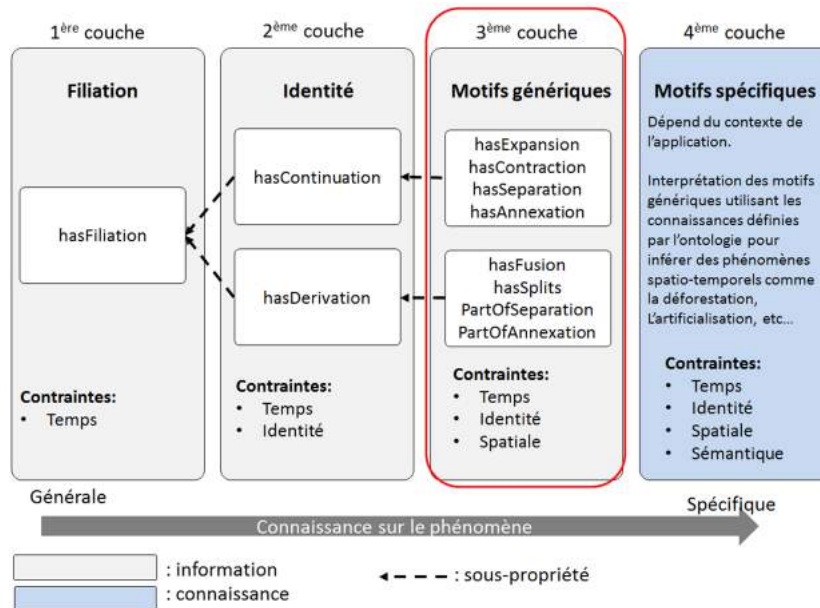


FIGURE 3.5 – Description de la couche des motifs génériques

Nous venons d'étudier les contraintes d'intégrités définies pour la deuxième couche de notre modèle, la couche identité. A présent, l'ajout des contraintes spatiales permet d'obtenir un troisième niveau d'expressivité. Les contraintes de chacune des relations du modèle sont définies ci-après et représentent des spécialisations des relations précédentes, le bloc SEM n'étant pas amené à varier au sein de cette nouvelle couche, nous ne le ferons plus apparaître pour des raisons pédagogiques.

hasGrowth : La relation "d'expansion" indique un agrandissement de la géométrie. En outre, cette relation est une spécialisation de la relation de "continuation". Par conséquent, l'identité n'est pas affectée par le changement.

$$\begin{aligned}
 \mathbf{REL}_{\text{Growth}} &: \forall p(\forall c(\text{hasGrowth}(p, c) \wedge \mathcal{T}S(p) \wedge \mathcal{T}S(c)) \\
 \mathbf{GEOM}_{\text{Growth}} &: \exists g_p \exists g_c(\text{hasGeometry}(p, g_p) \wedge \text{hasGeometry}(c, g_c) \rightarrow \text{Within}(g_c, g_p) \\
 \mathbf{ID}_{\text{Growth}} &= \mathbf{ID}_{\text{Continuation}} \\
 \mathbf{TEMP}_{\text{Growth}} &= \mathbf{TEMP}_{\text{Continuation}}
 \end{aligned} \tag{3.25}$$

$$\mathbf{REL}_{\text{Growth}} \rightarrow \mathbf{GEOM}_{\text{Growth}} \wedge \mathbf{ID}_{\text{Growth}} \wedge \mathbf{TEMP}_{\text{Growth}}$$

La relation d'expansion est une spécialisation de la relation de continuation. La spécificité de cette relation intervient sur le bloc $GEOM_{Growth}$ en définissant que la géométrie du timeslice parent g_p doit être contenue dans la géométrie du timeslice enfant g_c .

hasReduction : La relation de "réduction" indique une diminution de la géométrie. Cette relation est également une spécialisation de la relation de "continuation"

$$\begin{aligned} \mathbf{REL}_{Reduction} &: \forall p(\forall c(hasReduction(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c))) \\ \mathbf{GEOM}_{Reduction} &: \exists g_p \exists g_c(hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \rightarrow \textit{Within}(g_p, g_c)) \\ \mathbf{ID}_{Reduction} &= \mathbf{ID}_{Continuation} \\ \mathbf{TEMP}_{Reduction} &= \mathbf{TEMP}_{Continuation} \end{aligned}$$

$$\mathbf{REL}_{Reduction} \rightarrow \mathbf{GEOM}_{Reduction} \wedge \mathbf{ID}_{Reduction} \wedge \mathbf{TEMP}_{Reduction} \quad (3.26)$$

La relation de réduction est une spécialisation de la relation de continuation. La spécificité de cette relation intervient sur le bloc $GEOM$ en définissant que la géométrie du timeslice parent g_p doit contenir la géométrie du timeslice enfant g_c .

hasEquality : La relation "d'égalité" indique qu'aucun changement ne s'est produit sur une aucune composante.

$$\begin{aligned} \mathbf{REL}_{Equality} &: \forall p(\forall c(hasEquality(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c))) \\ \mathbf{GEOM}_{Equality} &: \exists g_p \exists g_c(hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \rightarrow \textit{hasEqualGeometry}(g_p, g_c)) \\ \mathbf{ID}_{Equality} &= \mathbf{ID}_{Continuation} \\ \mathbf{TEMP}_{Equality} &= \mathbf{TEMP}_{Continuation} \end{aligned}$$

$$\mathbf{REL}_{Equality} \rightarrow \mathbf{GEOM}_{Equality} \wedge \mathbf{ID}_{Equality} \wedge \mathbf{TEMP}_{Equality} \quad (3.27)$$

La relation d'égalité est une spécialisation de la relation de continuation. La spécificité de cette relation intervient sur le bloc $GEOM_{Equality}$ en définissant que la géométrie du timeslice parent g_p doit être égale à la géométrie du timeslice enfant g_c .

hasConversion : La relation de "conversion" indique qu'aucune modification ne s'est produite sur les composantes sémantiques et géométriques. En revanche, l'identité a été affectée par le changement.

$$\begin{aligned} \mathbf{REL}_{Conversion} &: \forall p(\forall c(hasConversion(p, c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c))) \\ \mathbf{GEOM}_{Conversion} &: \exists g_p \exists g_c(hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \rightarrow \textit{hasEqualGeometry}(g_p, g_c)) \\ \mathbf{ID}_{Conversion} &= \mathbf{ID}_{Derivation} \\ \mathbf{TEMP}_{Conversion} &= \mathbf{TEMP}_{Derivation} \end{aligned}$$

$$\mathbf{REL}_{Conversion} \rightarrow \mathbf{GEOM}_{Conversion} \wedge \mathbf{ID}_{Conversion} \wedge \mathbf{TEMP}_{Conversion} \quad (3.28)$$

Contrairement aux trois précédentes relations, la relation de conversion est une spécialisation de la relation de dérivation. Ainsi, par héritage, l'identité du timeslice parent est différente de celle du timeslice enfant. En outre, la relation de conversion impose que la géométrie du timeslice parent g_p soit égale à la géométrie du timeslice enfant g_c .

Les prochaines relations témoignent de motifs d'évolution plus complexes impliquant non plus un timeslice parent et un timeslice enfant mais plusieurs timeslices parents et un timeslice enfant ou encore un timeslice parent et plusieurs timeslices enfants.

hasSplits : La relation de "division" révèle un motif plus complexe d'évolution. Elle indique une division de l'entité parent en deux (ou plus) entités enfants. Par conséquent, chaque relation établie entre l'entité parent et les entités enfant sont des relations de "dérivation". D'autre part, l'union des géométries enfant doit être égal à la géométrie parent.

$$\begin{aligned}
\mathbf{REL}_{\text{Splits}} &: \forall p \forall c \forall c' (hasSplits(p, c) \wedge hasSplits(p, c') \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c) \wedge \mathcal{TS}(c')) \\
\mathbf{GEOM}_{\text{Splits}} &: \exists g_p \exists g_c \exists g_{c'} hasGeometry(p, g_p) \wedge hasGeometry(c, g_c) \wedge hasGeometry(c', g_{c'}) \\
&\rightarrow hasEqualGeometry(Union(g_c, g_{c'}), g_p) \\
\mathbf{ID}_{\text{Splits}} &: \exists o_p \exists o_c \exists o_{c'} isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \wedge isTimesliceOf(c', o_{c'}) \\
&\rightarrow (o_p \neq o_c) \wedge (o_p \neq o_{c'}) \\
\mathbf{TEMP}_{\text{Splits}} &: \exists i_p \exists i_c \exists i_{c'} hasTime(p, i_p) \wedge hasTime(c, i_c) \wedge hasTime(c', i_{c'}) \\
&\rightarrow Meets(i_p, i_c) \wedge Meets(i_p, i_{c'})
\end{aligned}$$

$$\mathbf{REL}_{\text{Splits}} \rightarrow \mathbf{GEOM}_{\text{Splits}} \wedge \mathbf{ID}_{\text{Splits}} \wedge \mathbf{TEMP}_{\text{Splits}} \quad (3.29)$$

La relation de division est une spécialisation de la relation de dérivation impliquant un timeslice parent et plusieurs timeslices enfants. Par héritage, le bloc ID indique que l'identité du timeslice parent doit être différente de celles des timeslices enfant. En revanche, il n'existe aucune contrainte d'identité entre les timeslices enfants. Le bloc $\mathbf{TEMP}_{\text{Splits}}$, quant à lui, indique que l'instant ou l'intervalle d'existence des timeslices enfants sont consécutifs à l'instant ou l'intervalle d'existence du timeslice parent. Enfin, la relation de division se distingue de la relation de dérivation par l'ajout d'une contrainte spatiale spécifiant que l'union des géométries des timeslices enfants g_c et $g_{c'}$ doit être contenue dans la géométrie g_p du timeslice parent.

hasSeparation : La relation de "séparation" révèle également un motif complexe d'évolution similaire à la relation "division" à la différence qu'une des entités enfants a conservé l'identité du parent.

$$\begin{aligned}
\mathbf{REL}_{\text{Separation}} &: \forall p \forall c \forall c' (hasSeparation(p, c) \wedge PartOfSeparation(p, c') \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(c) \wedge \mathcal{TS}(c')) \\
\mathbf{GEOM}_{\text{Separation}} &= \mathbf{GEOM}_{\text{Splits}} \\
\mathbf{ID}_{\text{Separation}} &: \exists o_p \exists o_c \exists o_{c'} isTimesliceOf(p, o_p) \wedge isTimesliceOf(c, o_c) \wedge isTimesliceOf(c', o_{c'}) \\
&\rightarrow (o_p = o_c) \wedge (o_p \neq o_{c'}) \\
\mathbf{TEMP}_{\text{Separation}} &= \mathbf{TEMP}_{\text{Splits}}
\end{aligned}$$

$$\mathbf{REL}_{\text{Separation}} \rightarrow \mathbf{GEOM}_{\text{Separation}} \wedge \mathbf{ID}_{\text{Separation}} \wedge \mathbf{TEMP}_{\text{Separation}} \quad (3.30)$$

Le motif de séparation se traduit par deux relations distinctes au sein de notre modèle. L'une héritant de la relation de continuation tandis que l'autre hérite de la relation de dérivation que nous nommons respectivement relation de séparation et relation de "partie de séparation". Ainsi, la seule contrainte spécifique de ce motif concerne le bloc ID qui définit qu'il doit exister au moins un cas où l'identité du timeslice parent (o_p) est identique à l'identité d'un timeslice enfant (o_c) et au moins un cas où l'identité du timeslice parent (o_p) est différente de l'identité d'un timeslice enfant ($o_{c'}$). D'un point de vue spatial, la

relation de séparation est identique à la relation de division ainsi le bloc GEOM ne varie pas.

hasFusion : La relation de "fusion" révèle un motif complexe d'évolution. Elle indique une fusion de deux ou plusieurs entités parents en une seule et même entité enfant. Par conséquent, chaque relation établie entre l'entité parent et les entités enfants sont des relations de "dérivation". D'autre part, l'union des géométries parents doit être égale à la géométrie enfant.

$$\begin{aligned}
\mathbf{REL}_{\text{Fusion}} &: \forall p \forall p' \forall c (hasFusion(p, c) \wedge hasFusion(p', c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(p') \wedge \mathcal{TS}(c)) \\
\mathbf{GEOM}_{\text{Fusion}} &: \exists g_p \exists g_{p'} \exists g_c hasGeometry(p, g_p) \wedge hasGeometry(p', g_{p'}) \wedge hasGeometry(c, g_c) \\
&\rightarrow \textit{Within}(\textit{Union}(g_p, g_{p'}), g_c) \\
\mathbf{ID}_{\text{Fusion}} &: \exists o_p \exists o_{p'} \exists o_c isTimesliceOf(p, o_p) \wedge isTimesliceOf(p', o_{p'}) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow (o_p \neq o_c) \wedge (o_{p'} \neq o_c) \\
\mathbf{TEMP}_{\text{Fusion}} &: \exists i_p \exists i_{p'} \exists i_c (hasTime(p, i_p) \wedge hasTime(p', i_{p'}) \wedge hasTime(c, i_c)) \\
&\rightarrow \textit{Meets}(i_p, i_c) \wedge \textit{Meets}(i_{p'}, i_c)
\end{aligned}$$

$$\mathbf{REL}_{\text{Fusion}} \rightarrow \mathbf{GEOM}_{\text{Fusion}} \wedge \mathbf{ID}_{\text{Fusion}} \wedge \mathbf{TEMP}_{\text{Fusion}} \tag{3.31}$$

À l'instar de la relation de division, la relation de fusion est une spécialisation de la relation de dérivation mais se distingue en impliquant plusieurs timeslices parents et un timeslice enfant. Par héritage, le bloc ID indique que l'identité des timeslices parents doit être différente de celle du timeslice enfant. En revanche, il n'existe aucune contrainte d'identité entre les timeslices parents. Le bloc TEMP, quant à lui, indique que l'instant ou l'intervalle d'existence du timeslice enfant est consécutif à l'instant ou l'intervalle d'existence des timeslices parents. Enfin, la relation de fusion se distingue de la relation de dérivation par l'ajout d'une contrainte spatiale spécifiant que l'union des géométries des timeslices parents g_p et $g_{p'}$ doit être contenue dans la géométrie g_c du timeslice enfant.

hasAnnexion : La relation d'annexion révèle également un motif complexe d'évolution similaire à la relation de fusion à la différence qu'une des entités parents conserve son identité lors du changement.

$$\begin{aligned}
\mathbf{REL}_{\text{Annexion}} &: \forall p \forall p' \forall c (hasAnnexion(p, c) \wedge PartOfAnnexion(p', c) \wedge \mathcal{TS}(p) \wedge \mathcal{TS}(p') \wedge \mathcal{TS}(c)) \\
\mathbf{GEOM}_{\text{Annexion}} &= \mathbf{GEOM}_{\text{Fusion}} \\
\mathbf{ID}_{\text{Annexion}} &: \exists o_p \exists o_{p'} \exists o_c isTimesliceOf(p, o_p) \wedge isTimesliceOf(p', o_{p'}) \wedge isTimesliceOf(c, o_c) \\
&\rightarrow (o_p = o_c) \wedge (o_{p'} \neq o_c) \\
\mathbf{TEMP}_{\text{Annexion}} &= \mathbf{TEMP}_{\text{Fusion}}
\end{aligned}$$

$$\mathbf{REL}_{\text{Annexion}} \rightarrow \mathbf{GEOM}_{\text{Annexion}} \wedge \mathbf{ID}_{\text{Annexion}} \wedge \mathbf{TEMP}_{\text{Annexion}} \tag{3.32}$$

Tout comme le motif de séparation, le motif d'annexion se traduit par deux relations distinctes au sein de notre modèle. L'une héritant de la relation de continuation tandis que l'autre hérite de la relation de dérivation que nous nommons respectivement relation d'annexion et relation de "partie d'annexion". Ainsi, la seule contrainte spécifique de ce motif concerne le bloc ID qui définit qu'il doit exister au moins un cas où l'identité du timeslice parent (o_p) est identique à l'identité d'un timeslice enfant (o_c) et au moins un cas où l'identité du timeslice parent (o_p) est différente de l'identité d'un timeslice enfant ($o_{c'}$).

D'un point de vue spatial, la relation d'annexion est identique à la relation de fusion ainsi le bloc GEOM ne varie pas.

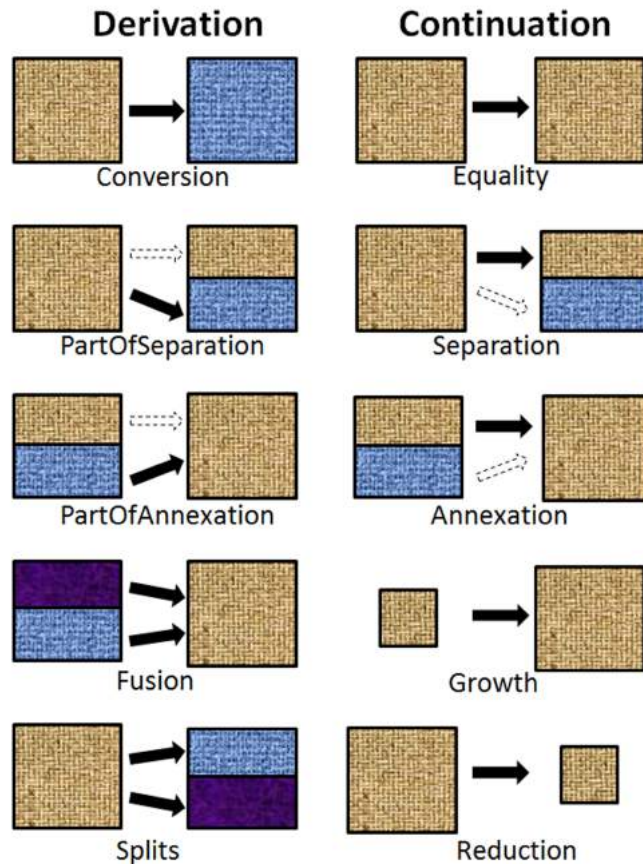


FIGURE 3.6 – Liste des motifs génériques détectés dans le modèle Continuum

Les contraintes d'intégrité que nous venons d'exprimer en logique du premier ordre pour décrire nos différentes couches ne peuvent être définies à l'aide d'axiome OWL à cause de la nécessité d'utiliser des opérateurs de calcul entre les géométries et les intervalles temporels. Toutefois, il est possible de vérifier nos relations en utilisant nos propres règles à l'aide du langage SPARQL Update. Nous aborderons ce point dans la suite de ce rapport. Actuellement, il n'existe évidemment aucun jeu de données formaté spécifiquement en fonction des besoins du modèle Continuum. Tout particulièrement, aucun jeu de données ne fournit les relations de filiation. En revanche, il est indispensable d'inférer ces relations à partir des jeux de données existants pour étudier le bon fonctionnement de la vérification des contraintes d'intégrité. Pour cela, le modèle Continuum nécessite la présence absolue des composantes spatiale, identitaire et temporelle du timeslice afin de servir de base aux futurs raisonnements nécessaires pour l'inférence des relations de filiation. Ces contraintes d'intégrité ne nécessitant pas d'opérateur, il est alors possible de les définir en utilisant des axiomes OWL qui seront interprétés sous l'hypothèse du monde fermé afin de permettre la vérification de la consistance de l'ontologie. La section suivante décrit la mise en oeuvre de ces contraintes au sein de notre modèle.

3.2.2/ CONTRAINTES D'INTÉGRITÉ SUR LES PRIMITIVES DU MODÈLE CONTINUUM

Les données spatio-temporelles sont, par leur nature évolutive, très sujettes à des mises à jour. En conséquence, l'intégration des données est une composante intrinsèque de la modélisation spatio-temporelle. La plupart des systèmes d'informations sont conçus en conformité avec le modèle de données afin de structurer, manipuler et rechercher des données. En outre, les modèles doivent fournir des langages ou règles implicites pour garantir l'intégrité des données. Par conséquent, l'intégration des données est fortement liée à la validation des données afin d'assurer la cohérence du système d'informations. Traditionnellement, les contraintes sont interprétées de différentes manières selon qu'elles respectent l'hypothèse du monde ouvert ou l'hypothèse du monde fermé. En effet, les contraintes sont interprétées en tant que contraintes d'intégrité dans les approches relationnelles et en tant que axiome logique dans l'approche Web Sémantique. Dans le premier cas, une erreur est soulevée lorsque les données insérées violent les contraintes définies tandis que dans le second cas, les contraintes servent de support pour les raisonnements d'inférence. Par exemple, considérons un système d'informations comportant les assertions suivantes :

Sujet	Prédicat	Objet
hasGrowth	range	Timeslice
Timeslice_01_1990	hasGrowth	Timeslice_01_2000

La première assertion définit qu'un triplet utilisant une relation d'expansion comme prédicat doit être une instance de la classe *Timeslice*. La seconde assertion est un exemple de triplet utilisant la relation d'expansion comme prédicat. Par conséquent, selon la première assertion, *Timeslice_01_2000* doit être une instance de la classe *Timeslice*. Mais ce fait n'est pas explicitement défini dans le système d'informations. Ainsi, la seconde assertion ne satisfait pas la première. Par conséquent, plusieurs interprétations sont possibles sur ce système d'informations. Sous l'hypothèse du monde ouvert, le système considérera que la base de données est consistante en inférant que *Timeslice_01_2000* est une instance de la classe *Timeslice*. Ainsi, une nouvelle assertion est automatiquement ajoutée dans le système d'information. En d'autres termes, de la connaissance est générée afin de satisfaire la contrainte et éviter l'inconsistance du système. A présent, considérons l'hypothèse du monde fermé. Le système ne dispose pas de fait explicite définissant *Timeslice_01_2000* comme une instance de la classe *Timeslice*. Par conséquent, une erreur sera levée en considérant que la base de données n'est pas consistante.

Dans ces travaux, l'approche est intégrative, car nous considérons que les deux approches possèdent leurs propres avantages. Par conséquent, nous proposons alors un système hybride pour lequel certaines assertions bien définies sont interprétées en monde fermé afin de garantir la consistance du système, tandis que les autres assertions seront laissées à l'interprétation sous l'hypothèse du monde ouvert afin de générer de la connaissance automatique à partir des données présentes dans le système. Notre modèle est conçu pour établir des raisonnements sur la relation de filiation. Comme étudié précédemment, les relations de filiation s'établissent entre des timeslices. Par conséquent, il est impératif que tous les timeslices soient correctement définis dans le respect du modèle Continuum. Dès lors, les contraintes devant être interprétées en monde fermé sont celles permettant de vérifier que chaque entité du modèle est définie

sous la forme d'un timeslice possédant obligatoirement une identité, une géométrie et une estampille temporelle. Peu d'outil de la littérature autorise la co-existence du monde ouvert et du monde fermé. A notre connaissance, le seul outil offrant cette fonctionnalité est le triple store *Stardog*. Pour cela, les contraintes devant être interprétées en monde fermé (les contraintes d'intégrité) doivent être définie dans un fichier spécifique. Ensuite, *Stardog* lance simplement une validation de données pour vérifier la consistance du système dans le respect des contraintes définies dans le fichier. Le contenu de ce fichier est donné dans le code 3.1 :

Listing 3.1 – fichier des contraintes d'intégrité utilisé pour la validation de données

```

1 @prefix owl: <http://www.w3.org/2002/07/owl#> .
2 @prefix : <http://www.w3.org/2012/12/rdf-val/SOTA-ex#> .
3 @prefix foaf: <http://xmlns.com/foaf/0.1/'> .
4 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
5 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
6 @prefix cs: <http://www.checksem.fr#> .
7
8
9 cs:Timeslice a owl:Class ;
10   rdfs:subClassOf
11     [ owl:onProperty cs:hasGeometry ; owl:cardinality 1 ] ,
12     [ owl:onProperty cs:hasTime ; owl:cardinality 1 ] .
13
14 cs:hasGeometry a owl:DatatypeProperty ; rdfs:domain cs:Timeslice;rdfs:range xsd:String .
15 cs:hasTime a owl:ObjectProperty ; rdfs:domain cs:Timeslice ; rdfs:range checksem:Time .
16 cs:hasTSArea a owl:DatatypeProperty ; rdfs:domain cs:Timeslice; rdfs:range xsd:double .
17
18
19
20
21 cs:SpatialTemporalRelation a owl:Class ;
22   rdfs:subClassOf
23     [ owl:onProperty cs:hasGeometry ; owl:cardinality 1 ] ,
24     [ owl:onProperty cs:hasTSInitial ; owl:cardinality 1 ] ,
25     [ owl:onProperty cs:hasTSResulting ; owl:cardinality 1 ] ,
26     [ owl:onProperty cs:hasArea ; owl:cardinality 1 ] ,
27     [ owl:onProperty cs:hasOverlappingChild ; owl:cardinality 1 ] ,
28     [ owl:onProperty cs:hasOverlappingParent ; owl:cardinality 1 ] .
29
30 cs:hasTSInitial a owl:ObjectProperty ; rdfs:domain cs:SpatialTemporalRelation ;
31   rdfs:range cs:TimeSlice .
32 cs:hasTSResulting a owl:ObjectProperty ; rdfs:domain cs:SpatialTemporalRelation ;
33   rdfs:range cs:TimeSlice .
34 cs:hasArea a owl:DatatypeProperty ; rdfs:domain cs:SpatialTemporalRelation;
35   rdfs:range xsd:double .
36 cs:hasOverlappingChild a owl:DatatypeProperty ; rdfs:domain cs:SpatialTemporalRelation;
37   rdfs:range xsd:integer .
38 cs:hasOverlappingParent a owl:DatatypeProperty ; rdfs:domain cs:SpatialTemporalRelation;
39   rdfs:range xsd:integer .

```

Dans le modèle Continuum, la primitive permettant de décrire nos entités est le timeslice. Chaque timeslice doit obligatoirement posséder une géométrie, une estampille temporelle et une identité. Le langage OWL permet de définir l'identité au travers d'une classe pouvant se substituer à la relation *isTimesliceof*. Ainsi l'identité est définie par une classe et cette classe est une sous-classe de la classe *Timeslice*. Par conséquent, la classe correspondant à l'identité hérite des contraintes définies sur la classe *Timeslice*. Les lignes 9 à 12 permettent de définir les contraintes d'intégrité des timeslices. Les relations du modèle doivent également être définies en contraintes d'intégrité. En effet, la propriété *datatype hasGeometry* ne peut s'appliquer que sur des timeslices, c'est pourquoi le *domain* de cette propriété doit être la classe *Timeslice*. Les coordonnées de géométrie de

notre modèle étant définies et stockées dans le format *WKT*, c'est-à-dire sous la forme d'une chaîne de caractère, le *range* de la propriété *hasGeometry* est une valeur de type *xsd:String*. De façon similaire, la propriété objet *hasTime* s'applique sur un timeslice et permet de définir un instant ou un intervalle de temps pour chacun d'eux. Ainsi, la classe *Timeslice* est définie comme *domain* et la classe *Time* comme *range* de cette propriété. A noter que contrairement à la propriété *hasGeometry*, le *range* est défini par une ressource et non une valeur. Cette ressource peut être une instance de la classe *Interval* ou de la classe *Instant* selon les besoins de l'application. Dans le premier cas, une valeur de type *xsd:dateTime* sera définie pour l'instant de départ et l'instant de fin de l'intervalle de temps, dans le second cas, une seule valeur de type *xsd:dateTime* sera définie. Les lignes 14 et 15 définissent les contraintes d'intégrité pour les propriétés *hasGeometry* et *hasTime* de notre modèle.

Les lignes suivantes du fichier concernent les travaux de pré-traitement nécessaires lorsque les relations de filiation ne sont pas directement présentes dans le jeu de données. Plus précisément, ces contraintes d'intégrité permettent de garantir que tous les éléments nécessaires aux calculs des relations de filiation sont présents dans le système d'information et pour toutes les parcelles du territoire.

Ce chapitre a proposé une description formelle du modèle Continuum en distinguant les couches spatio-temporelles des transitions spatio-temporelles. Les transitions ont été exprimé au travers de trois couches d'expressivité. Les chapitres suivants abordent la méthodologie de calcul de ces couches pour permettre de peupler le modèle Continuum à l'aide d'un jeu de données.

IMPLÉMENTATION ET CALCUL DE LA 1ÈRE ET 2ÈME COUCHE

Ce chapitre illustre un exemple d'implémentation de notre modèle à l'aide du jeu de données Corine Land Cover. Ce jeu de données définit trois couches temporelles pour tout le territoire européen. Chacune d'entre elles est composée d'un ensemble de parcelles évoluant entre chaque couche consécutive. Dans la section précédente, nous avons présenté notre modèle en supposant que les relations de filiation étaient déjà présentes dans le jeu de données. Cependant, à notre connaissance, aucun jeu de données n'est fourni avec des relations de filiation. Corine Land Cover ne fait pas exception, c'est pourquoi il est nécessaire de mettre en place une nouvelle méthodologie et quelques ajustements afin d'identifier les relations de filiation entre les parcelles d'un jeu de données spatio-temporelles. Dans un premier temps, nous introduirons les bases de notre implémentation avant de présenter la méthode utilisée pour inférer les relations de filiation.

Sommaire

4.1	Le jeu de données Corine Land Cover	93
4.2	Calcul de la couche 1 et 2	96
4.2.1	La transition spatio-temporelle comme base d'étude des relations de filiation	96
4.2.2	Calcul de la couche 1 : généalogie	98
4.2.3	Élimination des changements négligeables	98
4.2.4	Calcul de la couche 2 : cycle de vie	101

4.1/ LE JEU DE DONNÉES CORINE LAND COVER

Le jeu de données Corine Land Cover de [Bossard et al., 2000] est un inventaire biophysique de l'occupation des terres couvrant plusieurs pays du territoire européen. Chaque couche représente les années 1990, 2000 et 2006. Le jeu de données définit une hiérarchie de classes composées de trois niveaux pour classer chaque entité en fonction de leur couverture terrestre. Le premier niveau compte cinq catégories tandis que le second et le troisième compte respectivement quinze et quarante quatre catégories permettant de classer les entités. Le tableau 4.1 résume l'ensemble de la hiérarchie de classe.

TABLE 4.1 – La hiérarchie de classe de Corine Land Cover

Level 1	Level2	Level 3
Artificial surfaces	Urban fabric	Continuous urban fabric Discontinuous urban fabric
	Industrial, commercial and transport units	Industrial or commercial units Road and rail networks and associated land Port areas Airports
	Mine, dump and construction sites	Mineral extraction sites Dump sites Construction sites
	Artificial, non-agricultural vegetated areas	Green urban areas Sport and leisure facilities
Agricultural areas	Arable land	Non-irrigated arable land Permanently irrigated land Rice fields
	Permanent crops	Vineyards Fruit trees and berry plantations Olive groves
	Pastures	Pastures
	Heterogeneous agricultural areas	Annual crops associated with permanent crops Complex cultivation patterns Land principally occupied by agriculture, with significant areas of natural vegetation Agro-forestry areas
Forest and semi natural areas	Forests	Broad-leaved forest Coniferous forest Mixed forest
	Scrub and/or herbaceous vegetation associations	Natural grasslands Moors and heathland Sclerophyllous vegetation Transitional woodland-shrub
	Open spaces with little or no vegetation	Beaches, dunes, sands Bare rocks Sparsely vegetated areas Burnt areas Glaciers and perpetual snow
Wetlands	Inland wetlands	Inland marshes Peat bogs
	Maritime wetlands	Salt marshes Salines Intertidal flats
Water bodies	Inland waters	Water courses Water bodies
	Marine waters	Coastal lagoons Estuaries Sea and ocean

Dans notre modèle chaque parcelle de terre est convertie en un timeslice possédant une géométrie, une estampille temporelle (1990, 2000 ou 2006) et une identité déterminée par

une des classes de la hiérarchie Corine Land Cover. La figure 4.1 présente un exemple d'évolution du territoire sur trois couches temporelles.

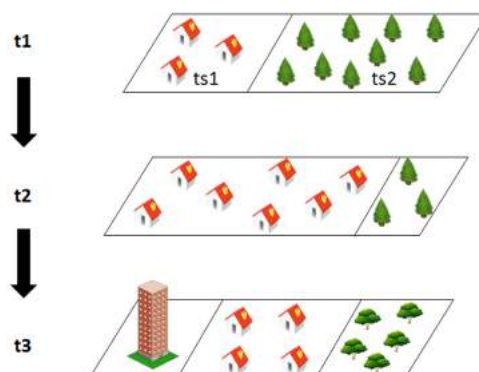


FIGURE 4.1 – Exemple d'évolution d'un territoire modélisé à l'aide de timeslices

A l'instant t_1 , le territoire est composé d'une parcelle de *tissu urbain continu* et d'une parcelle de *forêt de Conifère*. A l'instant t_2 , une bonne partie de la forêt s'est transformée en *tissu urbain continu*. Par conséquent, la parcelle de *tissu urbain continu* s'est étendue tandis que la *forêt de Conifère* s'est réduite. A l'instant t_3 , le *tissu urbain continu* se divise en deux parties. L'une d'entre elle reste inchangée tandis que l'autre se transforme en *zones industrielles et commerciales*. Enfin, la *forêt de Conifère* se convertit en *forêts mélangées*. L'étude de l'évolution d'un territoire tel que représenté par cet exemple nécessite une méthodologie afin de manipuler des entités spatio-temporelles.

De plus, Corine Land Cover fournit des données encodées sous forme de shapefiles [ESRI, 1998]. Afin d'intégrer ces données dans notre système, ceux-ci ont été convertis en triplets RDF en utilisant un programme JAVA que nous avons réalisé en utilisant la bibliothèque GeoTools¹. Les informations sous forme de triplets ont ensuite été chargées dans le triplestore Stardog² [Parsia, 2014]. Ce triplestore ne propose pas de support pour GeoSPARQL³ contrairement à certains autres. Toutefois, la principale raison pour laquelle nous avons opté pour ce triplestore concerne ses capacités de raisonnement. Plus précisément, Stardog supporte des raisonnements en OWL 2⁴ et SWRL⁵ et offre des options uniques pour traiter des contraintes sous l'hypothèse du monde fermé ou du monde ouvert. En somme, stardog est à notre connaissance, le seul système hybride permettant d'effectuer aussi bien des raisonnements d'inférence que de la vérification de contraintes d'intégrité en OWL.

Stardog ne proposant pas de support pour GeoSPARQL, l'analyse spatiale doit être faite avec des outils externes au triplestore. Dans notre cas, nous avons développé un outil basé sur JAVA/Geotools pour effectuer toutes les analyses spatiales requises. Dans nos travaux, les polygones de chaque strates temporelles ont été identifiés et implémentés sous la forme de timeslices (les primitives du modèle Continuum). Dans un second temps,

1. <http://www.geotools.org/>

2. <http://stardog.com/>

3. <http://www.opengeospatial.org/standards/geosparql>

4. <http://www.w3.org/TR/owl-overview/>

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

notre programme nous a permis d'interroger et récupérer tous ces timeslices. Pour notre exemple, nous nous sommes limités à l'étude du territoire de la Gironde. Pour cela, un index spatial a été utilisé afin de sélectionner uniquement les timeslices de ce département français. Enfin, la dernière étape de notre pré-traitement du jeu de données a consisté à identifier les relations de filiation, en prenant en considération les timeslices dont les géométries se chevauchaient à des instants consécutifs. En outre, notre application identifie également les relations d'adjacence entre des timeslices d'une même couche temporelle. Une fois les relations de filiation et d'adjacence identifiées, celles-ci ont également été converties en triplet RDF et chargées dans le triplestore.

4.2/ CALCUL DE LA COUCHE 1 ET 2

Une fois les données de base intégrées dans notre modèle, il est alors possible de s'intéresser aux relations de filiation. Les relations de filiation mettent en corrélation des entités présentes dans le jeu de données à différents instants de temps. Une relation de filiation est le témoin d'un changement affectant la composante sémantique ou la composante spatiale d'un timeslice. Les changements spatiaux incluent des divisions, des fusions, des expansions, des réductions ou tout simplement une modification de la forme des entités. En conséquence, établir un lien de filiation entre deux entités à différents instants peu s'avérer être une tâche difficile. Certains changements spatiaux sont très petits et insignifiants. Dans ce cas, ils ne doivent pas être pris en compte afin de conserver la pertinence des informations stockées ou inférées par le système. Le résultat est donc dépendant du choix d'un seuil qui impactera la filiation. Dans la suite, un postulat est émis pour fournir des données et visualiser les résultats. Toutefois, il serait plus pertinent que la filiation soit capturée par l'expert, ce qui n'est pas le cas pour l'exemple explicatif de Corine Land Cover.

4.2.1/ LA TRANSITION SPATIO-TEMPORELLE COMME BASE D'ÉTUDE DES RELATIONS DE FILIATION

Le jeu de données Corine Land Cover offre des données spatiales pour les années 1990, 2000 et 2006. Toutes les données nécessaires pour la représentation sous forme de timeslice sont présentes au sein du jeu de données. En revanche, les relations de filiation, comme pour la grande majorité des jeux de données disponibles, ne sont pas fournies et doivent être calculées afin de représenter et comprendre l'évolution du territoire. Sans aucune connaissance préalable sur le jeu de données, la première étape pour permettre d'identifier les relations de filiation dans le respect d'une certaine réalité, s'appuie sur l'idée naïve que si les géométries de deux timeslices consécutifs dans le temps ont une intersection alors ces timeslices possèdent un lien spatio-temporel. Cette idée est naïve car elle ne prend pas en compte les éventuelles micro-évolutions non-capturées entre deux instants de temps. Bien que la précision et l'incertitude soient des éléments pertinents à prendre en compte, nous postulons que les liens de filiation sont valides car elles sont fournies par l'expert. Ces points ne sont pas étudiés dans le cas de Corine Land Cover. Pour réaliser cette étape, nous avons établis des comparaisons entre toutes les parcelles du territoire entre 1990 et 2000, et entre 2000 et 2006. Le résultat de ce calcul d'intersection est alors stocké dans le système en tant qu'instance de la classe appelée *SpatialTemporalTransition*. La formule 4.1 illustre la spécification formelle de cette classe.

$$\begin{aligned}
 \text{SpatialTemporalTransition} \equiv & \\
 & \exists \text{hasParentTimeSlice} . \mathcal{T} S \sqcap \\
 & \exists \text{hasChildTimeSlice} . \mathcal{T} S \sqcap \\
 & \exists \text{hasOverlappingParent} . \text{Double} \sqcap \\
 & \exists \text{hasOverlappingChild} . \text{Double} \sqcap \\
 & \exists \text{hasArea} . \text{Double} \sqcap \\
 & \exists \text{hasGeometry} . \mathcal{G}
 \end{aligned}
 \tag{4.1}$$

La figure 4.2 illustre un exemple d'évolution modélisé à l'aide des transitions spatio-temporelles.

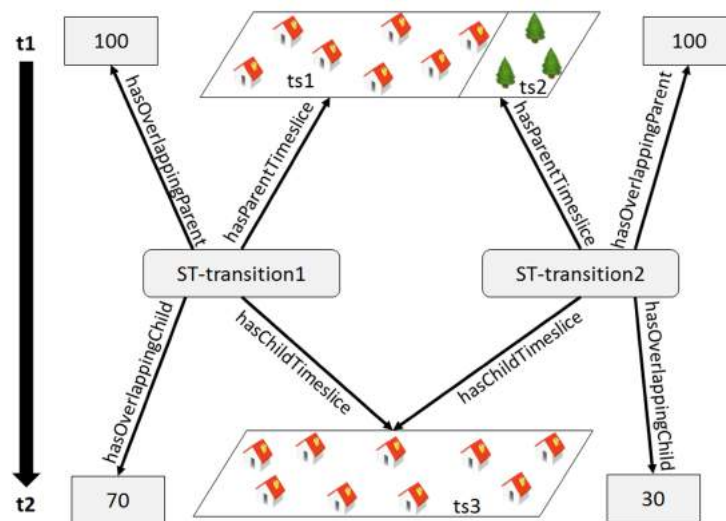


FIGURE 4.2 – Exemple d'évolution d'un territoire modélisé à l'aide de transitions

Chaque instance d'une transition spatio-temporelle est connectée à un timeslice parent et un timeslice enfant respectivement au travers des relations *hasParentTimeSlice* et *hasChildTimeSlice*. En outre, la surface d'intersection est stockée sous la forme d'une géométrie au format WKT et accessible via la propriété *hasGeometry*. La valeur en m^2 de cette surface est également disponible via la propriété *hasArea*. Sur la figure 4.2, ces deux propriétés ne sont pas représentées dans un souci de lisibilité. Enfin, en comparant la surface d'intersection respectivement avec la surface du timeslice parent et avec la surface du timeslice enfant, nous obtenons un pourcentage compris entre 0 et 100 représentant le taux d'intersection. Ce taux d'intersection est accessible via les propriétés *hasOverlappingParent* et *hasOverlappingChild*. Plus d'explications à propos du calcul et de l'utilité de ces valeurs seront données dans la suite de cette thèse. En définitive, le résultat de cette analyse des transitions spatio-temporelles fournit un ensemble de paires de timeslice possédant une intersection spatio-temporelle. Certaines de ces transitions représentent des relations de filiation tandis que d'autres doivent être considérées comme une erreur (ou un bruit) dû à des imprécisions du jeu de données ou des changements négligeables. Dès lors, une méthodologie doit être établie afin de distinguer les relations devant être stockées de celles ne devant pas être retenues.

4.2.2/ CALCUL DE LA COUCHE 1 : GÉNÉALOGIE

L'idée la plus simple pour retrouver les relations de filiation au sein d'un jeu de données spatio-temporel est de considérer qu'une entité fille ayant une intersection spatiale avec une entité parent possède un lien spatio-temporel avec celle-ci. La figure 4.3 illustre le graph spatio-temporel résultant de cette approche. La figure 4.3 représente la même évolution de territoire que la figure 4.1. Cependant, cette fois, chaque entité est associée à un timeslice ($ts1, ts2, \dots, ts7$). Le territoire est alors représenté selon trois couche temporelle $t1$, $t2$ et $t3$, chacune composée d'un ensemble de timeslices. Dès lors, il est possible de comparer tous les timeslices de deux couches temporelles consécutives. Dans notre exemple, nous comparons les timeslices de la couche temporelle $t1$ ($ts1, ts2$) avec les timeslices de la couche temporelle $t2$ ($ts3, ts4$) pour détecter l'évolution. Puis, nous comparons les timeslices de la couche temporelle $t2$ avec les timeslices de la couche temporelle $t3$ ($ts5, ts6, ts7$).

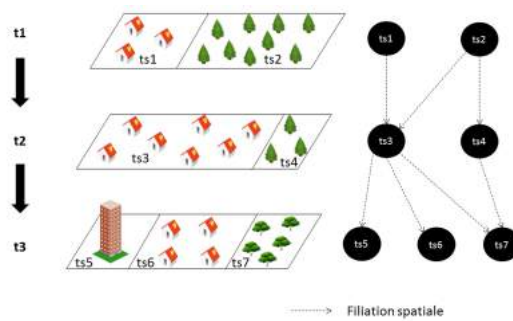


FIGURE 4.3 – Calcul spatial des relations de filiation

Dans cet exemple, la plupart des relations de filiation semblent correctes. Toutefois, il reste beaucoup de cas pour lesquels cette approche naïve n'est pas satisfaisante. La faiblesse de cette approche est mise en avant entre la couche $t2$ et la couche $t3$ comme le montre la figure 4.4. En effet, le système récupère une relation de filiation entre $ts4$ et $ts7$. Cependant, nous pouvons constater une expansion très minime entre $ts4$ et $ts7$ induisant une légère intersection spatiale entre $ts3$ et $ts7$ et donc une relation de filiation entre ces deux timeslices. En conséquence, le système considérera que $ts4$ s'est agrandi tandis que $ts3$ s'est divisé pour donner naissance à $ts5$, $ts6$ et également $ts7$. Bien que ces conclusions soient en accord avec la méthodologie utilisée, elles ne reflètent pas la réalité de la situation, car l'expansion détectée est trop infime pour être considérée comme significative. D'autre part, aucune évolution spatiale ne doit être prise en compte entre $ts4$ et $ts7$ si l'on se réfère à la réalité. Par conséquent, la relation de filiation détectée entre $ts3$ et $ts7$ est fautive.

4.2.3/ ÉLIMINATION DES CHANGEMENTS NÉGLIGEABLES

L'exemple illustré par la figure 4.4 est l'un des nombreux exemples reflétant le besoin de flexibilité du système lors du calcul des relations de filiation. Dès lors, il existe un besoin pour trouver des méthodes permettant à l'utilisateur de définir un seuil dans l'optique de filtrer les relations de filiations spatiales. Une filiation spatiale est une relation de filiation détectée par la méthode des transitions spatio-temporelles. Elle peut être valide ou non-

valide. Cette sous-section propose une méthode pour éliminer les filiations spatiales non-valides.

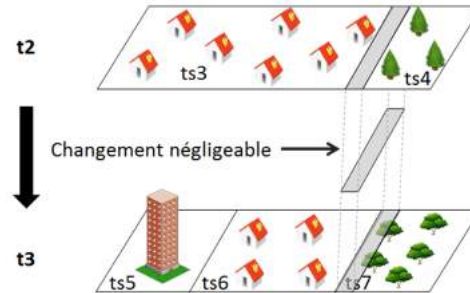


FIGURE 4.4 – Exemple de changement négligeable

Dans l'optique d'améliorer la fiabilité du processus de récupération des liens de filiation, nous considérons que plus le pourcentage d'intersection spatiale est élevé et plus il est probable que les entités consécutives soient liées dans le temps.

Dans l'équation 4.1, nous avons défini le concept de *SpatialTemporalTransition*. Les valeurs des propriétés *hasOverlappingParent* et *hasOverlappingChild* nous donne des informations à propos du type de relation spatiale qui existe entre un timeslice *parent* et un timeslice *enfant*. Le *domain* de ces deux propriétés est une valeur de type *Double* comprise entre 0 et 100. Pour calculer la valeur de ces propriétés, nous effectuons l'intersection des géométries des timeslices donnant lieu à une nouvelle géométrie comme le montre l'équation 4.2.

$$timeSliceParent_{area} \cap timeSliceChild_{area} = Intersection_{area} \quad (4.2)$$

Ensuite, nous comparons l'aire de cette géométrie résultante avec l'aire des timeslices *parent* et *enfant*. L'équation 4.3 montre la procédure pour calculer la valeur des propriétés *hasOverlappingChild*, tandis que l'équation 4.4 montre le calcul de la valeur de la propriété *hasOverlappingParent*.

$$\left(\frac{Intersection_{area}}{timeSliceChild_{area}} \right) * 100 = oc \quad (4.3)$$

$$\left(\frac{Intersection_{area}}{timeSliceParent_{area}} \right) * 100 = op \quad (4.4)$$

La valeur des propriétés *hasOverlappingParent* et *hasOverlappingChild* nous permet de quantifier quelle proportion la géométrie *parent* représente pour la géométrie *enfant* et vice-versa. Lors d'une évolution, un timeslice *enfant* peut avoir plusieurs *parents*. Inversement, un timeslice *parent* peut générer plusieurs timeslices *enfant*. Cependant, certains timeslices peuvent avoir une légère intersection qui ne révèle pas de réelle relation de filiation. Dans ces travaux, nous proposons d'utiliser les propriétés *hasOverlappingParent* et *hasOverlappingChild* pour déterminer lorsque deux timeslices consécutifs doivent effectivement être associés par une relation de filiation. Dès lors, nous décidons de valider

les relations de filiation seulement lorsque l'intersection du timeslice *parent* et du timeslice *enfant* dépasse un certain seuil. Nous définissons deux valeurs de seuil ρ et χ pour lesquelles respectivement les valeurs d'intersection *parent* et *enfant* op et oc doivent être supérieures pour valider la relation de filiation. La figure 4.5 illustre des exemples de filiations acceptées ou rejetées avec des valeurs ρ et χ définies à 5.

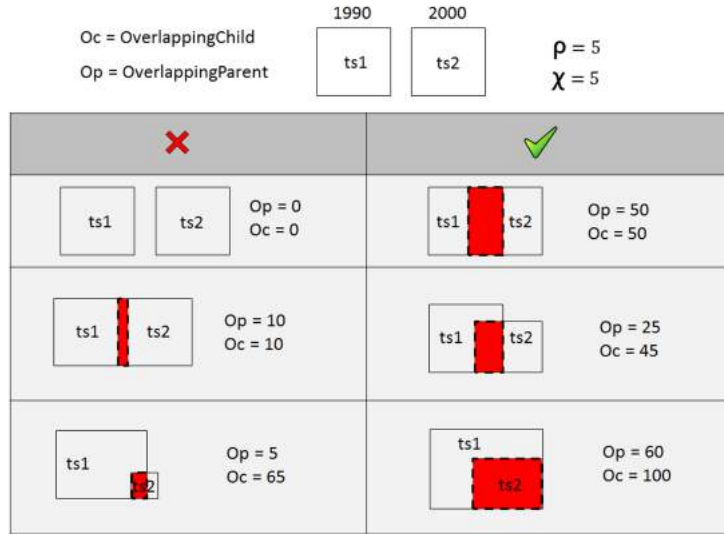


FIGURE 4.5 – Exemples de filiations acceptées et rejetées avec ρ et χ définis à 5

L'équation 4.5 spécifie les contraintes d'une relation de filiation valide :

$$\begin{aligned}
 & [\forall p \forall c (\mathcal{TS}(p) \wedge \mathcal{TS}(c) \wedge hasFiliation(p, c) \rightarrow \\
 & \quad (\exists st (SpatialTemporalTransition(st) \wedge \\
 & \quad \quad hasParentTimeSlice(st, p) \wedge \\
 & \quad \quad hasChildTimeSlice(st, c) \wedge \\
 & \quad \quad hasOverlappingParent(p, op) \wedge \\
 & \quad \quad hasOverlappingChild(c, oc) \wedge \\
 & \quad \quad (op > \rho_{Filiation}) \wedge \\
 & \quad \quad (oc > \chi_{Filiation})))))]
 \end{aligned} \tag{4.5}$$

avec $op, oc \in \mathbb{Q}$

Dans le cadre de notre exemple, nous avons arbitrairement défini un seuil à 5 pour les valeurs ρ et χ . Le tableau 4.2 montre le résultat du filtrage.

Finalement, une seule filiation spatiale n'a pas été retenue lors du seuillage du graphe spatial comme le montre la figure 4.6.

Nous venons de détailler une méthode de calcul de relation de filiation au sein d'un jeu de données spatio-temporelles. La méthode propose un filtrage des relations de filiation spatiale afin de garantir un résultat plus proche de la réalité. Les relations de filiation de la couche 1 permettent de retracer la généalogie d'entités spatio-temporelles mais n'offre pas une compréhension approfondie de l'évolution de celles-ci. La prochaine sous-

TABLE 4.2 – Relation de filiations valides entre les timeslices

Time	Parent	Child	Overlapping Parent	Overlapping Child	Filiation
t1 to t2	ts1	ts3	100	45	VRAI
	ts2	ts3	70	55	VRAI
	ts2	ts4	30	100	VRAI
t2 to t3	ts3	ts5	40	100	VRAI
	ts3	ts6	60	100	VRAI
	ts3	ts7	2	4	FAUX
	ts4	ts7	100	96	VRAI

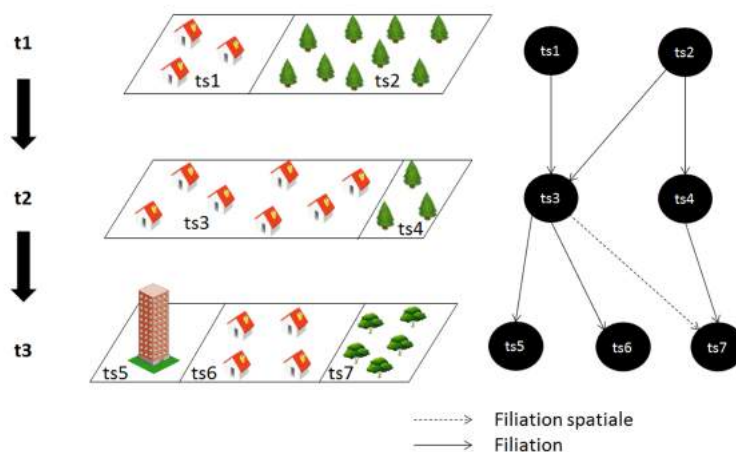


FIGURE 4.6 – Graphe de filiation filtré dans le respect de la réalité observable

section spécialise la relation de filiation pour décrire le cycle de vie des entités au travers de la couche 2.

4.2.4/ CALCUL DE LA COUCHE 2 : CYCLE DE VIE

La couche 2 se fonde sur la notion d'identité pour favoriser l'étude du cycle de vie au travers des relations de filiation. Dès lors, la relation de filiation est spécialisée par deux nouvelles relations : les continuations et les dérivations. La naissance et la mort d'une entité peuvent être étudiées grâce à ce processus. La dérivation, quant à elle, permet d'analyser le cycle de vie dérivé d'une entité, c'est-à-dire d'étudier les éventuelles mutations de l'entité au cours du temps.

Pour présenter notre implémentation, dans la suite de cette section, nous montrerons pour chaque relation héritée de la relation de filiation la règle en logique du premier ordre ainsi que la requête équivalente utilisée dans le système pour détecter nos relations. Pour cela, nous nous sommes basés sur le langage SPARQL 1.1 implémenté dans le triple store Stardog. Plus spécifiquement, nous avons utilisé SPARQL 1.1 Update pour ses aptitudes à fournir des mécanismes de mise à jour, d'insertion et de suppression de triplet RDF dans le support de graphe. Les formules 4.6 et 4.7 illustre les règles utilisées pour déterminer si une relation de filiation correspond à une *continuation* ou une *dérivation*.

$$\begin{aligned}
 & [\forall p \forall c (hasFiliation(p, c) \wedge isTimesliceOf(p, classP) \\
 & \quad \wedge isTimesliceOf(c, classC) \wedge (classP = classC) \\
 & \quad \rightarrow hasContinuation(p, c))]
 \end{aligned}
 \tag{4.6}$$

Listing 4.1 – Requête SPARQL pour détecter les relations de continuation

```

insert
{
  ?p checksem:hasContinuation ?c
}
where
{
  ?st a checksem:SpatiotemporalTransition .
  ?st checksem:hasParentTimeSlice ?p.
  ?st checksem:hasChildTimeSlice ?c.
  ?p a ?classP .
  ?c a ?classC .
  ?p checksem:hasOverlappingParent ?rho .
  ?c checksem:hasOverlappingChild ?xhi .
  FILTER(?classP=?classC)
}

```

$$\begin{aligned}
 & [\forall p \forall c (hasFiliation(p, c) \wedge isTimesliceOf(p, classP) \\
 & \quad \wedge isTimesliceOf(c, classC) \wedge (classP \neq classC) \\
 & \quad \rightarrow hasDerivation(p, c))]
 \end{aligned}
 \tag{4.7}$$

Listing 4.2 – Requête SPARQL Query pour détecter les relations de dérivation

```

insert
{
  ?p checksem:hasDerivation ?c
}
where
{
  ?st a checksem:SpatiotemporalTransition .
  ?st checksem:hasParentTimeSlice ?p.
  ?st checksem:hasChildTimeSlice ?c.
  ?p a ?classP .
  ?c a ?classC .
  ?f checksem:hasOverlappingParent ?rho .
  ?f checksem:hasOverlappingChild ?xhi .
  FILTER(?landcoverP != ?landcoverC)
}

```

En utilisant les valeurs de *hasOverlappingParent* et *hasOverlappingChild* avec les règles définies dans les équations 4.6 et 4.7, nous obtenons le graphe illustré par la figure 4.7 composé uniquement de *continuations* et de *dérivations*.

Ce chapitre a présenté une méthodologie pour calculer des relations de filiation. La méthode proposée se fonde sur un graphe de relations spatiales pour lequel un filtrage est nécessaire afin de garantir des résultats en accord avec la réalité. Le graphe de filiation valide décrit la couche 1 de notre modèle et permet de mettre en avant la généalogie d'un territoire. Sur les bases de la relation de filiation, il est possible d'augmenter l'expressivité avec des relations spécialisées, que sont les continuations et dérivations, calculées

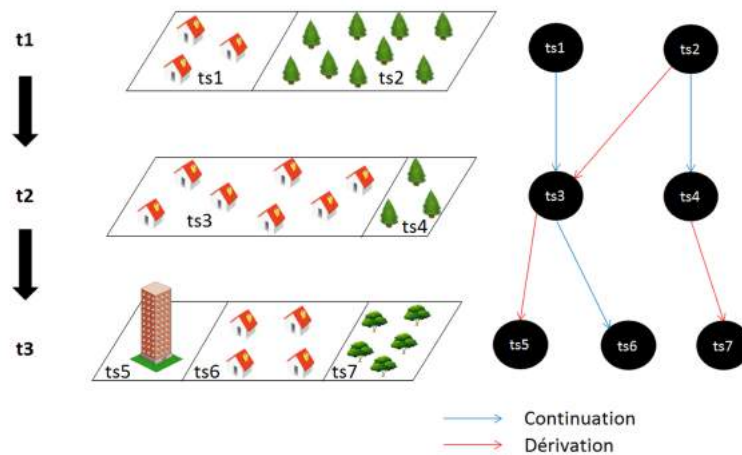


FIGURE 4.7 – Les relations de continuation et dérivation

à l'aide de la composante d'identité. Les continuations permettent d'étudier le cycle de vie primaire d'une entité, c'est-à-dire le processus généalogique au cours duquel l'identité de l'entité n'a pas varié au cours du temps. Les dérivations, quant-à-elles, sont généralement le témoin de causalités entre l'entité parent et l'entité enfant. Par exemple, si une zone urbaine à un temps $T1$ est reliée par une relation de dérivation à une entité assimilée à une étendue d'eau à un temps $T2$ alors il est fort probable qu'il y ait eu un phénomène d'inondation. Toutefois, cette conclusion ne peut l'être que grâce à l'assistance d'un expert. Par conséquent, il est nécessaire d'accroître davantage l'expressivité de la relation de filiation afin de permettre à la machine de qualifier de façon plus précise les évolutions des entités au sein du jeu de données. Le chapitre suivant décrit la 3ème couche du modèle Continuum en proposant de représenter l'évolution des entités au travers de motifs spatiaux.

CALCUL DE LA COUCHE 3 : LES MOTIFS GÉNÉRIQUES SPATIAUX

Ce chapitre se fonde sur les relations de filiation valides pour offrir des analyses plus complexes de l'évolution du territoire. Il existe deux approches complémentaires permettant d'étudier l'évolution : quantitative et qualitative. Une analyse quantitative révèle combien la surface d'un territoire terrestre s'est transformée entre deux instants, tandis que l'analyse qualitative tente de comprendre comment s'est produit le changement. L'approche quantitative, qui est la méthode classique d'étude des changements intervenant sur un territoire, a été abordée dans la section 2.6 de l'état de l'art. Ce chapitre abordera l'approche qualitative. Cette seconde méthode se fonde sur l'étude des relations qualitatives stockées dans le modèle. Cette méthode permet d'inférer des phénomènes au sein du jeu de données, en se basant sur les motifs génériques définis dans la troisième couche d'expressivité des relations de filiation de notre modèle.

Sommaire

5.1 Relations de continuation unaire	106
5.2 Relations de dérivation unaire	108
5.3 Motifs d'évolution N-aire	109

L'analyse quantitative est une approche globale pour détecter des changements sur la couverture terrestre, mais ne constitue pas un support satisfaisant pour pleinement comprendre les évolutions du territoire. Pour dépasser cette limitation, nous travaillons à un niveau plus détaillé en utilisant les relations de filiation existantes entre les parcelles de chaque strate temporelle. Les relations de filiation peuvent être des continuations ou des dérivations. Ensuite, continuations et dérivations peuvent intervenir dans le cadre de l'évolution d'une entité unique (relation 1-1) ou dans des configurations spatiales impliquant plusieurs entités (relations 0-N ou N-0) tels que des motifs basés sur des fusions ou des divisions. Dans le cas d'une relation unaire, la relation de continuation se décline en trois relations que sont l'expansion, la réduction et l'égalité. En parallèle, une relation dérivation peut être spécialisée en une relation de conversion. Dans ce travail, nous parlerons de continuations unaires et de dérivations unaires. Dans le cas de relations N-aires, quatre motifs d'évolution différents se distinguent : division, fusion, annexion, séparation. Qu'il s'agisse de motifs unaires ou N-aires, chacune des relations représentant ces motifs est une spécialisation de la couche 2 du modèle Continuum, c'est-à-dire une spécialisation de la relation de continuation ou de dérivation. Ce chapitre présente le calcul des motifs spatiaux de la couche 3.

5.1/ RELATIONS DE CONTINUATION UNAIRE

Les continuations sont ces relations pour lesquelles l'identité reste inchangée entre deux points de temps consécutifs. Nous distinguons trois types différents de continuation faisant référence à des phénomènes de croissance, de réduction ou de maintien de l'aire d'une parcelle. Ces motifs sont détectés au sein du jeu de données pour servir de base aux traitements, dans l'optique d'améliorer les connaissances liées à l'évolution du territoire représenté. De prime abord, détecter ces continuations apparaît comme un processus trivial. En considérant deux timeslices $ts1$ et $ts2$ possédant une intersection spatiale et existant respectivement à deux points consécutifs de temps, alors en se basant sur l'analyse spatiale de ces entités, il suffit de déduire nos trois relations de continuation de la manière suivante :

- Expansion : $ts1$ est contenu dans $ts2$
- Réduction : $ts1$ contient $ts2$
- Égalité : $ts1$ est égale à $ts2$

Deux problèmes se posent lors de la mise en application de cette analyse. Le premier concerne le temps de calcul nécessaire pour fournir l'analyse spatiale pour chaque relation de filiation de l'ensemble du jeu de données. Le deuxième problème concerne, une fois de plus les changements négligeables, qui doivent être éliminés du résultat de l'analyse.

Par exemple, une très légère croissance ou réduction ne devrait pas être interprétée comme telle, de même une légère modification des frontières entre deux timeslices consécutifs ne devrait pas être interprétée comme un changement, mais plutôt comme une relation d'égalité. Une fois de plus, un seuil est nécessaire pour faire face à ces imprécisions.

Dans cette approche, nous avons décidé d'utiliser, une fois de plus, les valeurs d'intersection parent et enfant. Le premier problème, soulevé précédemment, implique de trouver un moyen pour déterminer le résultat des fonctions spatiales comme *contient*, *est contenu dans* et *est égal à*. Les opérations spatiales traditionnelles sont en mesure de résoudre ce problème, mais nécessite des temps de calcul très longs alors que les valeurs d'intersection permettent un calcul bien plus rapide pour ces opérations. Une valeur d'intersection représente une disjonction tandis qu'une valeur d'intersection de 100 conduit à trois options distinctes :

- Le timeslice fils est contenu dans le timeslice parent
- Le timeslice fils contient le timeslice parent
- Le timeslice fils est égal au timeslice parent

Pour distinguer chaque cas, il est nécessaire de comparer les valeurs d'intersection enfant et parent. Ainsi, si la valeur d'intersection parent est de 100 et la valeur d'intersection enfant est inférieure à 100, alors nous pouvons déduire une croissance entre deux timeslices liés par une relation de continuation.

De même, si la valeur d'intersection enfant est de 100 et que la valeur d'intersection parent est inférieure à 100, alors nous pouvons en déduire une réduction. Enfin, si les valeurs d'intersection du parent et de l'enfant sont de 100, alors nous pouvons en déduire une égalité.

De façon théorique, cette méthode est valable. Cependant dans la pratique, cette méthode ne prend pas en considération les changements négligeables et peut conduire à de mauvaises interprétations dans le cadre des relations définies, creusant ainsi l'écart entre le monde réel tel qu'il existe et sa représentation au sein du système d'information géographique.

Par exemple, une continuation avec une valeur d'intersection parent définie à 100 et une valeur d'intersection enfant définie à 97, sera interprétée comme une expansion là où le sens commun suggère davantage une relation égalité. Pour répondre à cette problématique, nous définissons un seuil définissant un taux minimum de croissance ou de réduction et un écart acceptable entre les deux valeurs en cas d'égalité.

Afin de définir un taux minimum de croissance ou de réduction, la différence entre les valeurs d'intersection parent et enfant est calculée puis les relations pour lesquelles les résultats obtenus dépassent le seuil φ sont acceptées comme étant valides comme illustrées par les formules 5.1 et 5.2. En cas d'égalité, seules les relations pour lesquelles les valeurs d'intersection *parent* et *enfant* sont supérieures ou égales à $100 - \varphi$ sont acceptées comme étant valides comme illustré dans la formule 5.3. Dans le cadre de nos travaux, la valeur de φ a été fixée arbitrairement à 10.

$$[\forall p \forall c (hasContinuation(p, c) \wedge hasOverlappingParent(p, op) \wedge hasOverlappingChild(c, oc) \wedge (op = 100) \wedge (op > oc) \wedge ((op - oc) > \varphi) \rightarrow hasGrowth(p, c))] \quad (5.1)$$

$$[\forall p \forall c (hasContinuation(p, c) \wedge hasOverlappingParent(p, op) \wedge hasOverlappingChild(c, oc) \wedge (oc = 100) \wedge (oc > op) \wedge ((oc - op) > \varphi) \rightarrow hasReduction(p, c))] \quad (5.2)$$

$$[\forall p \forall c (hasContinuation(p, c) \wedge hasOverlappingParent(p, op) \wedge hasOverlappingChild(c, oc) \wedge (oc \geq (100 - \varphi)) \wedge (op \geq (100 - \varphi)) \rightarrow hasEquality(p, c))] \quad (5.3)$$

Les formules 5.1, 5.2 et 5.3 sont traduites comme suit en langage SPARQL :

Listing 5.1 – Requête SPARQL pour détecter la relation hasGrowth

```

insert
{
  ?p checksem:hasGrowth ?c
}
where
{
  ?p checksem:hasContinuation ?c .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?Op=100 && ?Op>?Oc && (?Op-?Oc)>10)
}

```


Listing 5.2 – Requête SPARQL pour détecter la relation hasReduction

```

insert
{
  ?p checksem:hasReduction ?c
}
where
{
  ?p checksem:hasContinuation ?c .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?Oc=100 && ?Oc>?Op && (?Oc-?Op)>10)
}

```

Listing 5.3 – Requête SPARQL pour détecter la relation hasEquality

```

insert
{
  ?p checksem:hasEquality ?c
}
where
{
  ?p checksem:hasContinuation ?c .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?Op>90 && ?Oc>90)
}

```

5.2/ RELATIONS DE DÉRIVATION UNAIRE

Les dérivations sont ces relations pour lesquelles l'identité varie entre deux instants de temps consécutifs. Dans notre modèle, nous ne définissons qu'une seule dérivation unique, correspondant à un changement de la couverture d'un timeslice sans modification spatial de celui-ci. Cette relation est appelée *Conversion*. Ce type de relation est très semblable à la relation *Egalité* à la différence que l'identité du timeslice parent est différente de celle de l'enfant. L'équation permettant d'inférer une relation de conversion en utilisant les valeurs d'intersections se définit comme suit :

$$\begin{aligned}
 & [\forall p \forall c (hasDerivation(p, c) \wedge hasOverlappingParent(p, op) \wedge \\
 & \quad hasOverlappingChild(c, oc) \wedge (oc \geq (100 - \varphi)) \wedge \\
 & \quad (op \geq (100 - \varphi)) \rightarrow hasConversion(p, c))]
 \end{aligned}
 \tag{5.4}$$

Listing 5.4 – Requête SPARQL pour détecter la relation hasConversion

```

insert
{
  ?p checksem:hasConversion ?c
}
where
{
  ?p checksem:hasDerivation ?c .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?Op>90 && ?Oc>90)
}

```

5.3/ MOTIFS D'ÉVOLUTION N-AIRE

Dans notre approche, d'autres motifs peuvent être détectés sur la base de multiples relations de filiation. Ces motifs N-aires sont des divisions ou des fusions. Chaque division ou fusion peut impliquer une continuation et au moins une dérivation, ou peut n'impliquer que des dérivations donnant lieu à quatre principaux motifs N-aires : scission, fusion, annexion, séparation. Afin de décrire nos motifs, nous introduisons un autre opérateur *Sum* permettant d'additionner les valeurs d'intersection.

$$Sum^I = \{\langle val, val' \rangle \mid val + val' = res, val, val', res \in \mathbb{Q}\} \quad (5.5)$$

$$\begin{aligned} & [\forall p \forall c \forall c' (hasContinuation(p, c) \wedge hasDerivation(p, c') \wedge \\ & hasOverlappingParent(p, op) \wedge hasOverlappingChild(c, oc) \wedge \\ & hasOverlappingChild(c', oc') \wedge (op = 100) \wedge (Sum(oc, oc') = 100) \\ & \rightarrow (hasSeparation(p, c) \wedge PartOfSeparation(p, c')))] \end{aligned} \quad (5.6)$$

$$\begin{aligned} & [\forall p \forall p' \forall c (hasContinuation(p, c) \wedge hasDerivation(p', c) \wedge \\ & hasOverlappingParent(p, op) \wedge hasOverlappingChild(p', op') \wedge \\ & hasOverlappingChild(c, oc) \wedge (oc = 100) \wedge (Sum(op, op') = 100) \\ & \rightarrow (hasAnnexation(p, c) \wedge PartOfAnnexation(p', c)))] \end{aligned} \quad (5.7)$$

$$\begin{aligned} & [\forall p \forall c \forall c' (hasDerivation(p, c) \wedge hasDerivation(p, c') \wedge \\ & hasOverlappingParent(p, op) \wedge hasOverlappingChild(c, oc) \wedge \\ & hasOverlappingChild(c', oc') \wedge (op = 100) \wedge (Sum(oc, oc') = 100) \\ & \rightarrow (hasSplits(p, c) \wedge hasSplits(p, c')))] \end{aligned} \quad (5.8)$$

$$\begin{aligned} & [\forall p \forall p' \forall c (hasDerivation(p, c) \wedge hasDerivation(p', c) \wedge \\ & hasOverlappingParent(p, op) \wedge hasOverlappingChild(p', op') \wedge \\ & hasOverlappingChild(c, oc) \wedge (oc = 100) \wedge (Sum(op, op') = 100) \\ & \rightarrow (hasMerging(p, c) \wedge hasMerging(p', c)))] \end{aligned} \quad (5.9)$$

Afin de détecter le schéma des motifs N-aires dans notre graphe, nous avons défini deux étapes. La première se concentre uniquement sur la composante spatiale des timelines en cherchant des motifs de divisions et de fusions (la composante d'identité n'est ici pas prise en compte dans l'analyse). Ensuite, la deuxième étape se focalise sur la composante d'identité afin de distinguer la fusion de l'annexion ou encore la division de la séparation. Il est à noter que toute l'analyse pourrait se faire en une seule grande étape. Cependant, dans un souci de clarté, nous avons volontairement distingué ces deux phases, afin d'expliquer comment notre système est mis en œuvre.

Pour effectuer notre première étape, nous avons décidé d'utiliser et insérer des relations temporaires dans notre triplestore : *TemporaryMerger* et *TemporaryDivision*. Ces

relations temporaires correspondent à des motifs de divisions spatiales et de fusions spatiales calculées à l'aide des valeurs d'intersection. Les requêtes suivantes montrent comment ces relations ont été calculées à l'aide des données présentes dans le triplestore.

Listing 5.5 – Requête SPARQL pour le calcul des relations de fusion temporaire

```

insert
{
  ?p checksem:TemporaryMerger ?c
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
      (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?st a checksem:SpatiotemporalTransition .
      ?st checksem:hasParentTimeSlice ?p.
      ?st checksem:hasChildTimeSlice ?c.
      ?p checksem:hasOverlappingParent ?Op.
      ?c checksem:hasOverlappingChild ?Oc.
      FILTER(?Oc=100)
    }
    group by ?c
  }
  FILTER((?SumOp=100)&&( ?countC >1))
}

```

Pour détecter les motifs de fusion, l'idée principale est d'avoir une entité enfant qui contient tous ses parents dans sa géométrie. En d'autres termes, il s'agit de rechercher toutes les relations de filiation impliquant la même entité enfant et plusieurs entité parent. Afin d'obtenir ces résultats en utilisant les valeurs d'intersection, la somme des valeurs d'intersection parent de toutes relations impliquée dans le motif doivent être égales à 100 tandis que la valeur d'intersection enfant doit être de 100 pour chacune d'entre elles.

Listing 5.6 – Requête SPARQL pour le calcul des relations de division temporaire

```

insert
{
  ?p checksem:TemporaryDivision ?c
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
      (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?st a checksem:SpatiotemporalTransition .
      ?st checksem:hasParentTimeSlice ?p.
      ?st checksem:hasChildTimeSlice ?c.
      ?p checksem:hasOverlappingParent ?Op.
      ?c checksem:hasOverlappingChild ?Oc.
      FILTER(?Op=100)
    }
    group by ?c
  }
  FILTER((?SumOc=100)&&( ?countP >1))
}

```

A contrario, pour calculer les motifs de division, l'entité parent doit alors contenir toutes les entités enfant dans sa géométrie. En d'autres termes, nous recherchons toutes les

relations de filiation impliquant la même entité parent et plusieurs entités enfants. En conséquence, la somme des valeurs d'intersection des entités enfants impliquées dans les relations du motif doit être égale à 100, tandis que la valeur d'intersection parent doit être de 100 pour chacune d'entre elles.

Le calcul de la première étape, que nous venons d'aborder, s'intéresse uniquement à la composante spatiale de nos entités. Pour la deuxième étape, nous nous concentrons sur la composante de l'identité, pour compléter les précédentes requêtes, et distinguer des motifs plus complexes. Pour ajouter des informations d'identité dans notre requête, nous utilisons la classe *timeslice* (et les sous-classes de cette classe) stockées dans notre ontologie. Chaque classe représente l'identité d'un timeslice et correspond à une couverture spécifique définie.

Dans un premier temps, nous établissons les requêtes permettant de déduire les motifs complexes de fusion et de division. Ces deux motifs mettent en jeu uniquement des relations de dérivation. Par conséquent, il suffit de s'assurer que, pour chacune des relations de filiation impliquée dans le motif, l'entité parent et l'entité enfant possèdent une couverture différente.

Listing 5.7 – Requête SPARQL pour le calcul des relations de fusion

```

insert
{
  ?p checksem:hasFusion ?c
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
              (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?p checksem:TemporaryMerger ?c .
      ?p a ?classP .
      ?c a ?classC .
      ?p checksem:hasOverlappingParent ?Op.
      ?c checksem:hasOverlappingChild ?Oc.
      FILTER(?Oc=100 && ?classP!=?classC)
    }
    group by ?c
  }
  FILTER((?SumOp=100)&&(countC > 1))
}

```

Listing 5.8 – Requête SPARQL pour l'inférence des relations de division

```

insert
{
  ?p checksem:hasSplit ?c
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
              (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?p checksem:TemporaryDivision ?c .
      ?p a ?classP .
      ?c a ?classC .
      ?p checksem:hasOverlappingParent ?Op.
      ?c checksem:hasOverlappingChild ?Oc.
      FILTER(?Op=100 && ?classP!=?classC)
    }
  }
}

```

```

    group by ?c
  }
  FILTER (((?SumOc)=100)&&(countC > 1))
}

```

Dans les deux cas, l'identité de l'entité parent est comparée avec celle de l'entité enfant. Seules les relations où l'identité est différente entre le parent et l'enfant sont conservées. Dans les cas d'annexion et de séparation, il est nécessaire de détecter deux relations différentes au sein du même motif. L'une correspondant à une continuation et l'autre correspondant à une dérivation. Ainsi, la requête doit parcourir le graphe de notre triplestore pour tenter de trouver des motifs de division spatiale ou de fusion spatiale impliquant au moins une continuation et une dérivation. Les requêtes suivantes montrent comment détecter ces motifs de séparation et d'annexion en respectant les contraintes définies.

Listing 5.9 – Requête SPARQL pour l'inférence des relations d'annexion

```

insert
{
  ?p1 checksem:hasAnnexion ?c .
  ?p2 checksem:PartOfSeparation ?c .
}
where
{
  {
    select ?c (sum(?Oc) as ?SumOc)
    (count(?p) as ?countP) (sum(?Op1) as ?sumOp1)
    (sum(?Op2) as ?SumOp2)
    where{
      ?p1 checksem:TemporaryDivision ?c .
      ?p2 checksem:TemporaryDivision ?c .
      ?p1 a ?classP1 .
      ?p2 a ?classP2 .
      ?c a ?classC .
      ?p1 checksem:hasOverlappingParent ?Op1.
      ?p2 checksem:hasOverlappingParent ?Op2.
      ?c checksem:hasOverlappingChild ?Oc.
      FILTER (?Op=100 && ?classP1=?classC &&
      ?classP2!=?classC2 )
    }
    group by ?c
  }
  FILTER (((?SumOp1+?SumOp2)=100)&&(countC > 1))
}

```

Listing 5.10 – Requête SPARQL pour l'inférence des relations de séparation

```

insert
{
  ?p checksem:hasSeparation ?c1 .
  ?p checksem:PartOfSeparation ?c2 .
}
where
{
  {
    select ?c1 ?c2 (sum(?Oc1) as ?SumOc1)
    (sum(?Oc2) as ?SumOc2)
    (count(?p) as ?countP) (sum(?Op) as ?sumOp)
    where{
      ?p checksem:TemporaryDivision ?c1 .
      ?p checksem:TemporaryDivision ?c2 .
      ?p a ?landcoverP .
      ?c1 a ?classC1 .
    }
  }
}

```

```

?c2 a ?classC2 .
?p checksem:hasOverlappingParent ?Op.
?c1 checksem:hasOverlappingChild ?Oc1.
?c2 checksem:hasOverlappingChild ?Oc2.
FILTER(?Op=100 && ?classP=?classC1
&& ?classP!=?classC2)
}
group by ?c1 ?c2
}
FILTER(((?SumOc1+?SumOc2)=100)&&( ?countC > 1))
}

```

La figure 5.1 montre le résultat obtenu de l'analyse des motifs. Entre $T1$ et $T2$, l'analyse a détecté un motif d'annexion (impliquant $ts1$, $ts2$ et $ts3$) et un motif de séparation (impliquant $ts2$, $ts3$ et $ts4$). Entre $T2$ et $T3$, le système a détecté un motif de séparation (impliquant $ts3$, $ts5$ et $ts6$) et un motif de conversion (impliquant $ts4$ et $ts7$).

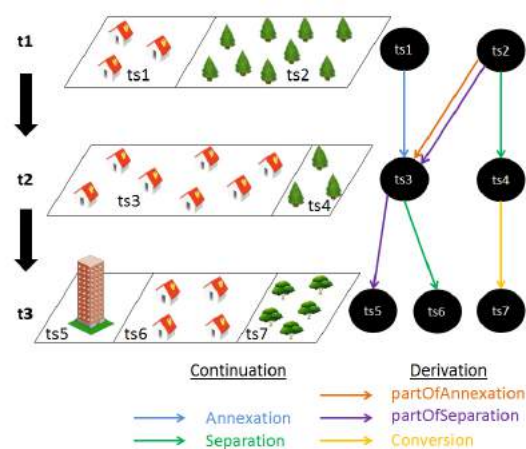


FIGURE 5.1 – Résultat de l'analyse des motifs

Ce chapitre a montré comment implémenter la 3ème couche du modèle Continuum. Pour cela, un ensemble de motifs spatiaux génériques a été défini au travers d'un ensemble de relations dérivées de la couche 2. Bien que ces motifs offrent une représentation de la dynamique de l'environnement permettant de comprendre partiellement l'évolution, la sémantique des entités spatio-temporelles et du contexte n'est pas prise en considération. Par conséquent, il est nécessaire de prendre en compte la sémantique des entités afin d'interpréter cette dynamique spatiale et d'offrir une représentation mentale au sein du monde réel, en qualifiant les phénomènes sous-jacents. La découverte de phénomènes dynamiques, en se basant sur la connaissance liée à l'environnement géospatial, fait l'objet du prochain chapitre.

COUCHE 4 : DÉCOUVERTE DE CONNAISSANCE

Ce chapitre présente le principal apport de cette thèse et décrit la 4ème couche du modèle. Contrairement aux autres, cette couche ne propose pas de relations génériques, mais tente de décrire des phénomènes du monde réel en interprétant les motifs génériques de la 3ème couche à l'aide des informations contextuelles définies dans l'ontologie. Le modèle Continuum permet d'étudier la profondeur des changements afin d'offrir une interprétation des motifs au plus proche de la réalité. Nous étudierons également les capacités du modèle pour analyser un processus spatio-temporel, c'est-à-dire un ensemble de phénomènes liés et contigus dans le temps. Enfin, nous introduirons la prise en compte des relations d'adjacence pour permettre l'étude des phénomènes connexes, c'est-à-dire mettre en évidence la causalité d'évolution en fonction des évolutions des parcelles voisines.

Sommaire

6.1 Étude de la profondeur du changement	115
6.2 Les motifs de tendance : du phénomène au processus	123
6.3 Étude des évolutions connexes	128

6.1/ ÉTUDE DE LA PROFONDEUR DU CHANGEMENT

Les dérivations sont d'une importance capitale dans le cas d'une annexion et d'une séparation. En effet, nous considérons que lorsqu'une entité est annexée, la nouvelle parcelle, venant se greffer sur cette entité, se substitue alors à une ou plusieurs autres entités, existant antérieurement et possédant une couverture terrestre différente. Inversement, lorsqu'une entité se divise, au moins une des parcelles existant antérieurement, est substituée par une ou plusieurs autres entités possédant une couverture différente. En conséquence, l'annexion et la séparation sont inévitablement liées à une ou plusieurs dérivations.

Dans ces travaux, nous soutenons que les dérivations aident à obtenir une meilleure compréhension de l'évolution d'un territoire. Par exemple, si sur un territoire, les espaces correspondant à des zones forestières, diminuent fréquemment via des motifs de division ou de séparation, et si la couverture terrestre des parcelles ayant substitué l'espace forestier perdu sont fréquemment des espaces urbains continus, alors nous pouvons en

déduire que la déforestation est fortement liée à l'urbanisation.

Dans cet exemple, la parcelle substituée (la zone forestière) a subi une forte évolution de la couverture terrestre. En effet, la transformation d'une zone forestière en zone urbaine révèle un changement sémantiquement fort au niveau de la couverture terrestre et donc de l'identité des parcelles impliquées. Cependant, tous les changements de couvertures ne sont pas aussi marqués et révèlent la nécessité de nuancer l'impact des dérivations au sein d'un territoire.

Par exemple, lorsqu'une couverture terrestre représentant une forêt de conifères se transforme en couverture représentant une forêt mixte, la sémantique des deux couvertures reste relativement similaire. De plus, si la forêt de conifères se rétrécit via un motif de séparation et que la parcelle subissant la dérivation se transforme en forêt mixte, cette information est pertinente seulement si l'étude du territoire porte sur l'évolution des différents types de forêt. Toutefois, si le champ d'étude de l'utilisateur est plus générique, par exemple l'étude des espaces forestiers indépendamment du type de forêt, l'information ne serait pas pertinente, et aucun motif de séparation ne serait détecté car il n'y aurait aucun changement d'identité.

Les exemples précédents soulignent le besoin de traiter différents niveaux sémantiques pour étudier les évolutions du territoire. Dans ces travaux, nous utilisons la hiérarchie, telle que définie dans le jeu de données Corine Land Cover, en transposant la hiérarchie de couvertures terrestres en hiérarchie de classes dans notre ontologie. Elle offre ainsi à notre système des possibilités d'analyse sur différents niveaux sémantiques.

Le nombre de niveaux sémantiques pouvant être analysés dépend entièrement de la profondeur de la hiérarchie de la couverture terrestre de la classification. Dans le modèle Continuum, les couvertures terrestres des timeslices possèdent la particularité d'être organisées au sein d'une taxonomie permettant de regrouper ces timeslices à l'aide de concepts plus ou moins génériques. Ces concepts sont organisés selon plusieurs niveaux hiérarchiques de la façon suivante :

$$C_n \sqsubseteq \dots \sqsubseteq C_i \sqsubseteq \dots \sqsubseteq C_0 \quad (6.1)$$

où C_0 correspond à la classe Timeslice et $n \in \mathbb{N}$ représente la profondeur de la hiérarchie.

La figure 6.1 illustre l'idée de profondeur de la hiérarchie de concept. Sur les bases de cette hiérarchie, plusieurs degrés de changement peuvent alors être déterminés. Dans le contexte de profondeur du changement, un changement est assimilé à une dérivation. Ainsi, lors d'une dérivation, on observe un changement de l'identité, qui se traduit par un changement depuis une classe vers une autre classe au sein de la hiérarchie de concept.

Dans les applications spatio-temporelles classiques, aucune hiérarchie n'est définie pour représenter les couvertures terrestres des parcelles d'un territoire. Ou bien, il est possible de considérer qu'il existe une hiérarchie ne possédant qu'un seul niveau. Par conséquent, le degré de changement est le même pour chacune des dérivations pouvant intervenir au sein de l'application. L'ajout de niveaux supplémentaires, au sein de la hiérarchie, induit la possibilité de décrire autant de degrés de changement qu'il existe de niveaux dans la hiérarchie. En considérant notre hiérarchie comme un arbre, alors plus le changement intervient vers la racine de l'arbre et plus ce changement révèle une forte modification de l'identité du timeslice. De même, plus le changement s'opère sur un niveau proche de

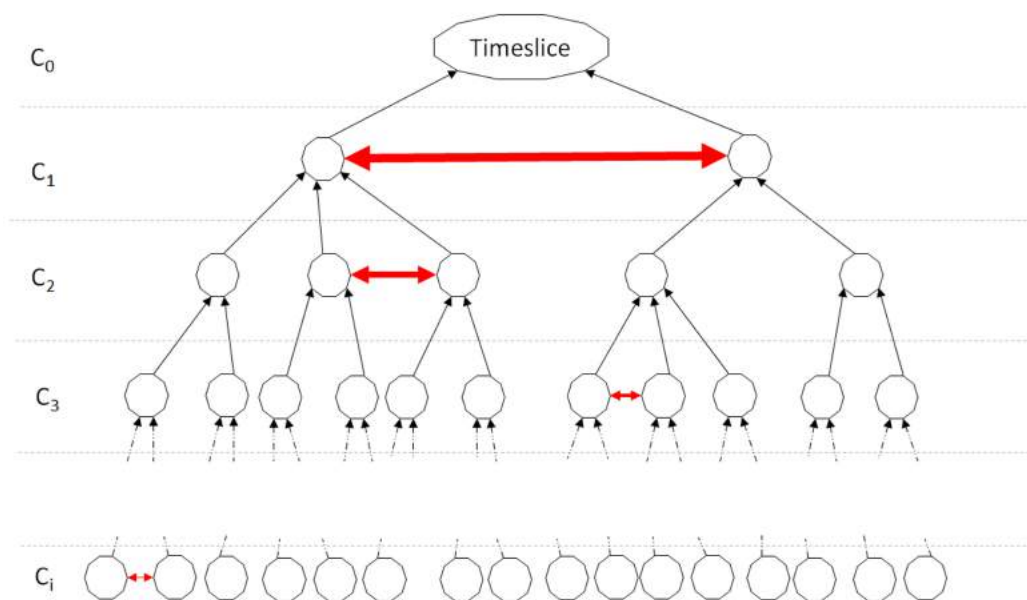


FIGURE 6.1 – Illustration de la hiérarchie de concept et de la profondeur de changement associée

celui des feuilles de l'arbre, et plus le changement sera faible voir négligeable. La figure 6.1 illustre un exemple de changement sur chacun des niveaux de la hiérarchie.

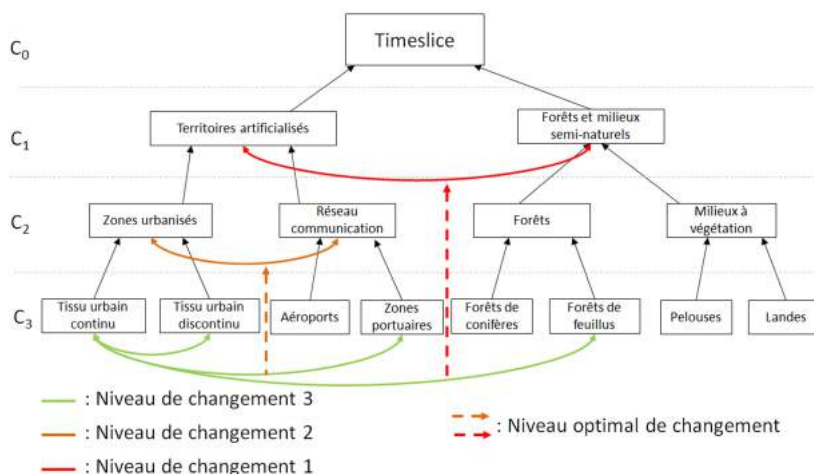


FIGURE 6.2 – Illustration du niveau optimal de changement

Toutefois, il est intéressant de noter que quel que soit le changement, celui-ci existe obligatoirement au niveau des feuilles de l'arbre comme le montre la figure 6.2. En effet, tout changement sur des concepts proches du niveau de la racine de l'arbre, existe sur les sous-concepts constituant le niveau des feuilles de l'arbre. En revanche, l'inverse n'est pas vrai. Se limiter à l'étude des changements au niveau des feuilles de l'arbre, n'est pas intéressant dans notre propos car cela revient à ne pas utiliser la hiérarchie et ne permet donc pas de discriminer les changements sur différents degrés. Par conséquent,

en partant du niveau des feuilles de l'arbre, il est nécessaire de remonter les niveaux de l'arbre afin de déterminer le niveau le plus haut sur lequel se répercute le changement au sein de la hiérarchie. Nous appellerons ce niveau le **niveau optimal du changement**. C'est ce niveau optimal du changement qui permettra d'évaluer la profondeur d'un changement. Dans un souci de clarté, il est nécessaire d'introduire quelques notions relatives à notre propos au travers des définitions suivantes :

- **Niveau de changement** : il correspond au niveau de concept de la hiérarchie sur lequel un changement est décrit.
- **Niveau de changement optimal** : quel que soit le changement, il peut, à minima, être représenté entre des concepts du dernier niveau de la hiérarchie. Dans ce cas, tous les changements sont équivalents. Le niveau optimal de changement est le niveau de changement le plus proche de la racine pour lequel il est possible de représenter ce changement au sein de la hiérarchie. Il est obtenu en cherchant le concept subsumant commun le plus spécifique des concepts *parents* et *enfants*
- **Profondeur du changement** : elle est directement liée au niveau optimal de changement. Plus le niveau optimal de changement se situe proche de la racine, plus le changement est profond. A l'inverse, plus le niveau optimal de changement se situe proche des feuilles de la hiérarchie, plus le changement est faible.
- **Classe de changement** : le niveau optimal de changement décrit des indices numériques permettant de mesurer la profondeur du changement. La classe de changement permet de classer ces indices au sein de sous-ensembles. Par exemple, imaginons, la classe de changement appelée *faible* et définie par l'ensemble des changements dont le niveau optimal est compris entre 5 et 6. A noter, que plus les indices sont élevés plus le changement est faible dans le modèle Continuum.

Listing 6.1 – Algorithme pour obtenir le niveau optimal d'un changement

```

Fonction NiveauOptimal(ConceptPere , ConceptFils , TailleHierarchie)
  NivOptimal = TailleHierarchie + 1
  TANT.QUE (ConceptPere != ConceptFils)
    ConceptPere = RetourneSuperConcept(ConceptPere)
    ConceptFils = RetourneSuperConcept(ConceptFils)
    NivOptimal = NivOptimal - 1
  Fin TANT.QUE
  Retourner NivOptimal
Fin Fonction

```

L'algorithme ci-dessus permet de récupérer le niveau optimal de changement. Les données en entrée sont les concepts père et fils, c'est-à-dire les concepts représentant l'identité d'une parcelle avant et après le changement, ainsi que la taille de la hiérarchie. Les concepts pères et fils sont ceux définis sur les feuilles de l'arbre. Pour déterminer le niveau optimal du changement, il suffit de comparer le concept père et le concept fils à mesure que l'on remonte les différents niveaux de l'arbre en direction de la racine afin d'étudier le niveau à partir duquel ceux-ci sont identiques. Ainsi, tant que le concept père et le concept fils sont différents, l'algorithme va chercher le super-concept du père et du fils afin de remonter d'un niveau dans la hiérarchie. Lorsque les concepts père et fils sont finalement identiques alors cela signifie que le niveau optimal du changement se situe au niveau directement inférieur.

Par exemple, si l'algorithme s'arrête au niveau 0 de la hiérarchie alors les variables *ConceptPere* et *ConceptFils* ont pour valeur "Timeslice" et sont donc identiques. Le ni-

veau optimal de changement est alors le niveau 1. On notera au passage que ce cas révèle le changement le plus fort parmi les différents degrés offerts par la hiérarchie. La figure 6.3 illustre un exemple visuel du déroulement de l'algorithme du niveau optimal de changement.

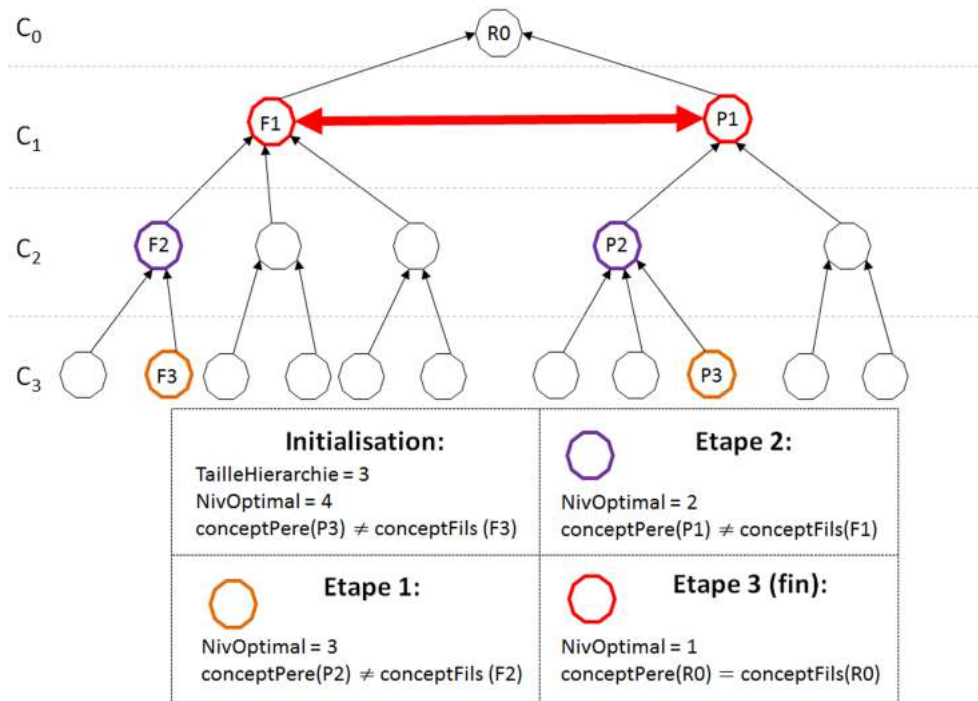


FIGURE 6.3 – Déroulement de l'algorithme du niveau optimal de changement

Comme nous venons de le voir, il existe autant de degrés d'étude de profondeur du changement qu'il existe de niveaux dans la hiérarchie de concept. Pour autant, sur des applications utilisant des hiérarchies avec de nombreux niveaux, un tel niveau de détail sur le changement est rarement nécessaire. Pour résoudre ce problème, nous proposons d'utiliser la logique classique afin de répartir les différents degrés de changement au sein de classes de changement. Pour cela, nous utilisons une variable linguistique correspondant au triplet (V, X, T_V) avec V une variable, X la plage de valeurs de cette variable et $T_V = \{A_1, A_2, \dots, A_n\}$ un ensemble fini de sous-ensembles. Dans le cas du jeu de données Corine Land cover, nous définissons la variable linguistique suivante :

$$\begin{aligned}
 V &= \text{profondeur du changement} \\
 X &= [C_0, C_i] \\
 T_V &= \{A_1, A_2, \dots, A_n\}
 \end{aligned}
 \tag{6.2}$$

avec X défini par la taille de la hiérarchie et T_V permettant de classifier tous les changements selon un nombre fini de sous-ensembles ordonnés tel que :

$$\bigcup_{i=1}^n A_i = X
 \tag{6.3}$$

Ensuite, à l'aide d'une fonction d'appartenance, il est possible de déterminer l'appartenance d'un changement à l'un des ensembles défini par T_V en fonction de la valeur du niveau optimal de changement.

Supposons x un entier avec $x \in X$ et A_i un sous-ensemble de T_V alors la fonction d'appartenance se définit comme suit :

$$A_i(x) = \begin{cases} 1 & \text{si } x \in A_i \\ 0 & \text{si } x \notin A_i \end{cases} \quad (6.4)$$

A présent, prenons une variable linguistique pour laquelle T_V définit arbitrairement trois sous-ensembles *faible*, *moyen* et *fort* permettant de classifier les changements. L'équation 6.5 représente cette variable.

$$\begin{aligned} V &= \text{profondeur du changement} \\ X &= [C_0, C_n] \\ T_V &= \{\text{faible}, \text{moyen}, \text{fort}\} \end{aligned} \quad (6.5)$$

En supposant une hiérarchie composée de 6 niveaux, il est possible de classifier arbitrairement les changements au sein des trois sous-ensembles tel que :

$$\begin{aligned} \text{fort} &= \{C_1, C_2\} \\ \text{moyen} &= \{C_3, C_4, C_5\} \\ \text{faible} &= \{C_6\} \end{aligned} \quad (6.6)$$

Ensuite à l'aide de la fonction d'appartenance, il est possible de classifier chacun des changements. Par exemple :

$$\begin{aligned} \text{fort}(C_1) &= 1 \\ \text{fort}(C_6) &= 0 \end{aligned} \quad (6.7)$$

Le jeu de données Corine Land Cover, quant à lui, propose une hiérarchie en trois niveaux de classification, pour cet exemple, notre système est capable de détecter trois niveaux de changement. Logiquement, nous utiliserons à nouveau les termes de : faible changement, moyenne changement et forte changement pour transcrire la profondeur du changement. La figure 6.4 illustre ces différents niveaux de dérivation sur le jeu de données Corine Land Cover.

En utilisant les informations de profondeur de changement, il est intéressant de noter que les motifs peuvent conduire à des interprétations très différentes. Dans notre exemple, nous avons détecté deux motifs de séparation, un motif d'annexion et un motif de conversion. Le premier motif de séparation se produit entre t1 et t2 avec une partie de la forêt de conifères qui se convertit en tissu urbain continu. Cette dérivation révèle un changement fort car il implique un changement de classe au premier niveau de la hiérarchie Corine Land Cover. Pour ce premier cas, il est facilement possible de conclure à un phénomène de déforestation concomitant à un phénomène d'urbanisation (à noter que l'analyse met en évidence le lien de causalité entre l'urbanisation et la déforestation).

A présent, intéressons nous au deuxième motif de séparation pour lequel une partie du tissu urbain continu se convertit en une zone industrielle et commerciale. De prime abord, une conclusion naïve nous incite à considérer qu'il s'agit ici d'un phénomène de déclin urbain puisque la couverture du tissu urbain continu se réduit. Cependant, en utilisant

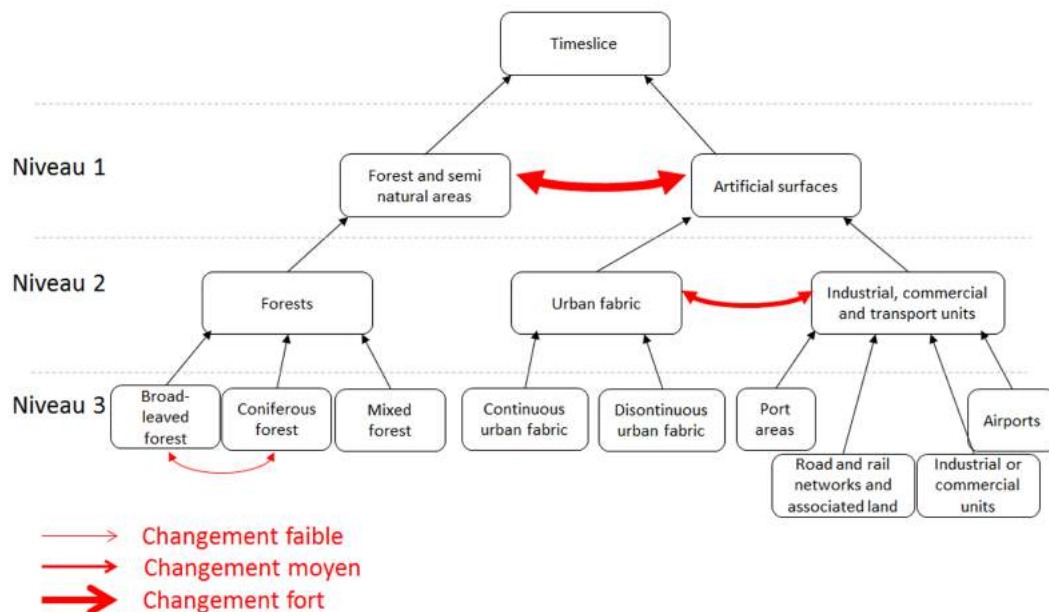


FIGURE 6.4 – Exemple de types de dérivation du jeu de données Corine Land Cover

l'analyse de profondeur du changement, nous nous rendons compte qu'il s'agit là d'une dérivation moyenne ce qui signifie que la partie de tissu urbain ayant disparu a été remplacée par une parcelle de couverture terrestre sémantiquement proche. Par conséquent, contrairement au premier cas, la conclusion d'un déclin urbain, bien que possible, ne semble pas être le phénomène le plus probable à interpréter. Le tissu urbain s'étant partiellement converti en zone industrielle et commerciale, l'hypothèse la plus plausible nous porte à croire qu'il s'agit d'un processus d'intensification urbaine. La figure 6.6 représente le résultat de l'analyse.

La figure 6.5 présente le résultat de l'analyse permettant de distinguer les différents niveaux de changement.

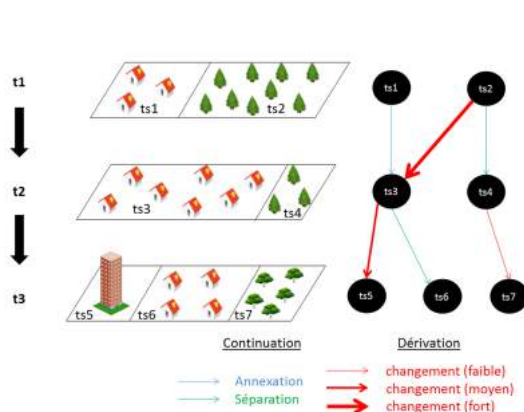


FIGURE 6.5 – Détection de motifs basée sur l'analyse de la taxonomie

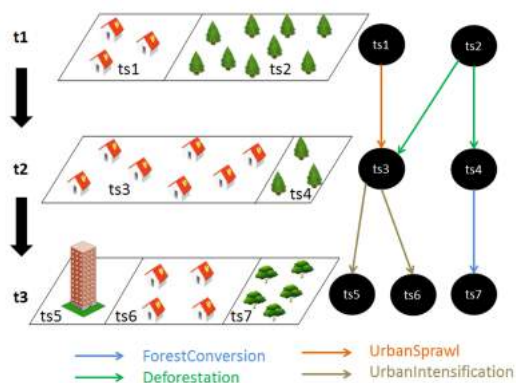


FIGURE 6.6 – Interprétation de phénomènes

Le tableau 6.1 montre un exemple d'analyse qui peut être effectuée en utilisant la profondeur d'un changement sur le jeu de données Corine Land Cover limité à la région de la Gironde. Dans ce tableau, nous avons calculé le nombre de timeslices et la surface touchée par des motifs de conversion. En outre, toutes les couvertures terrestres avant et après changement sont données. Enfin, chaque changement est également réparti en fonction de sa profondeur : faible, moyenne ou forte.

TABLE 6.1 – Conversion

(faible) Conversion	1990 to 2000		2000 to 2006	
	Nb	Area (dam ²)	Nb	Area (dam ²)
Construction to Dumps	-	-	2	8338
NaturalGrassLands to VegMoors	-	-	1	902162
NaturalGrassLands to ShrubWoodland	-	-	1	3189
ArableNonirrigated to ArablePermlrrigated	-	-	2	105443
ConiferousForest to MixedForest	-	-	1	37667
AgricAndNatural to complexCultivation	-	-	1	4336
IndCommercial to Ports	-	-	2	13008
Mines to Dumps	-	-	1	2731
(moyen) Conversion	1990 to 2000		2000 to 2006	
	Nb	Area (dam ²)	Nb	Area (dam ²)
Forest to ShrubVegetation	5	22403	10	133433
ShrubVegetation to Forest	23	113225	-	-
ArableLand to HeterogenousAgric	1	2607	5	22885
Pasture to HeterogeneousAgric	-	-	11	69725
HeterogeneousAgric to PermanentCrops	-	-	1	5374
MinesDumpConstruction to IndustrialCommercial-Transportation	-	-	3	8501
ArableLand to pasture	-	-	7	54473
MinesDumpConstruction to ArtificialVegetated	-	-	1	2965
ArableLand to PermanentCrops	-	-	1	2516
HeterogeneousAgric to Pasture	-	-	3	13392
(fort) Conversion	1990 to 2000		2000 to 2006	
	Nb	Area (dam ²)	Nb	Area (dam ²)
ForestSemiNaturalAreas to AgriculturalAreas	1	19080	2	11962
AgriculturalAreas to ArtificialSurfaces	2	6871	4	18761
ArtificialSurfaces to WaterBodies	2	7600	3	17248
ArtificialSurfaces to AgriculturalAreas	-	-	1	4000
ForestSemiNaturalAreas to ArtificialSurfaces	-	-	1	3865
AgriculturalAreas to ForestSemiNaturalAreas	-	-	2	16255

Dans cet exemple, nous remarquons rapidement un phénomène étrange de conversion de surfaces artificielles vers des étendues d'eau. D'après [Salomon, 2002], [LABOREE12 et al., 2012], la Gironde est une région très touchée par les inondations principalement à cause du fleuve qui la traverse : la Garonne. Pour faire face à ce problème récurrent le paysage de la Gironde a subi quelques transformations dans l'optique de gérer les risques liés aux inondations urbaines conformément au Plan de Prévention des Risques d'Inondation (PPRI) régi par le code de l'environnement. Son objectif est d'établir une cartographie des zones de risque et d'interdire, de limiter ou de soumettre à prescription, les constructions nouvelles. Il permet aussi de prescrire des mesures pour réduire la vulnérabilité des installations et constructions existantes. La figure 6.7 illustre les conséquences de la mise en application de ce plan de prévention.

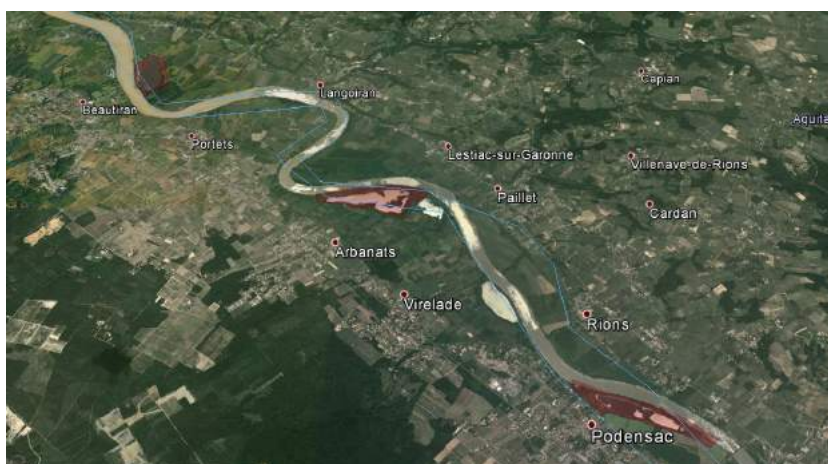


FIGURE 6.7 – Timeslices transformés de surface artificielle en étendue d'eau le long de la Garonne

6.2/ LES MOTIFS DE TENDANCE : DU PHÉNOMÈNE AU PROCESSUS

La compréhension de l'évolution ne se limite pas obligatoirement à un seul intervalle de changement. Certains phénomènes se produisent progressivement sur plusieurs intervalles de changement successifs. Dans ce travail, nous avons appelé motifs de tendance l'étude des phénomènes basée sur la récurrence des motifs cités dans les précédentes sections. La détection de motifs de tendance peut être assimilée à la recherche de chemin au sein du graphe.

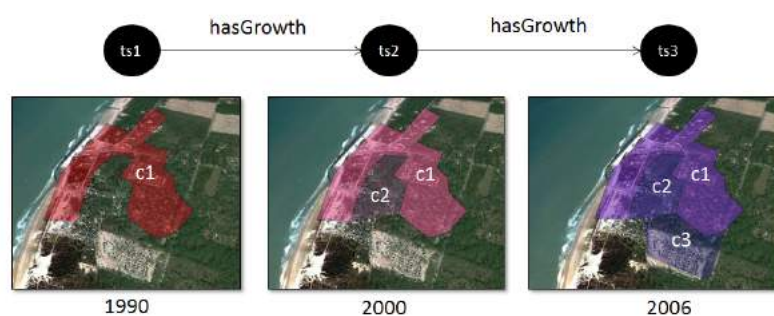


FIGURE 6.8 – Exemple de motif de tendance

Comme notre modèle Continuum est un graphe RDF, le langage SPARQL Query associé à ces technologies apparaît comme particulièrement adapté pour détecter des motifs de tendance. Plus précisément, nous utiliserons un outil particulier, inclus dans SPARQL Query, appelé *chemin de propriété*. Un *chemin de propriété* est un chemin empruntable dans un graphe entre deux nœuds du graphe. Le cas le plus trivial concerne un chemin de propriété de longueur 1, ce qui équivaut à 1 triplet. Les *chemins de propriété* permettent de fournir des expressions régulières plus concises pour détecter des motifs en SPARQL au sein du graphe. De plus, ils offrent également la possibilité d'établir

une connectivité entre deux ressources par un chemin de longueur arbitraire. Ici, nous présentons quelques exemples pour introduire les capacités offertes par les chemins de propriété :

Alternatives Retourner les tissus urbains ayant subi une conversion ou une séparation

```
{
ck:TSUrban1990 ck:hasConversion|ck:PartOfSeparation ?TS2000
}
```

Les relations interrogées sont des dérivations. Par conséquent, les timeslices enfants possèdent obligatoirement une identité différente de celle du parent. Il est alors facile d'étudier précisément vers quelle couverture chaque timeslice a évolué.

Séquence Trouver toutes les nouvelles couvertures des tissus urbains ayant subi une conversion.

```
{
?TS1990 rdf:type ck:UrbanDiscontinuous .
?TS1990 ck:hasConversion/rdf:type ?LandCover2000 .
}
```

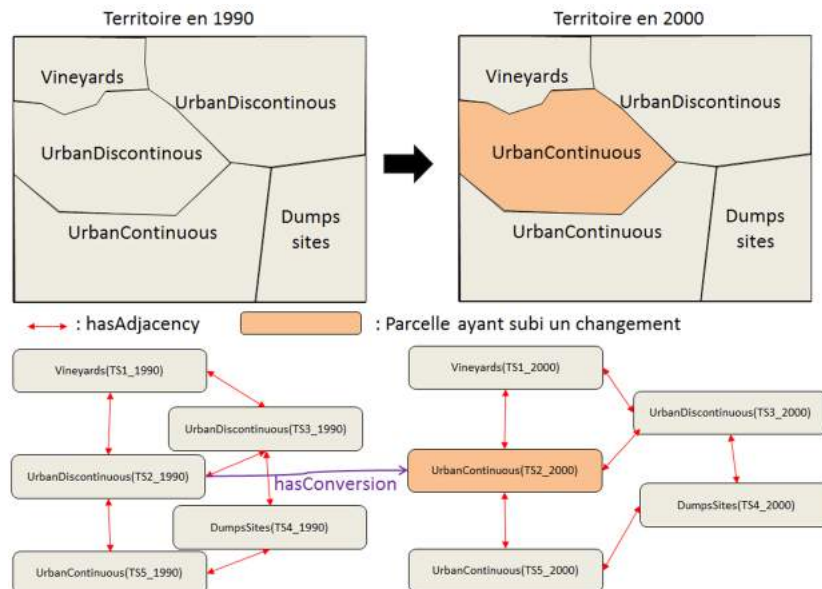


FIGURE 6.9 – Toutes les parcelles converties en tissu urbain

Séquence Trouver toutes les couvertures des parcelles voisines des voisins du timeslice *timeslice_urban1*.

```
{
ck:timeslice_urban1 ck:hasAdjacency/ck:hasAdjacency/rdf:type ?LandCover .
}
```

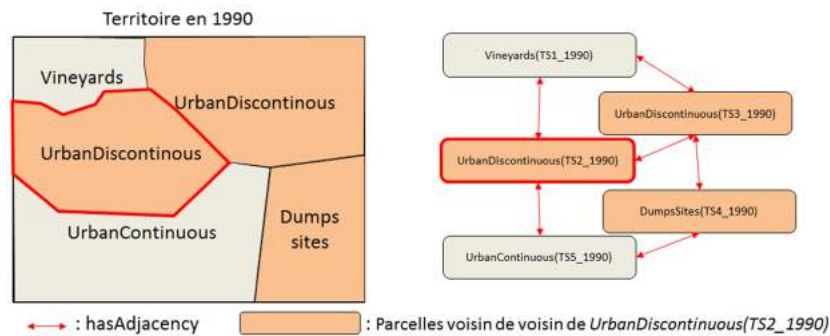


FIGURE 6.10 – Toutes les parcelles voisines de voisines d'un timeslice

Filterer les doublons Du fait que checksem :hasAdjacency est une propriété symétrique, l'exemple précédent peut inclure le timeslice *timeslice_urban1* lui-même comme résultat. Ceci peut être évité avec cette requête :

```
{
  ck:timeslice_urban1 ck:hasAdjacency/ck:hasAdjacency ?ts.
  FILTER ( ck:timeslice_urban1 != ?ts )
  ?ts rdf:type ?LandCover
}
```

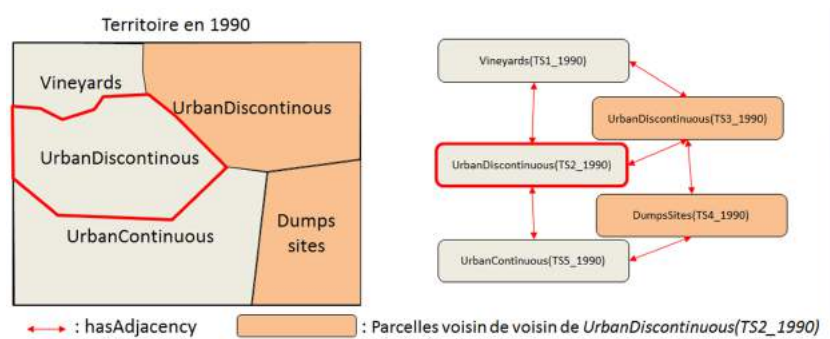


FIGURE 6.11 – Toutes les parcelles voisines de voisines d'un timeslice sans doublons

Chemin de propriété inversés Ces deux requêtes sont équivalentes : la seconde inverse seulement la direction de la propriété en inter-changeant le sujet et l'objet du triplet.

```
{ ?ts rdf:type ?LandCover }
{ ?LandCover ^rdf:type ?ts }
```

Séquences inversées Trouver tous les timeslices qui ont été annexés.

```
{
  ?x ck:hasAnnexation/^ck:PartOfAnnexation ?y .
  FILTER(?x != ?y)
}
```

ce qui est équivalent à (?gen1 est une variable générée par le système) :

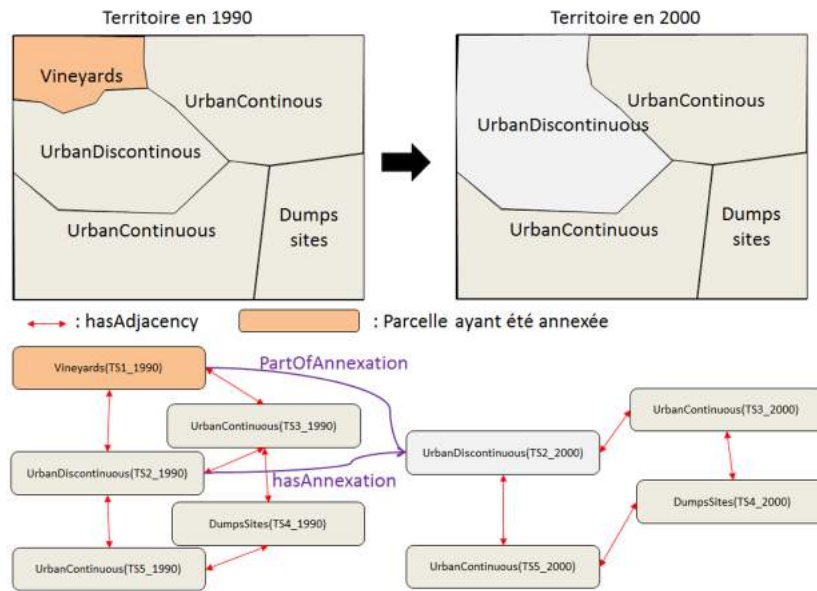


FIGURE 6.12 – Toutes les timeslices annexés

```

{
  ?x ck:hasAnnexation ?gen1 .
  ?y ck:PartOfAnnexation ?gen1 .
  FILTER(?x != ?y)
}

```

Chemin de propriété de longueur arbitraire trouver tous les timeslices pouvant être atteints depuis *checksem :timeslice_urban1* via la propriété *checksem :hasGrowth*. La requête renvoie également les timeslices atteints lorsque la relation *checksem :hasGrowth* apparaît plusieurs fois dans le chemin de propriété :

```

{
  ck:timeslice_urban1 rdf:type ck:UrbanContinuous .
  ck:timeslice_urban1 ck:hasGrowth+ ?ts .
}

```

Alternatives dans un chemin de propriété de longueur arbitraire

```

{
  ck:timeslice_urban1 (ck:hasGrowth|ck:hasAnnexation)+ ?ts
}

```

Chemin de propriété de longueur arbitraire Quelques formes de calcul sont également possibles. Par exemple, en RDFS, donner toutes les classes et super-classes d'une ressource :

```

{ <http://example/thing> rdf:type/rdfs:subClassOf* ?type }

```

Donner toutes les ressources et leurs classes inférées:

```

{ ?x rdf:type/rdfs:subClassOf* ?type }

```

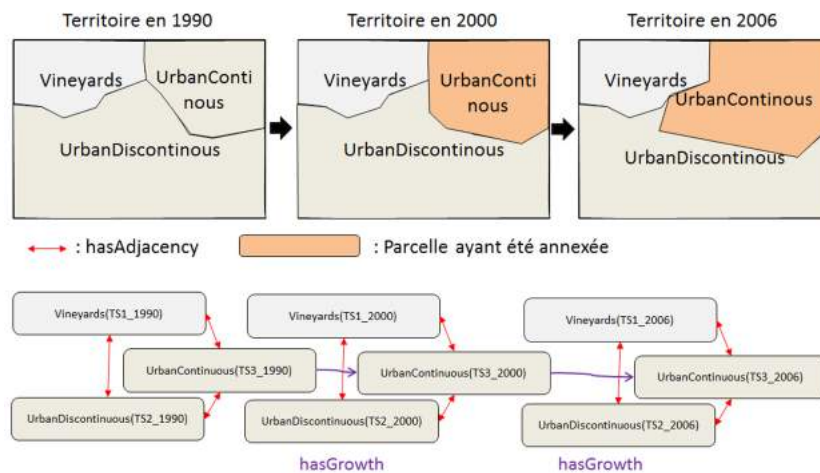


FIGURE 6.13 – Tous les timeslices impliqués dans un processus de croissance

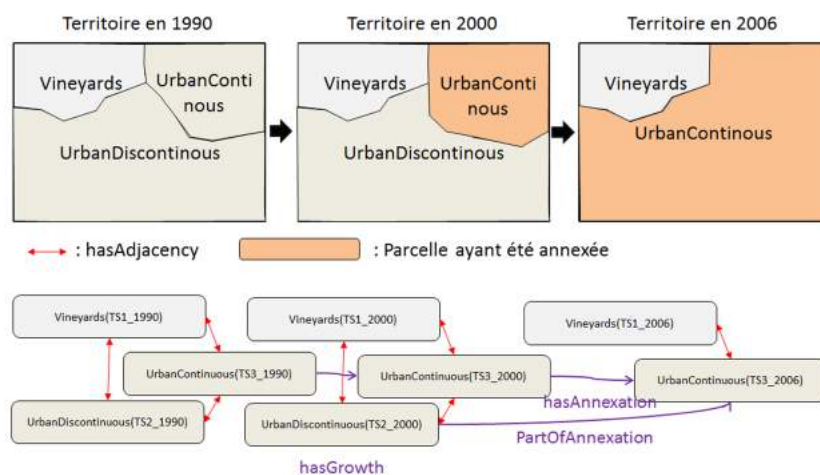


FIGURE 6.14 – Tous les timeslices impliqués dans un processus de croissance ou d'annexion

Sous-propriétés Donner toutes les sous-propriétés du jeu de données

```
{
  ?x ?p ?v .
  ?p rdfs:subPropertyOf* :property .
}
```

Chemin de propriété prohibé Trouver tous les noeuds connectés sauf ceux connectés par la relation rdf:type (ni dans un sens ni dans l'autre) :

```
{ ?x !(rdf:type|^rdf:type) ?y }
```

Cette section a montré comment il est possible de modéliser, à l'aide d'une requête, un motif de tendance et de le rechercher dans le graphe.

6.3/ ÉTUDE DES ÉVOLUTIONS CONNEXES

Jusqu'à présent, notre compréhension de l'évolution se basait uniquement sur une vision des parcelles en fonction de l'espace, du temps et de l'évolution d'une parcelle mais indépendamment les unes des autres. Cette vision permet d'extraire de la connaissance liée au cycle de vie des parcelles en répondant à des questions telles que : *Quand, quoi, où et comment est le changement ?*.

Cependant, certaines interactions peuvent se produire entre des parcelles existantes sur un même intervalle de temps en particulier lorsque l'on étudie le voisinage de chacune des parcelles. D'une manière générale, l'étude du voisinage d'une entité spatio-temporelle ouvre la voie vers de nouvelles connaissances en répondant à la question : *pourquoi le changement a eu lieu ?*.

Dans ces travaux, notre application se limite à l'usage de données parcellaires. Par conséquent, l'étude du voisinage sur des données parcellaires fait office de preuve de concept. L'emploi futur d'autres structures de données, telles que des lignes ou des points associés aux données parcellaires, offrent des perspectives intéressantes, qui n'ont pas été étudiées dans ces travaux.

Considérons par exemple, le cas de l'étude de l'impact de la construction d'une route sur le paysage environnant. La figure 6.15 illustre un exemple de ce type, à l'exception près que les routes sont représentées sous forme de parcelles dans le jeu de données Corine Land Cover. Dans cette étude de cas, nous nous concentrons sur trois parcelles possédant chacune une identité distincte (Une route, une zone industrielle ou commerciale et une zone agricole ou espace naturelle). L'étude de cas est représentée sur les trois points de temps de notre jeu de données : 1990, 2000, 2006. Le graphe stocké dans notre triplestore et les impressions d'écran de notre application sont illustrés dans la figure 6.15.

Le graphe se compose de 3 timeslices *R1990*, *Ind1990* et *Ag1990* appartenant respectivement aux classes **route**, **zone industrielle ou commerciale** et **zone agricole ou espace naturelle** au début du processus. *Ind1990* et *Ag1990* sont reliés par une relation d'adjacence tandis *R1990* est disjoint de ces deux timeslices. Entre 1990 et 2000, les timeslices *R1990*, *Ind1990* et *Ag1990* sont liés dans le temps respectivement à *R2000*, *Ind2000* et *Ag2000* par une relation de continuation. Plus précisément, *R1990* est relié à *R2000* grâce à une relation d'expansion tandis que *Ind1990* et *Ag1990* sont reliés respectivement à *Ind2000* et *AG2000* par une relation d'égalité. Si la relation d'adjacence continue d'exister entre *Ind2000* et *AG2000*, une nouvelle relation d'adjacence s'est également créée entre *Ind2000* et *R2000*. Enfin, entre 2000 et 2006, les timeslices *R2000*, *Ind2000* et *AG2000* sont reliés dans le temps respectivement à *R2006*, *Ind2006* et *Ag2006* par une relation de continuation. Cette fois, *R2000* est relié à *R2006* grâce à une relation d'égalité tandis que *Ind2000* et *AG2000* sont reliés respectivement à *Ind2006* et *Ag2006* par une relation d'expansion et une relation de réduction.

En se basant sur ce graphe de connaissance, il est possible alors de donner une interprétation pour expliquer la situation. Comme le montre les images de la figure 6.15, nous pouvons constater qu'entre 1990 et 2000, une route a été étendue pour relier la zone industrielle ou commerciale. Comme cette parcelle est devenue plus accessible aux consommateurs, elle s'est alors développée en élargissant son infrastructure initiant un processus d'urbanisation. Cependant, du fait de l'adjacence de cette zone avec une

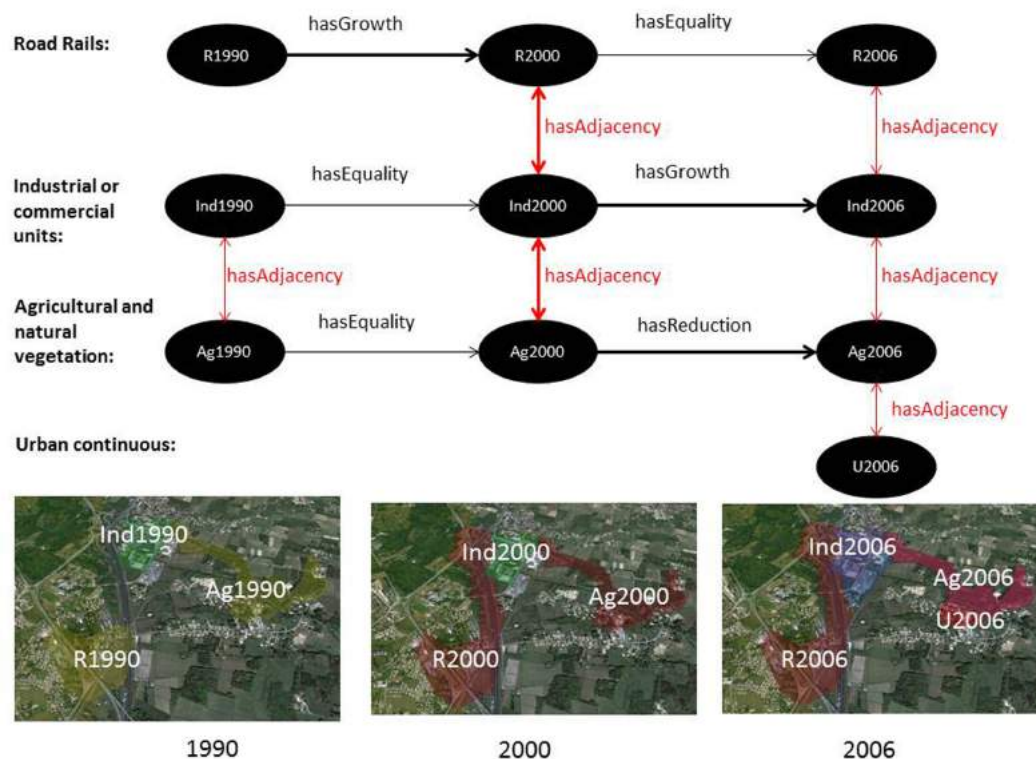


FIGURE 6.15 – Détection des évolutions connexes sur le développement des parcelles voisines

parcelle agricole (ou un espace naturel), cette dernière a subi les conséquences du processus d'urbanisation et a vu son territoire se réduire. Pendant ce temps, le territoire agricole perdu est remplacé par une parcelle correspondant à du tissu urbain continu.

La requête suivante a été utilisée afin de détecter les effets de l'expansion d'une route sur le voisinage. La requête recherche des motifs de croissance ou de réduction aux environs de la route :

Listing 6.2 – Requête SPARQL pour détecter les expansions et réduction aux environs d'une route nouvellement construite

```

SELECT ?tsNeighbour ?LandCover
WHERE
{
  ?ts rdf:type checksem:Road_Rails .
  ?ts checksem:hasGrowth ?ts2 .
  ?ts2 checksem:hasAdjacency/(checksem:hasGrowth|checksem:hasReduction) ?tsNeighbour .
  ?tsNeighbour rdf:type ?LandCover .
}

```

Cet exemple montre les capacités du modèle Continuum à capturer la dynamique d'un environnement. De plus, la modélisation sous forme de graphe, et les capacités d'interrogation du graphe, permettent d'identifier où se trouve le changement d'un point de vue spatio-temporel. Une bonne connaissance de l'expert dans les évolutions possibles de l'environnement est nécessaire pour capturer les motifs d'évolutions connexes.

C'est pourquoi la définition de motifs, relativement complexes, est possible si l'expert est assisté, ce qui permet à l'expert de suivre l'évolution et la dynamique d'un environnement. Toutefois, la définition de motifs complexes pourrait aboutir à la non-détection d'un phénomène. En conséquence, il serait très intéressant d'appliquer un processus de fouilles de graphes, voir de grands graphes, pour détecter des phénomènes complexes hors du champ de connaissance de l'expert, sur des interactions complexes et un volume de données conséquent. Cette thèse ouvre la voie à une nouvelle méthode d'étude et de suivis de dynamiques d'environnement, tout en laissant un champ nouveau d'investigations pour le domaine de recherches en fouilles de grands graphes.

Dans les exemples fournis, le moteur d'inférence n'a pas été utilisé, car la connaissance experte n'est malheureusement pas suffisamment riche par rapport à la capacité de conceptualisation d'une ontologie. C'est pourquoi cette étude s'attarde sur les aspects structurels des requêtes sur le graphe. Toutefois, une connaissance plus riche du terrain, notamment dans la définition intrinsèque des propriétés des objets, telles que la densité de construction sur une parcelle urbaine, permettrait d'exploiter davantage les capacités de raisonnement. L'annexe D illustre plusieurs exemples d'inférence basés sur des cas d'application spatiaux ou spatio-temporel.

Il semblerait que les experts du domaine, après de nombreuses discussions, ne modélisent pas la connaissance implicite. Toutefois, les récentes collaborations entre OGC et W3C augurent l'instauration de nouveaux procédés pour enrichir les données exploitables par le modèle Continuum.

EXPÉRIMENTATION

Ce chapitre présente l'implémentation ainsi que les résultats chiffrés de la mise en oeuvre de la solution avec le jeu de données Corine Land Cover. Dans un premier temps, les données doivent être traitées afin d'être converties en timeslices pour permettre l'intégration dans le modèle Continuum. Corine Land Cover ne fournit que des couches temporelles, c'est pourquoi il est nécessaire d'établir des liens entre ces couches grâce au calcul des transitions spatio-temporelles. A partir de ces transitions, il est possible de calculer les relations de filiation utilisées sur les différentes couches du modèle. Enfin, le chapitre se conclut en présentant le prototype permettant à des utilisateurs d'interroger et visualiser les données en se basant sur le modèle Continuum.

Sommaire

7.1	Intégration du jeu de données Corine Land Cover	131
7.2	Création des transitions spatio-temporelles	133
7.3	Calcul des couches du modèle	133
7.4	Architecture de l'application	136

7.1/ INTÉGRATION DU JEU DE DONNÉES CORINE LAND COVER

Dans le cadre de ces travaux, l'utilisation du jeu de données Corine Land Cover est au centre des expérimentations. D'une part, il est l'un des rares jeux de données mis à disposition gratuitement, proposant plusieurs couches temporelles. D'autre part, il possède la particularité de fournir une hiérarchie pour classifier les différentes couvertures terrestres, et pouvant être exploitée par notre modèle.

Les données sont fournies sous la forme de shapefiles¹. Afin de répondre aux besoins du modèle Continuum, il est nécessaire de convertir ces données sous la forme de timeslices. Les timeslices nécessitent de décrire une identité, une géométrie et une estampille temporelle. Des données thématiques peuvent également être intégrées en sus, mais elles ne sont pas obligatoires pour satisfaire les besoins du modèle. Dans le cas de Corine Land Cover, il n'existe pas de données thématiques. Seules sont définies les couvertures terrestres, les géométries et les estampilles temporelles.

Pour permettre la conversion des données, la hiérarchie de Corine Land Cover a été reproduite sous forme de classes au sein de l'ontologie du modèle. Puis pour chaque

1. <http://wiki.openstreetmap.org/wiki/Shapefiles>

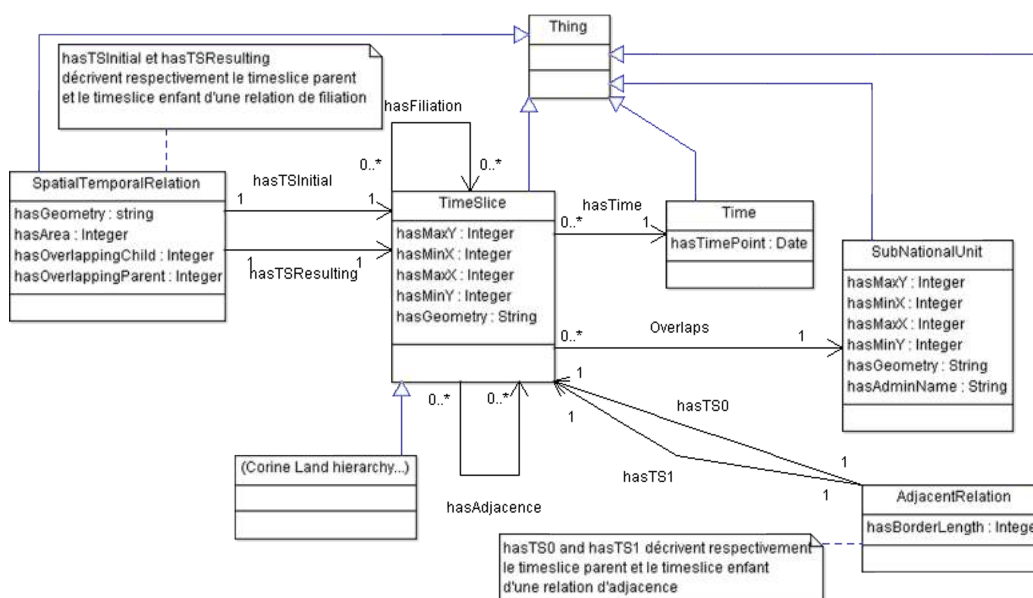


FIGURE 7.1 – Diagramme UML de l'application

parcelle contenue dans les shapfiles, un timeslice est créé comme une instance de la classe correspondant à la couverture de la parcelle. La géométrie est convertie au format Well Know Text (WKT) et stockée sous forme de chaîne de caractères via la propriété *hasGeometry*. Enfin, l'instant d'existence de la parcelle est également stocké via la propriété *hasTime*. La figure 7.1 présente le diagramme UML de l'application.

Le tableau 7.1 illustre les temps de calcul pour la conversion des shapfiles en timeslices. Les chiffres sont fournis sur le territoire de la France aux instants 1990, 2000 et 2006, soit un total de 841318 timeslices.

TABLE 7.1 – Temps de conversion des shapfiles en timeslices

	1990	2000	2006
Conversion des données en timeslices	57,099 sec	45,974 sec	44,302

Les timeslices sont alors encodés dans des fichiers en notation n3². Ensuite, ils sont chargés dans le triplestore. le tableau 7.2 présente le nombre de triplets encodés et chargés ainsi que le temps d'exécution du traitement.

TABLE 7.2 – Insertion des données dans le triplestore

	1990	2000	2006
Taille des données (nb triplets)	1.891.507	1.893.257	1.911.457
Temps d'insertion (en minutes)	2,01	2,05	2,09

Pour l'exemple présenté dans cette thèse, l'étude a été restreinte au territoire de la Gironde. Pour cela, un fichier shp contenant les géométries des départements français a

2. <http://www.w3.org/DesignIssues/Notation3.html>

été intégré dans le modèle via la classe *SubNationalUnit* (voir figure 7.1), puis les timeslices contenus dans chacun des départements ont été sélectionnés à l'aide d'un calcul spatial impliquant l'opérateur "overlaps". Cette étape a été réalisée à l'aide de la librairie SIG Geotools³ et a nécessité 47 minutes de traitement. Le tableau 7.3 présente le nombre de timeslices récupérés pour le département de la Gironde après cette phase de calcul.

TABLE 7.3 – Nombre de timeslices en Gironde

	1990	2000	2006
Nb timeslices	2807	2974	2953

7.2/ CRÉATION DES TRANSITIONS SPATIO-TEMPORELLES

Une fois les timeslices de la Gironde intégrés dans le triplestore, il est nécessaire de calculer les transitions spatio-temporelles en calculant tous les timeslices se chevauchant entre 1990 et 2000 et entre 2000 et 2006. La transition spatio-temporelle relie un timeslice enfant via la propriété *hasTSInitial* à un timeslice parent via la propriété *hasTSResulting* (voir figure 7.1). En outre, le taux de chevauchement du timeslice enfant et du timeslice parent est également calculé et stocké sous forme d'une valeur comprise entre 0 et 100 via les propriétés *hasOverlappingChild* et *hasOverlappingParent* (voir figure 7.1). Le tableau 7.4 présente le nombre de transitions spatio-temporelles calculées lors du traitement. Cette phase de calcul a nécessité 11,3 minutes de traitement.

TABLE 7.4 – Nombre de transitions spatio-temporelles

	1990/2000	2000/2006
Nb transitions	31 715	20 340

Les transitions spatio-temporelles ne détectent que des relations de chevauchement entre les timeslices à des instants consécutifs, qui sont assimilées à des filiations spatiales dans le modèle Continuum. Cependant, toutes les filiations spatiales ne sont pas valides au sein de la réalité modélisée. En d'autres termes, une filiation spatiale ne révèle pas obligatoirement de lien réel entre les timeslices concernés, car il existe une incertitude au sein du jeu de données, qui induit plusieurs chevauchements mineurs entre les timeslices des différentes couches. Un lien réel entre des timeslices est assimilé à une relation de filiation. Contrairement à la filiation spatiale, une relation de filiation suppose qu'il existe une causalité entre le timeslice parent et le timeslice enfant. Dès lors, il est nécessaire d'effectuer un filtrage des filiations spatiales afin de ne conserver que des filiations valides.

7.3/ CALCUL DES COUCHES DU MODÈLE

Ces filiations sont ensuite décrites au sein des quatre couches du modèle Continuum afin de rendre compte des différents aspects de la causalité entre les timeslices. Lors des

3. <http://geotools.org/>

précédents chapitres, un calcul a été présenté pour chacune des couches du modèle en ajoutant progressivement des contraintes depuis la couche 1 vers la couche 3. En réalité, il n'est pas nécessaire d'effectuer un calcul progressif de chacune des couches. En effet, le modèle Continuum se base sur une ontologie et permet de définir une relation de subsomption, non seulement sur les concepts, mais également sur les propriétés et relations. Le modèle Continuum tire avantage de cette particularité en hiérarchisant les relations définies dans chacune des couches du modèle grâce à un mécanisme de subsomption. La figure 7.2 illustre cette subsomption au sein du modèle.

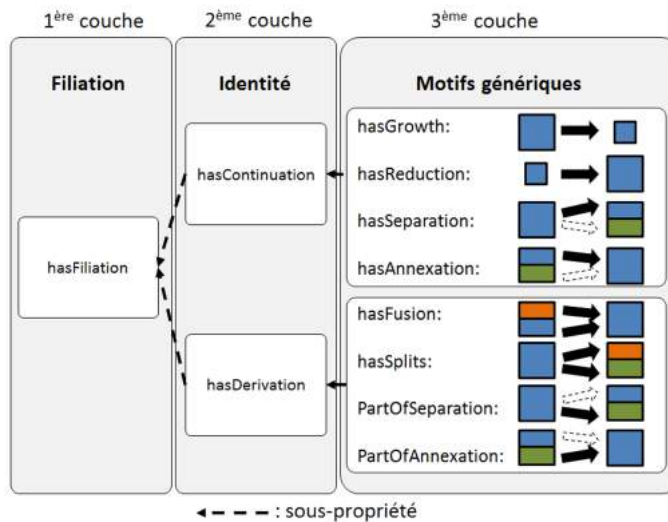


FIGURE 7.2 – La subsomption des couches du modèle Continuum

La figure 7.3 reprend un exemple de Corine Land Cover pour illustrer de façon plus concrète les différentes visions offertes par les couches du modèle. Les couches 1 et 2 peuvent être interrogées à l'aide d'inférences en se basant sur la relation de subsomption.

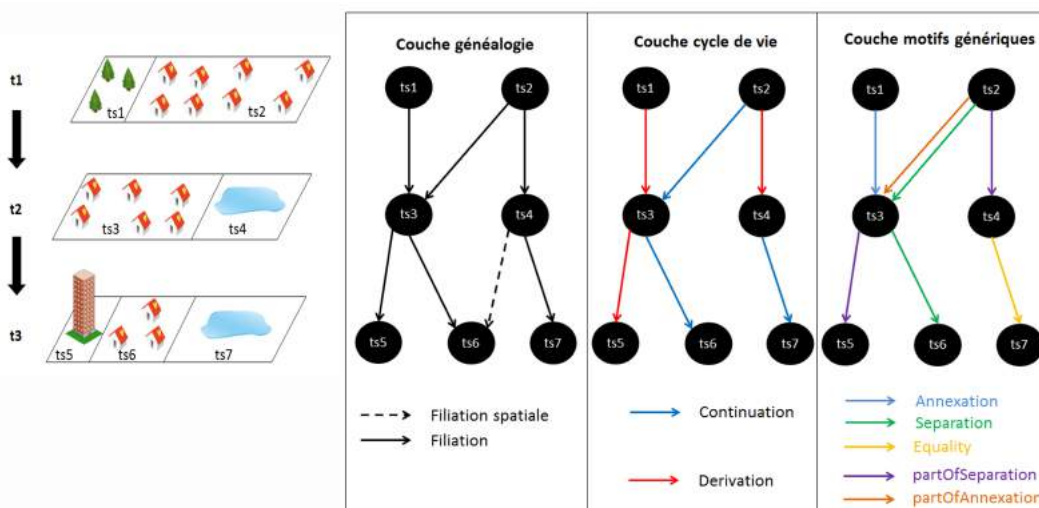


FIGURE 7.3 – Représentation de l'évolution au travers des trois couches du modèle

Dès lors, le calcul des relations de la 3ème couche suffit pour permettre l'exploitation du modèle sur toutes les couches. Il est intéressant de noter que la couche 4 peut être vue comme une spécialisation des relations de la couche 3. Toutefois, cette couche décrit des phénomènes ou motifs dépendants du domaine d'application. Son calcul ne peut donc pas être généralisé, c'est pourquoi l'attention est portée sur la couche 3 dont les motifs sont supposés génériques. La requête suivante montre comment calculer la relation d'expansion définie dans la couche 3 directement à partir d'une transition spatio-temporelle. Les seuils ρ et χ , permettant le filtrage des filiations spatiales, ont été fixés à 10. Le seuil φ , permettant de garantir un taux minimum de croissance afin d'éviter la prise en compte de changement négligeable pour le motif d'expansion, a également été fixé à 10.

Listing 7.1 – Requête SPARQL pour détecter les relations d'expansion à partir de transitions spatio-temporelles

```

insert
{
  ?p checksem:hasGrowth ?c
}
where
{
  ?st a checksem:SpatiotemporalTransition .
  ?st checksem:hasParentTimeSlice ?p.
  ?st checksem:hasChildTimeSlice ?c.
  ?p a ?classP .
  ?c a ?classC .
  ?p checksem:hasOverlappingParent ?Op.
  ?c checksem:hasOverlappingChild ?Oc.
  FILTER(?classP=?classC && ?Op>10 && ?Op=100 && ?Op>?Oc && (?Op-?Oc)>10)
}

```

Les requêtes similaires ont été définies pour calculer toutes les relations de la couche 3. Le tableau 7.5 présente le temps d'exécution pour chacune des requêtes ainsi que le nombre de motifs détectés sur le département de la Gironde.

TABLE 7.5 – Nombre de motifs détectés sur le département de la Gironde et temps d'exécution de la requête

	Temps de calcul (en ms)	Nombre de motifs détectés
hasGrowth	942	395
hasReduction	869	258
hasEquality	1053	6867
hasSeparation	2156	22
hasAnnexation	2458	24
hasConversion	967	103
hasFusion	201	0
hasSplit	1026	7
PartOfSeparation	2056	28
PartOfAnnexation	2198	31

Plusieurs détails nécessitent de plus amples explications afin de comprendre les résultats présentés. Tout d'abord, le tableau 7.5 révèle un nombre largement supérieur de relations *hasEquality*. Cette relation illustre toutes les parcelles du territoire qui n'ont pas varié dans le temps, c'est pourquoi il semble logique que la majeure partie du territoire soit restée à l'identique. Il est également intéressant de constater que le nombre de relations

hasSeparation et *PartOfSeparation*, représentant le même motif n'est pas exactement identique. Ceci s'explique par le fait que les motifs de séparation peuvent se séparer en deux entités ou plus. Ainsi le nombre de continuations et de dérivations impliquées dans le motif n'est pas obligatoirement identique. Le même principe s'applique pour les motifs d'annexion.

Les requêtes, permettant de calculer les motifs de la troisième couche nécessite, normalement de faire appel à des opérateurs spatiaux, qui allongent considérablement le temps de calcul, comme nous avons pu le constater pour le calcul des transitions spatio-temporelles. Grâce à la méthodologie proposée dans ces travaux, les calculs spatiaux ont été substitués par de simples calculs numériques à l'aide des taux d'intersection portés par les propriétés *hasOverlappingChild* et *hasOverlappingParent*. Cette méthodologie diminue considérablement le temps de calcul des relations et explique les faibles temps d'exécution des requêtes.

Pour permettre l'exploitation du modèle et la découverte de connaissance (voir chapitre 6), toutes les relations calculées ont été stockées explicitement dans le triplestore et ne nécessitent plus aucun calcul d'aucune sorte pour être interrogées. Afin de permettre l'interrogation et la visualisation des informations par des utilisateurs, le modèle Continuum a été intégré au sein d'une architecture. Cette architecture sera présentée à la prochaine section.

7.4/ ARCHITECTURE DE L'APPLICATION

Afin de fournir un système exploitable par l'utilisateur, le modèle est intégré au sein d'une architecture développée à l'aide de JAVA/J2EE⁴. La figure 7.4 illustre le fonctionnement de l'architecture. L'objectif de l'application est de permettre à l'utilisateur de fournir une requête au système et d'obtenir un résultat visuel dans un navigateur web.

Les données de l'application sont stockées dans le triplestore Stardog⁵. Les résultats de la requête de l'utilisateur sont encodées dans un fichier KML⁶, qui est un standard permettant l'affichage de données géographiques. Le traitement des données s'effectuent à l'aide du framework JAVA/J2EE. Enfin, l'affichage se fait à l'aide du langage HTML ainsi que du plug-in Google Earth⁷, qui permet l'affichage des fichiers KML dans un navigateur web.

Lorsque l'utilisateur formule une requête :

1. la requête est envoyée sur le triplestore à l'aide du langage SPARQL.
2. le résultat est récupérée par le module de traitement.
3. le résultat est encodé dans un fichier KML à l'aide de la librairie Java NanoXML⁸.
4. le fichier KML est chargé dans le navigateur.

Lors de l'encodage du fichier KML, l'identité, la géométrie, l'estampille temporelle et les éventuelles données thématiques de chaque timeslice doivent être converties, de

4. <http://java-source.net/open-source/j2ee-frameworks>

5. <http://stardog.com/>

6. <https://developers.google.com/kml/>

7. <http://www.google.fr/earth/explore/products/plugin.html>

8. <http://nanoxml.sourceforge.net/>

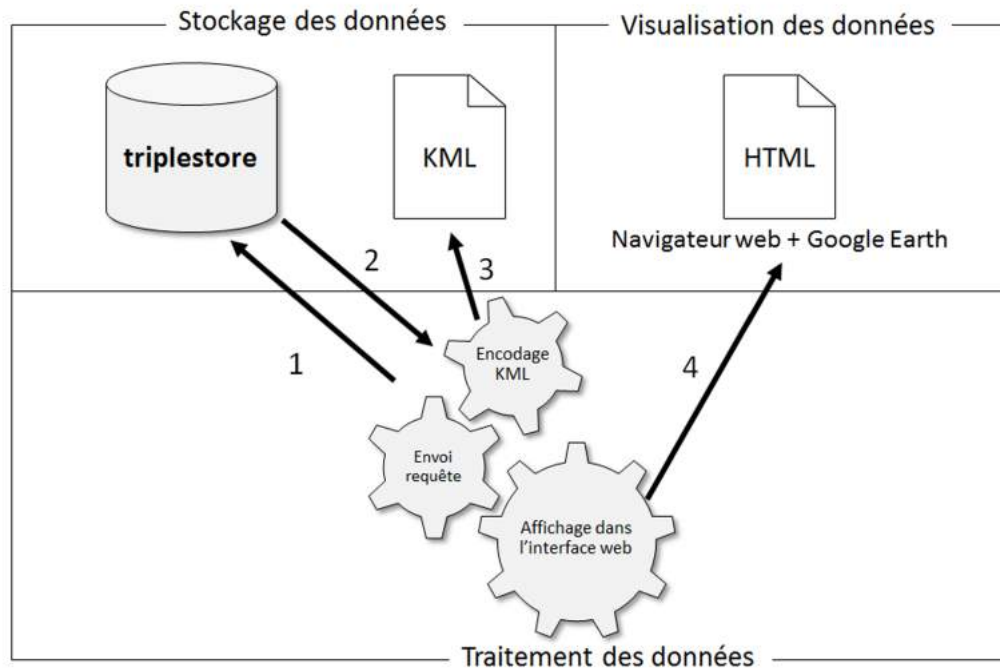


FIGURE 7.4 – Illustration de l'architecture de l'application

sorte que l'utilisateur puisse accéder à ces informations. Le code ci-dessous illustre la représentation d'un timeslice dans un fichier KML.

Listing 7.2 – Encodage d'un timeslice dans un fichier KML

```

<Placemark id="pmchecksem:timeslice_1990_FR-192322">
  <name>checksem:timeslice_1990_FR-192322</name>
  <description>
    <![CDATA[<table cellpadding="1" cellspacing="1">
  <tr>
  <td><b>Land Cover:</b></td>
  <td>checksem:Pastures</td>
  </tr>
  </table >]]><br/>
  </description>
  <TimeStamp>
    <when>1990</when>
  </TimeStamp>
  <styleUrl>#StyleMap_1990</styleUrl>
  <Polygon>
    <altitudeMode>clampToGround</altitudeMode>
    <outerBoundaryIs>
      <LinearRing>
        <coordinates>
          -0.47848072850975265,45.051385470304815
          -0.47672078393137396,45.05059082272803
          -0.47403557069107405,45.05111686379739
          ...
          -0.47870404165590896,45.051163771531876
          -0.47848072850975265,45.051385470304815
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</Placemark>

```

Chaque timeslice est décrit à l'aide d'une balise *Placemark* nécessaire pour afficher une entité à l'aide d'un fichier KML. L'identité et les éventuelles données thématiques demandées par l'utilisateur sont définies textuellement en format HTML dans la balise *description*. Le contenu de cette balise s'affiche lorsque l'utilisateur clique sur la géométrie du timeslice dans le navigateur.

Google Earth possède la particularité de fournir un outil pour visualiser des entités temporelles. A l'aide des balises *TimeStamp* et *when* pour décrire des instants de temps et à l'aide des balises *TimeSpan* et des balises *begin* et *end* pour décrire des intervalles de temps. Un curseur temporel est défini dans l'application et permet de n'afficher que les entités à un instant de temps précis (voir figure 7.5 B). Il est également possible de définir un intervalle de temps à l'aide du curseur et d'afficher toutes les entités valides sur cet intervalle de temps (voir figure 7.5 A).



FIGURE 7.5 – Illustration du curseur temporel

La balise *styleUrl* permet de définir un style pour l'entité. Dans ces travaux, une couleur différente est définie en fonction des estampilles temporelles (affichage jaune pour les entités valides en 1990, rouge pour les entités valides en 2000 et bleu pour les entités valides en 2006). La balise *Polygon* permet de décrire la vue par défaut de l'entité avec la balise *altitudeMode* et les coordonnées de la géométrie avec la balise *coordinates*. Pour cela, les données relatives à la géométrie, définies en WKT dans le triplestore, sont converties au format KML par le module de traitement. La figure 7.6 illustre l'affichage de ces balises et des différents styles dans Google Earth.

Une fois tous les timeslices encodés, le fichier est chargé dans l'interface web et offre à l'utilisateur une navigation interactive pour visualiser le résultat de la requête. La figure 7.7 présente l'interface de l'application développée dans le cadre de ces travaux.

Ce chapitre a présenté l'implémentation du modèle Continuum ainsi que les résultats de son expérimentation sur le jeu de données Corine Land Cover. Enfin, l'architecture basée sur le modèle Continuum a également été exposé pour mettre en avant la solution.



FIGURE 7.6 – Illustration des différents styles utilisés par l'application

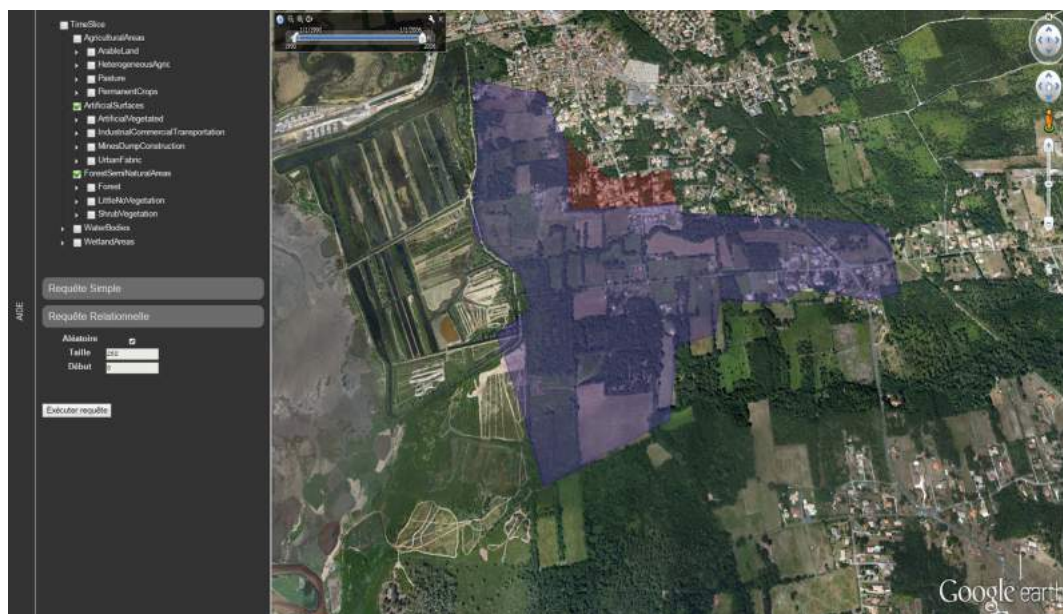


FIGURE 7.7 – Illustration de l'interface de l'application



CONCLUSION

CONCLUSION GÉNÉRALE

Ce document a présenté le modèle spatio-temporel nommé Continuum ayant pour objectif d'étudier la dynamique d'un environnement. Ce modèle a pour vocation d'assister les experts dans l'analyse du territoire, afin d'en comprendre les dynamiques et décider des futures orientations pour assurer la pérennité du territoire. Ce chapitre offre un bilan des travaux réalisés et ouvre des perspectives de développement du modèle Continuum.

Sommaire

8.1 Bilan	143
8.2 Perspectives	144

8.1/ BILAN

Afin de répondre à des problématiques non encore explorées des SIG spatio-temporels, le modèle Continuum a été proposé et implémenté pour répondre aux limites des solutions existantes. Celui-ci possède la particularité d'être basé sur une ontologie formelle à l'aide des technologies du Web Sémantique. De ce fait, la structure du modèle est, par nature, un graphe. Les graphes ont été identifiés comme des structures particulièrement adaptées aux besoins de la modélisation. En effet, l'étude de la dynamique d'un environnement nécessite de s'intéresser aux diverses interactions spatiales, temporelles et thématiques qu'entretiennent entre elles les entités d'un territoire. Le modèle s'intéresse essentiellement à la relation de filiation et définit quatre couches permettant chacune d'offrir une vision de la dynamique d'une entité. La première couche étudie la généalogie des entités au cours du temps. La deuxième couche étudie le cycle de vie primaires et dérivés des entités. La troisième couche décrit des motifs spatiaux génériques capables de représenter les dynamiques spatiales des entités d'un territoire. La quatrième couche, quant-à-elle, n'est pas générique, mais exploite les motifs de la troisième couche ainsi que les connaissances liées au domaine de l'application pour donner une interprétation de ces motifs et tenter d'en extraire des phénomènes du monde réel.

Dans ces travaux, une méthodologie a été proposée afin de distinguer les relations de filiation spatiales des relations de filiation valides. Les filiations spatiales font référence aux relations établies entre des parcelles dont les géométries se chevauchent dans le temps, ces relations peuvent être valides s'il est admis qu'il existe une filiation entre les parcelles dans le monde réel, ou bien non valides, dans le cas contraire. Toutefois, la distinction entre les deux types d'entités peut parfois s'avérer arbitraire et se rapporte au

problème de l'incertitude. Dans le cadre de cette thèse, un seuil est défini afin d'établir clairement la limite entre ces deux relations et sert de support au calcul des différentes couches du modèle.

Afin de détecter des phénomènes, au plus proche de la réalité modélisée, le modèle Continuum s'appuie sur une hiérarchie de concept permettant de regrouper les entités spatio-temporelles au sein de classes plus ou moins génériques. Dans ces travaux, la hiérarchie est exploitée afin de permettre d'étudier la profondeur du changement. Dès lors, un changement impliquant l'identité d'une entité et associé à la relation de dérivation dans le modèle, est vu comme un changement depuis une classe *parent* de la hiérarchie vers une classe *enfant* de la hiérarchie. Le modèle Continuum permet de hiérarchiser ce type de changement en étudiant le niveau maximum de la hiérarchie sur lequel il est possible de visualiser le changement. Ainsi, plus le changement intervient sur des concepts génériques et plus celui-ci est considéré comme fort, car plus des concepts sont génériques et plus leur sémantique est éloignée. A l'inverse, un changement, intervenant entre des concepts très spécifiques suggère une modification faible de la couverture terrestre.

La quatrième couche du modèle est la plus expressive et permet d'avoir une vision concrète des phénomènes dynamiques agissant sur le territoire. Toutefois, le résultat fourni au travers des différentes couches, n'est valable qu'entre deux instants de temps consécutifs. Il existe un besoin d'étudier la dynamique d'une entité dans la durée. Pour cela, le modèle Continuum propose des outils simples et efficaces de recherche de motifs, appelés motifs de tendance, au sein du graphe.

L'évolution d'une entité, qu'elle soit spontanée ou inscrite dans la durée, peut également être influencée par l'évolution des entités voisines. En effet, le territoire modélisé peut être vu comme un espace fixe et délimité, composé d'un ensemble de parcelles évoluant au cours du temps. Dès lors, on comprend aisément que l'évolution spatiale d'une entité implique une évolution conjointe des parcelles environnantes. L'ajout de sémantique ainsi que les outils de fouille de graphe proposés par le modèle Continuum, permettent d'assister les experts dans la découverte des phénomènes connexes au sein du territoire.

L'ensemble des travaux présentés dans cette thèse ont donné lieu à plusieurs publications dont la liste est donnée dans l'annexe A.

8.2/ PERSPECTIVES

Les travaux réalisés durant de cette thèse ouvrent la voie à un nombre important de perspectives. Le modèle présenté dans cette thèse opère sur des entités dynamiques dont la composante spatiale est définie à l'aide de polygones uniquement. Si cette représentation simplifie la définition du jeu de données ainsi que la complexité du modèle proposé, elle dénote cependant d'un écart important entre la réalité et sa représentation. Il est possible d'identifier quatre axes majeurs de développement.

1) Tout d'abord, au sein du jeu de données Corine Land Cover, les réseaux routiers sont définis à l'aide de polygones approximatifs plutôt que d'utiliser des géométries, telles que les lignes ou les polygones. La principale difficulté, induite par l'intégration de nouveaux types de géométries dans le modèle, concerne la définition de nouveaux motifs génériques plus complexes dans la troisième couche. La coexistence entre des poly-

gones et des lignes au sein du jeu de données, implique généralement la superposition d'entités. Par exemple, le réseau routier est fréquemment superposé aux couvertures terrestres, ce qui induit de nouvelles contraintes spatiales à prendre en compte. Cependant le modèle DE-9IM définit, d'ores et déjà, un ensemble de relations spatiales entre des types de géométrie différents pouvant servir de support à la définition des motifs génériques. Malgré la complexité accrue de la modélisation, l'intégration de nouveaux types de données spatiales ouvre la voie à la gestion de nouveaux types d'entités au sein d'un même environnement géospatial. Par exemple, un tel modèle permettrait de gérer l'évolution des infrastructures de transports indépendamment des parcelles de terrains tout en assurant la corrélation entre les deux types d'entités afin d'en comprendre les interactions.

2) Une autre perspective intéressante concerne l'intégration de données hétérogènes au sein du modèle. En effet, il n'existe actuellement aucun jeu de données formaté pour le modèle Continuum. Par conséquent, il apparaît indispensable de fournir un outil de type ETL (extraction, transformation, chargement) capable de convertir des jeux de données standard en timeslices.

3) Le modèle Continuum possède la particularité d'utiliser la relation de filiation ainsi qu'une hiérarchie pour classifier les entités du modèle. Ainsi, bien qu'une méthodologie ait été définie dans cette thèse pour calculer les relations de filiation sur un jeu de données incomplet, il existe un besoin pour de nouvelles approches pour traiter l'incertitude afin d'obtenir un jeu de données toujours plus proche de la réalité à modéliser. La hiérarchie, quand-à-elle, n'est que rarement présente au sein des jeux de données. Cependant, le modèle Continuum est basé sur une ontologie dont l'une des caractéristiques majeures consiste à être réutilisable. Ainsi, il est possible d'envisager de greffer une hiérarchie déjà existante pour compléter les jeux de données ne disposant que d'une liste de termes pour classifier les entités.

4) Enfin, la dernière perspective envisagée concerne la gestion de l'imprécision [Lukasiewicz et al., 2008]. En effet, la particularité majeure du modèle Continuum consiste à interpréter les motifs spatiaux de la troisième couche, à l'aide d'informations contextuelles de sorte à pouvoir qualifier le phénomène sous-jacent. Par exemple, la réduction spatiale d'une entité, dont la couverture est une forêt, est interprétée comme un phénomène de déforestation. Dès lors, il est intéressant de noter que l'interprétation est imprécise car le terme de *déforestation* peut paraître excessif dans le cas de réduction minimale de la taille de la forêt. Dès lors, l'interprétation devient une affirmation avec un certain degré de confiance, car il n'est pas possible d'établir que l'affirmation est complètement fautive, ni complètement vraie à cause de concepts vagues tels que la déforestation dont la définition est imprécise. Ainsi, il serait plus correct de définir des informations de la forme : " la forêt *forest01* a subi une déforestation entre un instant T1 et un instant T2 avec un degré de confiance de 0.2".

BIBLIOGRAPHIE

- [Ackoff, 1989] Ackoff, R. L. (1989). **From data to wisdom : Presidential address to isgsr, june 1988**. *Journal of applied systems analysis*, 16(1) :3–9.
- [Allen, 1983] Allen, J. F. (1983). **Maintaining knowledge about temporal intervals**. *Communications of the ACM*, 26(11) :832–843.
- [André, 2011] André, C. (2011). **Modèles de temps et de contraintes temporelles de marte et leurs applications**.
- [Arenas et al.,] Arenas, H., Harbelot, B., et Cruz, C. **A semantic analysis of moving objects, using as a case study maritime voyages from eighteenth and nineteenth centuries**. Dans *the Sixth International Conference on Advanced Geographic Information Systems, Applications, and Services*.
- [Arenas et al., 2013] Arenas, H., Harbelot, B., et Cruz, C. (2013). **A semantic web approach for geodata discovery**. Dans *7th International Workshop on Semantic and Conceptual Issues in GIS (SeCoGIS 2013)*.
- [Arenas et al., 2015] Arenas, H., Harbelot, B., et Cruz, C. (2015). **Reasoning with vague spatial information from upper mesopotamia (2000bc)**. *Procedia Environmental Sciences*, 27 :58–65.
- [Armstrong, 1988] Armstrong, M. P. (1988). **Temporality in spatial databases**. Dans *Proceedings : GIS/LIS*, volume 88, pages 880–889.
- [Artale et al., 1998] Artale, A., et Franconi, E. (1998). **A Temporal Description Logic for Reasoning About Actions and Plans**. *Journal of Artificial Intelligence Research*, 9(1) :463–506.
- [Baader et al., 2006] Baader, F., Lutz, C., et Suntisrivaraporn, B. (2006). **Cel a polynomial-time reasoner for life science ontologies**. Dans *Automated Reasoning*, pages 287–291. Springer.
- [Baader et al., 2003] Baader, F., et Nutt, W. (2003). **Basic description logics**. *Description logic handbook*, pages 43–95.
- [Beller et al., 1991] Beller, A., Giblin, T., Le, K. V., Litz, S., Kittel, T., et Schimel, D. (1991). **Temporal gis prototype for global change research**. *GIS/LIS'91, Atlanta, GA, USA, 10/28-11/01/91*, pages 752–765.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., Lassila, O., et others (2001). **The semantic web**. *Scientific american*, 284(5) :28–37.
- [Bossard et al., 2000] Bossard, M., Feranec, J., et Otahel, J. (2000). **Corine land cover technical guide : Addendum 2000**.
- [Brickley, 2003] Brickley, D. (2003). **Basic geo vocabulary**. *W3C Semantic Web Interest Group*. Available at : <http://www.w3.org/2003/01/geo>.
- [Caldas et al., 2015] Caldas, M. M., Goodin, D., Sherwood, S., Campos Krauer, J. M., et Wisely, S. M. (2015). **Land-cover change in the paraguayan chaco : 2000–2011**. *Journal of Land Use Science*, 10(1) :1–18.

- [Cheylan, 2003] Cheylan, J.-P. (2003). **Time and spatial database, a conceptual application framework**. *Life and Motion of Socio-economic Units, GISDATA Series, Taylor and Francis, New York/London*, (8) :35–48.
- [Claramunt et al., 1996] Claramunt, C., et Theriault, M. (1996). **Toward semantics for modelling spatio-temporal processes within gis**. *Advances in Gis Research I*, pages 27–43.
- [Cleveland, 1982] Cleveland, H. (1982). **Information as a resource**. *Futurist*, 16(6) :34–39.
- [Clifford et al., 1992] Clifford, J., et Isakowitz, T. (1992). **On the semantics of transaction time and valid time in bitemporal databases**. *Information Systems Working Papers Series, Vol.*
- [Cox et al., 2002] Cox, S., Cuthbert, A., Daisey, P., Davidson, J., Johnson, S., Keighan, E., Lake, R., Mabrouk, M., Margoulies, S., Martell, R., et others (2002). **Opengis® geography markup language (gml) implementation specification, version**.
- [Del Mondo et al., 2013] Del Mondo, G., Rodríguez, M., Claramunt, C., Bravo, L., et Thi-
baud, R. (2013). **Modeling consistency of spatio-temporal graphs**. *Data & Knowledge Engineering*, 84 :59–80.
- [Egenhofer, 2002] Egenhofer, M. J. (2002). **Toward the semantic geospatial web**. Dans *Proceedings of the 10th ACM international symposium on Advances in geographic information systems*, pages 1–4. ACM.
- [Egenhofer et al., 1992] Egenhofer, M. J., et Al-Taha, K. K. (1992). **Reasoning about gradual changes of topological relationships**. *Theories and methods of spatio-temporal reasoning in geographic space*, pages 196–219.
- [ESRI, 1998] ESRI, E. (1998). **Shapefile technical description**. *An ESRI White Paper*.
- [Frank, 2003] Frank, A. U. (2003). **Ontology for spatio-temporal databases**. Springer.
- [Frasincar et al., 2010] Frasincar, F., Milea, V., et Kaymak, U. (2010). **towl : Integrating time in owl**. *Semantic Web Information Management*, pages 225–246.
- [Gilchrist, 2003] Gilchrist, A. (2003). **Thesauri, taxonomies and ontologies-an etymological note**. *Journal of documentation*, 59(1) :7–18.
- [Grosz et al., 2003] Grosz, B. N., Horrocks, I., Volz, R., et Decker, S. (2003). **Description logic programs : Combining logic programs with description logic**. Dans *Proceedings of the 12th international conference on World Wide Web*, pages 48–57. ACM.
- [Gruber, 1995] Gruber, T. R. (1995). **Toward principles for the design of ontologies used for knowledge sharing ?** *International journal of human-computer studies*, 43(5) :907–928.
- [Gutierrez et al., 2007] Gutierrez, C., Hurtado, C. A., et Vaisman, A. (2007). **Introducing time into rdf**. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2) :207–218.
- [Harbelot et al., 2013a] Harbelot, B., Arenas, H., et Cruz, C. (2013a). **Continuum : A spatiotemporal data model to represent and qualify filiation relationships**. Dans *4th ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS) 2013*.
- [Harbelot et al., 2013b] Harbelot, B., Arenas, H., et Cruz, C. (2013b). **Semantics for spatio-temporal'smart queries'**. Dans *9th International Conference on Web Information Systems and Technologies (WEBIST)*.

- [Harbelot et al., 2013c] Harbelot, B., Arenas, H., et Cruz, C. (2013c). **The spatio-temporal semantics from a perdurantism perspective**. Dans *In Proceedings of the Fifth International Conference on Advanced Geographic Information Systems, Applications, and Services GEOProcessing*.
- [Harbelot et al., 2013d] Harbelot, B., Arenas, H., et Cruz, C. (2013d). **Using semantic web technologies to follow the evolution of entities in time and space**. *International Journal On Advances in Intelligent Systems*.
- [Harbelot et al., 2014a] Harbelot, B., Arenas, H., et Cruz, C. (2014a). **A semantic model to query spatial-temporal data**. *Information Fusion and Geographic Information Systems (IF AND GIS 2013)*, pages 75–89.
- [Harbelot et al., 2014b] Harbelot, B., Arenas, H., et Cruz, C. (2014b). **A semantic model to query spatial temporal data**. Dans Popovich, V., Claramunt, C., Schrenk, M., et Korolenko, K., éditeurs, *Information Fusion and Geographic Information Systems (IF and GIS 2013)*, Lecture Notes in Geoinformation and Cartography, pages 75–89. Springer Berlin Heidelberg.
- [Harbelot et al., 2014c] Harbelot, B., Arenas, H., et Cruz, C. (2014c). **Un modèle sémantique spatio-temporel pour capturer la dynamique des environnements**. *Fouille de données spatiales et temporelles*.
- [Harbelot et al., 2015a] Harbelot, B., Arenas, H., et Cruz, C. (2015a). **Lc3 : un modèle spatial et sémantique pour découvrir la connaissance dans les jeux de données**. *Atelier Gestion et Analyse des données Spatiales et Temporelles*.
- [Harbelot et al., 2015b] Harbelot, B., Arenas, H., et Cruz, C. (2015b). **Lc3 : A spatio-temporal and semantic model for a knowledge discovery from geospatial datasets**. *Journal of Web Semantics*.
- [Heller, 1990] Heller, M. (1990). **The ontology of physical objects : Four-dimensional hunks of matter**. Cambridge University Press.
- [Hobbs et al., 2004] Hobbs, J. R., et Pan, F. (2004). **An ontology of time for the semantic web**. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1) :66–85.
- [Hornsby et al., 2000] Hornsby, K., et Egenhofer, M. J. (2000). **Identity-based change : a foundation for spatio-temporal knowledge representation**. *International Journal of Geographical Information Science*, 14(3) :207–224.
- [Horrocks et al., 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosz, B., Dean, M., et others (2004). **Swrl : A semantic web rule language combining owl and ruleml**. *W3C Member submission*, 21 :79.
- [Hustadt et al., 1994] Hustadt, U., et others (1994). **Do we need the closed world assumption in knowledge representation ?** Dans *KRDB*.
- [Jensen et al., 1992] Jensen, C. S., Clifford, J., Gadia, S. K., Segev, A., et Snodgrass, R. T. (1992). **A glossary of temporal database concepts**. *ACM Sigmod Record*, 21(3) :35–43.
- [Jensen et al., 1993] Jensen, C. S., Soo, M. D., et Snodgrass, R. T. (1993). **Unification of temporal data models**. Dans *Data Engineering, 1993. Proceedings. Ninth International Conference on*, pages 262–271. IEEE.
- [Jiang et al., 2009] Jiang, J., et Worboys, M. (2009). **Event-based topology for dynamic planar areal objects**. *International Journal of Geographical Information Science*, 23(1) :33–60.

- [Kazakov et al., 2009] Kazakov, Y., et others (2009). **Consequence-driven reasoning for horn shiq ontologies**. Dans *IJCAI*, volume 9, pages 2040–2045.
- [Kifer et al., 2010] Kifer, M., et Boley, H. (2010). **Rif overview**. *W3C Working Group Note*.
- [Klein et al., 2001] Klein, M. C., et Fensel, D. (2001). **Ontology versioning on the semantic web**. Dans *SWWS*, pages 75–91.
- [Knublauch et al., 2009] Knublauch, H., Hendler, J., et Idehen, K. (2009). **Spin-sparql inferencing notation**.
- [LABOREE12 et al., 2012] LABOREE12, V., HISSEL, F., et SERGENT, P. (2012). **Evolution de l'emprise des zones inondables de l'estuaire de la gironde sous l'influence du changement climatique**.
- [Langran et al., 1988] Langran, G., et Chrisman, N. R. (1988). **A framework for temporal geographic information**. *Cartographica : The International Journal for Geographic Information and Geovisualization*, 25(3) :1–14.
- [Laroche et al., 2006] Laroche, B., Thorette, J., et Lacassin, J.-C. (2006). **L'artificialisation des sols : pressions urbaines et inventaire des sols**. *Étude et gestion des sols*, 13(3) :223–236.
- [Levesque, 1984] Levesque, H. J. (1984). **The logic of incomplete knowledge bases**. Dans *On conceptual modelling*, pages 165–189. Springer.
- [Lieberman et al., 2007] Lieberman, J., Singh, R., et Goad, C. (2007). **W3c geospatial ontologies**. *Incubator group report, W3C*.
- [Lukasiewicz et al., 2008] Lukasiewicz, T., et Straccia, U. (2008). **Managing uncertainty and vagueness in description logics for the semantic web**. *Web Semantics : Science, Services and Agents on the World Wide Web*, 6(4) :291–308.
- [Mas, 1999] Mas, J.-F. (1999). **Monitoring land-cover changes : a comparison of change detection techniques**. *International journal of remote sensing*, 20(1) :139–152.
- [McGuinness et al., 2004] McGuinness, D. L., Van Harmelen, F., et others (2004). **Owl web ontology language overview**. *W3C recommendation*, 10(10) :2004.
- [Milea et al., 2008] Milea, V., Frasincar, F., et Kaymak, U. (2008). **Knowledge engineering in a temporal semantic web context**. Dans *Web Engineering, 2008. ICWE'08. Eighth International Conference on*, pages 65–74. IEEE.
- [Milea et al., 2012] Milea, V., Frasincar, F., et Kaymak, U. (2012). **towl : A temporal web ontology language**. *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, 42(1) :268–281.
- [Motik et al., 2009] Motik, B., Horrocks, I., et Sattler, U. (2009). **Bridging the gap between owl and relational databases**. *Web Semantics : Science, Services and Agents on the World Wide Web*, 7(2) :74–89.
- [Motro, 1989] Motro, A. (1989). **Integrity= validity+ completeness**. *ACM Transactions on Database Systems (TODS)*, 14(4) :480–502.
- [Noy et al., 2006] Noy, N., Rector, A., Hayes, P., et Welty, C. (2006). **Defining n-ary relations on the semantic web**. *W3C Working Group Note*, 12 :4.
- [O'Connor et al., 2011] O'Connor, M. J., et Das, A. K. (2011). **A method for representing and querying temporal information in owl**. Dans *Biomedical engineering systems and technologies*, pages 97–110. Springer.

- [Parent et al., 2013] Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., et others (2013). **Semantic trajectories modeling and analysis**. *ACM Computing Surveys (CSUR)*, 45(4) :42.
- [Parsia, 2014] Parsia, C. (2014). **Stardog website**. <http://stardog.com/>.
- [Pelekis et al., 2004] Pelekis, N., Theodoulidis, B., Kopanakis, I., et Theodoridis, Y. (2004). **Literature review of spatio-temporal database models**. *The Knowledge Engineering Review*, 19(03) :235–274.
- [Perry et al., 2011] Perry, M., Jain, P., et Sheth, A. P. (2011). **Sparql-st : Extending sparql to support spatiotemporal queries**. Dans *Geospatial semantics and the semantic web*, pages 61–86. Springer.
- [Peuquet, 1999] Peuquet, D. (1999). **Time in gis and geographical databases**. *Geographical information systems*, 1 :91–102.
- [Peuquet et al., 1995] Peuquet, D. J., et Duan, N. (1995). **An event-based spatiotemporal data model (estdm) for temporal analysis of geographical data**. *International journal of geographical information systems*, 9(1) :7–24.
- [Pointereau et al., 2009] Pointereau, P., et Coulon, F. (2009). **Abandon et artificialisation des terres agricoles**. *Courrier de l'environnement de l'INRA*, 57 :109–120.
- [Prud'Hommeaux et al., 2008] Prud'Hommeaux, E., Seaborne, A., et others (2008). **Sparql query language for rdf**. *W3C recommendation*, 15.
- [Ramankutty et al., 1999] Ramankutty, N., et Foley, J. A. (1999). **Estimating historical changes in global land cover : Croplands from 1700 to 1992**. *Global biogeochemical cycles*, 13(4) :997–1027.
- [Randell et al., 1992] Randell, D. A., Cui, Z., et Cohn, A. G. (1992). **A spatial logic based on regions and connection**. *KR*, 92 :165–176.
- [Reed et al., 2006] Reed, C., Singh, R., Lake, R., Lieberman, J., et Maron, M. (2006). **An introduction to georss : A standards based approach for geo-enabling rss feeds**. *White Paper OGC*.
- [Rowley, 2007] Rowley, J. E. (2007). **The wisdom hierarchy : representations of the dikw hierarchy**. *Journal of information science*.
- [Salomon, 2002] Salomon, J.-N. (2002). **L'inondation dans la basse vallée de la garonne et l'estuaire de la gironde lors de la " tempête du siècle"(27-28 décembre 1999)/flooding in the garonne valley and the gironde estuary caused by the " storm of the century"(27-28 december 1999)**. *Géomorphologie : relief, processus, environnement*, 8(2) :127–134.
- [Seaborne et al., 2008] Seaborne, A., Manjunath, G., Bizer, C., Breslin, J., Das, S., Davis, I., Harris, S., Idehen, K., Corby, O., Kjernsmo, K., et others (2008). **Sparql/update : A language for updating rdf graphs**. *W3C Member Submission*, 15.
- [Shadbolt et al., 2006] Shadbolt, N., Hall, W., et Berners-Lee, T. (2006). **The semantic web revisited**. *Intelligent Systems, IEEE*, 21(3) :96–101.
- [Sirin et al., 2007] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., et Katz, Y. (2007). **Pellet : A practical owl-dl reasoner**. *Web Semantics : science, services and agents on the World Wide Web*, 5(2) :51–53.
- [Staab et al., 2013] Staab, S., et Studer, R. (2013). **Handbook on ontologies**. Springer Science & Business Media.

- [Stell et al., 2011] Stell, J., Del Mondo, G., Thibaud, R., et Claramunt, C. (2011). **Spatio-temporal evolution as bigraph dynamics**. *Spatial Information Theory*, pages 148–167.
- [Strobl, 2008] Strobl, C. (2008). **Dimensionally extended nine-intersection model (de9im)**. *Encyclopedia of GIS*, pages 240–245.
- [Studer et al., 1998] Studer, R., Benjamins, V. R., et Fensel, D. (1998). **Knowledge engineering : principles and methods**. *Data & knowledge engineering*, 25(1) :161–197.
- [Swartout et al., 1999] Swartout, W., et Tate, A. (1999). **Guest editors' introduction : Ontologies**. *IEEE Intelligent Systems*, (1) :18–19.
- [Tao et al., 2010] Tao, J., Sirin, E., Bao, J., et McGuinness, D. L. (2010). **Integrity constraints in owl**. Dans *AAAI*.
- [ter Horst, 2005] ter Horst, H. J. (2005). **Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary**. *Web Semantics : Science, Services and Agents on the World Wide Web*, 3(2) :79–115.
- [Van Rijmenam, 2014] Van Rijmenam, M. (2014). **Think Bigger : Developing a Successful Big Data Strategy for Your Business**. AMACOM Div American Mgmt Assn.
- [Welty et al., 2006] Welty, C., Fikes, R., et Makarios, S. (2006). **A reusable ontology for fluents in owl**. Dans *FOIS*, volume 150, pages 226–236.
- [Wiederhold et al., 1991] Wiederhold, G., Jajodia, S., et Litwin, W. (1991). **Dealing with granularity of time in temporal databases**. Dans *Advanced Information Systems Engineering*, pages 124–140. Springer.
- [Worboys, 2005] Worboys, M. (2005). **Event-oriented approaches to geographic phenomena**. *International Journal of Geographical Information Science*, 19(1) :1–28.
- [Worboys, 1994] Worboys, M. F. (1994). **A unified model for spatial and temporal information**. *The Computer Journal*, 37(1) :26–34.

TABLE DES FIGURES

1.1	La réalité observable	6
1.2	Les objets de l'environnement observé ainsi que leur réalité sociale	7
1.3	Exemple d'évolution spatio-temporelle décrite au travers des différentes couches de la hiérarchie de la compréhension	9
1.4	Comparatif des outils de définition et d'analyse des bases de données (BD) classiques et du Web Sémantique(WS)	11
1.5	Représentation du contexte applicatif : Base de données relationnelles versus ontologie	16
1.6	Liste des couches et motifs génériques décrits dans le modèle	17
1.7	Illustration du processus de traitement des données au sein du modèle Continuum	18
2.1	Illustration des relations topologiques temporelles définies par Allen	25
2.2	Illustration des relations topologiques spatiales du modèle RCC8	28
2.3	Illustration des relations topologiques spatiales du modèle DE-9IM	28
2.4	Illustration des 8 scénarios de changements	30
2.5	Illustration des quatre types d'entités spatio-temporelles	31
2.6	Illustration du modèle snapshot	33
2.7	Illustration du modèle Space-Time Composite	34
2.8	Illustration du modèle ST-Objet [Worboys, 1994]	34
2.9	Illustration du modèle basé sur l'identité	35
2.10	Illustration du modèle ESTDM	36
2.11	Illustration du modèle des processus composites [Claramunt et al., 1996]	37
2.12	Illustration du modèle de changements topologiques [Jiang et al., 2009]	37
2.13	Illustration d'une base de connaissances [Baader et al., 2003]	40
2.14	Pile du Web Sémantique	43
2.15	Relation entre le Web Sémantique, les ontologies et les métadonnées	44
2.16	Ontologie GeoSPARQL	57
2.17	Temporal RDF basé sur le principe de la réification	62
2.18	Modèle des relations n-aires	63

2.19	Versioning	63
2.20	Exemple du modèle 4D-fluent	64
2.21	Illustration de la terminologie du modèle 4D-fluent	65
2.22	les 4 couches du modèle TOWL	65
2.23	Fluent Datatype Property	66
2.24	Object Datatype Property	66
2.25	Illustration du <i>Frame Problem</i>	66
2.26	Présentation du modèle	68
2.27	Illustration d'un phénomène d'artificialisation	71
2.28	Illustration d'un phénomène d'artificialisation représenté au sein d'un graphe sémantique	72
3.1	Les quatre composantes du timeslice	78
3.2	Transition entre deux timeslices	79
3.3	Description de la couche filiation	81
3.4	Description de la couche d'identité	83
3.5	Description de la couche des motifs génériques	84
3.6	Liste des motifs génériques détectés dans le modèle Continuum	88
4.1	Exemple d'évolution d'un territoire modélisé à l'aide de timeslices	95
4.2	Exemple d'évolution d'un territoire modélisé à l'aide de transitions	97
4.3	Calcul spatial des relations de filiation	98
4.4	Exemple de changement négligeable	99
4.5	Exemples de filiations acceptées et rejetées avec ρ et χ définis à 5	100
4.6	Graphe de filiation filtré dans le respect de la réalité observable	101
4.7	Les relations de continuation et dérivation	103
5.1	Résultat de l'analyse des motifs	113
6.1	Illustration de la hiérarchie de concept et de la profondeur de changement associée	117
6.2	Illustration du niveau optimal de changement	117
6.3	Déroulement de l'algorithme du niveau optimal de changement	119
6.4	Exemple de types de dérivation du jeu de données Corine Land Cover	121
6.5	Détection de motifs basée sur l'analyse de la taxonomie	121
6.6	Interprétation de phénomènes	121
6.7	Timeslices transformés de surface artificielle en étendue d'eau le long de la Garonne	123

6.8	Exemple de motif de tendance	123
6.9	Toutes les parcelles converties en tissu urbain	124
6.10	Toutes les parcelles voisines de voisines d'un timeslice	125
6.11	Toutes les parcelles voisines de voisines d'un timeslice sans doublons	125
6.12	Toutes les timeslices annexés	126
6.13	Tous les timeslices impliqués dans un processus de croissance	127
6.14	Tous les timeslices impliqués dans un processus de croissance ou d'annexion	127
6.15	Détection des évolutions connexes sur le développement des parcelles voisines	129
7.1	Diagramme UML de l'application	132
7.2	La subsomption des couches du modèle Continuum	134
7.3	Représentation de l'évolution au travers des trois couches du modèle	134
7.4	Illustration de l'architecture de l'application	137
7.5	Illustration du curseur temporel	138
7.6	Illustration des différents styles utilisés par l'application	139
7.7	Illustration de l'interface de l'application	139
B.1	Famille des logiques descriptives	171

LISTE DES TABLES

2.1	Les constructeurs OWL	45
2.2	Les axiomes OWL 2	46
2.3	Calcul de l'aire pour chaque classe de CORINE land cover au niveau 3. . .	70
4.1	La hiérarchie de classe de Corine Land Cover	94
4.2	Relation de filiations valides entre les timeslices	101
6.1	Conversion	122
7.1	Temps de conversion des shapefiles en timeslices	132
7.2	Insertion des données dans le triplestore	132
7.3	Nombre de timeslices en Gironde	133
7.4	Nombre de transitions spatio-temporelles	133
7.5	Nombre de motifs détectés sur le département de la Gironde et temps d'exécution de la requête	135
B.1	Récapitulatif des différents constructeurs en logique de description	170
D.1	Règle d'implication RDFS	192
D.2	Ensemble de règles d'implication OWL Horst	193

IV

ANNEXES

LISTE DES PUBLICATIONS SUR LES TRAVAUX

Les travaux effectués dans le cadre de cette thèse ont donné lieu à plusieurs publications dont la liste est donnée ci-dessous :

[Harbelot et al., 2015b] Harbelot, B. et al. 2015. LC3 : A spatio-temporal and semantic model for a knowledge discovery from geospatial datasets. *Journal of Web Semantics, Special Issue on Geospatial Semantics*, Elsevier, 2015. (ISI WoS IF 2.550, SCImago Q1) (A paraître)

[Harbelot et al., 2015a] Harbelot, B. et al. 2015. LC3 : un modèle spatial et sémantique pour découvrir la connaissance dans les jeux de données. (2015). *Atelier Gestion et Analyse des données Spatiales et Temporelles. Conférence Extraction et Gestion de la Connaissance 2015*.

[Arenas et al.,] Arenas, H. et al. 2014. A Semantic Analysis of Moving Objects Using as a Case Study Maritime Voyages from Eighteenth and Nineteenth Centuries. *GEOProcessing 2014, The Sixth International Conference on Advanced Geographic Information Systems, Applications, and Services (2014)*, 45–50.

[Arenas et al., 2015] Arenas, H. et al. 2014. Reasoning with Vague Spatial Information from Upper Mesopotamia (2000BC). *Procedia Environmental Sciences. Volume 27, 2015, Pages 58–65. Spatial Statistics conference 2015*.

[Harbelot et al., 2014b] : Harbelot, B. et al. 2014. A Semantic Model to Query Spatial–Temporal Data. *Information Fusion and Geographic Information Systems (IF AND GIS 2013)*. Springer Berlin Heidelberg. 75–89.

[Harbelot et al., 2014c] Harbelot, B. et al. 2014. Un modèle sémantique spatio-temporel pour capturer la dynamique des environnements. *Fouille de données spatiales et temporelles, 2014*.

[Harbelot et al., 2013a] Harbelot, B. et al. 2013. Continuum : A spatiotemporal data model to represent and qualify filiation relationships. *4th ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS) 2013 (2013)*.

[Harbelot et al., 2013b] Harbelot, B. et al. 2013. Semantics for Spatio-temporal Smart Queries. *9th International Conference on Web Information Systems and Technologies (WEBIST) (2013)*.

[Harbelot et al., 2013c] Harbelot, B. et al. 2013. The spatio-temporal semantics from a

perdurantism perspective. In Proceedings of the Fifth International Conference on Advanced Geographic Information Systems, Applications, and Services GEOProcessing (2013).

[Harbelot et al., 2013d] Harbelot, B. et al. 2013. Using Semantic Web Technologies to Follow the Evolution of Entities in Time and Space. International Journal On Advances in Intelligent Systems. (2013).

[Arenas et al., 2013] Arenas, H. et al. 2013. A Semantic Web Approach for Geodata Discovery. 7th International Workshop on Semantic and Conceptual Issues in GIS (SeCoGIS 2013) (2013).

INTRODUCTION AUX LOGIQUES DE DESCRIPTION

Sommaire

B.1 Les familles des logiques de description	163
B.1.1 \mathcal{AL} : la logique de description de base	164
B.1.2 Le constructeur de hiérarchie de rôles \mathcal{H}	165
B.1.3 Le constructeur de conjonction de rôles \mathcal{R}	166
B.1.4 Le constructeur de rôles transitifs \mathcal{R}^+	166
B.1.5 Le constructeur d'énumération d'individus \mathcal{O}	166
B.1.6 Le constructeur d'union \mathcal{U}	166
B.1.7 Le constructeur de quantification existentielle complète \mathcal{E}	167
B.1.8 Le constructeur de fonction \mathcal{F}	167
B.1.9 Le constructeur d'inverse \mathcal{I}	167
B.1.10 Les restrictions de cardinalité \mathcal{N}	168
B.1.11 Les restrictions de cardinalité qualifiée \mathcal{Q}	169
B.1.12 La négation sans restriction \mathcal{C}	169
B.1.13 Discussion	169
B.2 Les raisonnements en logiques de description	172
B.2.1 Raisonnement sur la TBox	172
B.2.2 Raisonnement sur la ABox	173
B.2.3 Complexité de l'inférence	174

B.1/ LES FAMILLES DES LOGIQUES DE DESCRIPTION

La logique de description est un langage formel permettant de représenter des connaissances. En réalité, la logique de description regroupe une famille de formalisme permettant de représenter des concepts (ou classes) d'un domaine ainsi que les relations (ou rôles) pouvant être établies entre les instances de ces classes. Les langages de la logique descriptive sont déterminés par la forme des énoncés qui sont permis. L'expressivité des langages ou des familles est généralement déterminé par une lettre pouvant correspondre, par exemple, à la notion de constructeurs, quantificateurs, restriction, etc. La plupart des langages utilisés découlent du langage \mathcal{AL} (Attributive Language). Si le langage \mathcal{AL} s'impose comme la logique de description de base, l'expressivité au sein de ce langage reste cependant limitée. Dans un premier temps, nous présenterons le

langage \mathcal{AL} puis nous introduirons les différents constructeurs pouvant être additionnés pour obtenir plusieurs nouveaux langages et accroître l'expressivité.

B.1.1/ \mathcal{AL} : LA LOGIQUE DE DESCRIPTION DE BASE

Dans le langage \mathcal{AL} , les axiomes sont définis à partir d'un ensemble de concepts. Il est, par exemple, possible de déclarer qu'une *zone industrielle* est une *zone d'activité* en utilisant l'axiome suivant :

$$ZoneIndustrielle \sqsubseteq ZoneActivité \quad (B.1)$$

Toutefois, cet axiome ne révèle que peu de connaissances en déclarant que les *zones industrielles* forment un sous-ensemble des *zones d'activités* et ne décrit pas réellement ce qu'est une zone industrielle. Pour décrire plus précisément le concept de *zone industrielle*, il est nécessaire de citer d'autres caractéristiques. Par exemple, en définissant qu'une *zone industrielle* est une zone d'activité comportant des parcs industriels et des infrastructures pour y accéder.

$$ZoneIndustrielle \equiv ZoneActivité \sqcap ParcIndustriel \sqcap Infrastructures \quad (B.2)$$

Cet axiome représentant une connaissance plus précise de notre concept de *Zone Industrielle* est un énoncé de la forme $C \equiv D$ avec C un concept atomique. Cet énoncé est un exemple de *définition* que nous avons abordé précédemment pour créer la TBox d'une base de connaissance. Lors de la construction d'une base de connaissance, les concepts et relations peuvent être organisés au sein d'une hiérarchie via un mécanisme de subsomption décrit par l'opérateur \sqsubseteq . Ainsi, la hiérarchie permet d'organiser des concepts du plus générique au plus spécifique (les concepts peuvent évidemment également exister au même niveau de la hiérarchie). Par exemple, le concept *Zone d'Activité* généralise le concept *Zone Industrielle* ou autrement dit, le concept *Zone Industrielle* spécialise le concept *Zone d'Activité*, ce qui signifie que tout individu appartenant au concept *Zone Industrielle* appartient également au concept *Zone d'Activité*.

Cette hiérarchie est encadrée par deux concepts spéciaux :

- le concept universel \top qui représente tous les individus du monde représenté, en d'autres termes, le concept universel généralise tous les concepts définis dans la hiérarchie. Ainsi pour tout concept C , l'axiome suivant est valable : $C \sqsubseteq \top$
- le concept impossible \perp qui définit qu'aucun individu ne peut appartenir à ce concept, en d'autres termes, le concept impossible spécialise tous les concepts définis dans la hiérarchie. Ainsi pour tout concept C , l'axiome suivant est valable : $\perp \sqsubseteq C$

Outre les opérateurs d'équivalence et de subsomption, le langage \mathcal{AL} comporte également la négation. Toutefois, celle-ci n'est applicable qu'à un concept atomique. Ainsi, il est possible de définir le concept $\neg ZoneIndustrielle$ représentant tous les individus qui ne sont pas des zones industrielles. En revanche, $\neg (ZoneActivité \sqcap ParcIndustriel \sqcap Infrastructures)$ ne peut être défini par l'opérateur de négation du langage \mathcal{AL} . Enfin, le langage \mathcal{AL} permet de définir des concepts à l'aide de restrictions sur les cardinalités.

Par exemple, il est possible de définir un concept qui, pour un territoire donné, regroupe

toutes les zones qui sont de type *Zone Commerciale* : $\forall \text{comporteZone.ZoneCommerciale}$. Il est également possible de définir un concept qui, pour un territoire donné, comporte au moins un centre commercial : $\exists \text{possèdeCentreCommercial}.\top$. Le concept universel \top est obligatoirement utilisé avec le quantificateur existentiel car le langage \mathcal{AL} ne permet pas de spécifier d'autres concepts pour ce quantificateur. Ainsi, il est impossible de décrire le concept : $\exists \text{comporteZone.ZoneCommerciale}$.

A présent, introduisons un exemple plus complexe avec le concept de *Territoire Industrialisé* qui est un territoire composé de zones qui sont toutes des *Zones Industrielles*.

$$\text{TerritoireIndustrialisé} \equiv \text{Territoire} \sqcap \exists \text{comporteZone}.\top \sqcap \forall \text{comporteZone.ZoneIndustrielle} \quad (\text{B.3})$$

Ainsi, un territoire industrialisé est décrit comme étant un territoire comportant au moins une zone et dont toutes les zones sont des zones industrielles. Dès lors, on remarque que simplifier la définition précédente avec la définition ci-après ne serait pas correcte :

$$\text{TerritoireIndustrialisé} \equiv \text{Territoire} \sqcap \forall \text{comporteZone.ZoneIndustrielle} \quad (\text{B.4})$$

En effet, bien que la définition ci-dessus interdirait les territoires qui posséderaient plusieurs zones dont au moins une ne serait pas une zone industrielle, elle autoriserait, en revanche, tous les territoires ne possédant aucune zone car si le territoire ne possède aucune zone alors il n'est pas possible de dire qu'il existe une zone qui ne soit pas une zone industrielle et la restriction sera donc respectée. Ainsi cette définition autorise des résultats qui ne sont pas en adéquation avec la réalité que nous souhaitons modéliser.

En résumé, le langage \mathcal{AL} comporte les descriptions suivantes :

Symbole	Description
\top	concept universel
\perp	concept impossible
A	concept atomique
$\neg A$	négation du concept atomique
$C \sqcap D$	intersection de concepts quelconques
$\forall R.C$	restriction de valeurs pour des concepts quelconques
$\exists R.\top$	quantification existentielle limitée
$C \sqsubseteq D$	inclusion de concepts quelconques
$C \equiv D$	équivalence de concepts quelconques
$C(a)$ ou $a : C$	instanciation de concepts
$R(a, b)$ ou $(a, b) : R$	instanciation de rôles

B.1.2/ LE CONSTRUCTEUR DE HIÉRARCHIE DE RÔLES \mathcal{H}

Dans le langage \mathcal{AL} , seule l'inclusion de concepts est autorisée. Le constructeur \mathcal{H} permet de définir l'inclusion de rôles au sein de la base de connaissances sous la forme $R_1 \sqsubseteq R_2$ avec R_1 et R_2 des rôles quelconques. Par exemple, prenons la propriété *estVilleDe* s'appliquant entre un concept *Ville* et un concept *Région* et la propriété *estCapitaleDe* s'appliquant entre un concept *Ville* et un concept *Région*. Le constructeur \mathcal{H} autorise de définir l'axiome suivant :

$$\text{estCapitaleDe} \sqsubseteq \text{estVilleDe} \quad (\text{B.5})$$

Ainsi, la propriété *estCapitaleDe* est définie comme sous-propriété de la propriété *estVillageDe*.

B.1.3/ LE CONSTRUCTEUR DE CONJONCTION DE RÔLES \mathcal{R}

Le constructeur \mathcal{R} permet d'ajouter la conjonction de rôles au sein de la base de connaissances sous la forme $R_1 \sqcap R_2$ avec R_1 et R_2 des rôles quelconques. Par exemple, la relation *prendreTransportEnCommun* représente la conjonction de ceux qui prennent le bus, le train et le métro :

$$\text{prendreTransportEnCommun} \equiv \text{prendreBus} \sqcap \text{prendreTrain} \sqcap \text{prendreMétro} \quad (\text{B.6})$$

B.1.4/ LE CONSTRUCTEUR DE RÔLES TRANSITIFS \mathcal{R}^+

Le constructeur \mathcal{R}^+ permet de définir des rôles comme étant transitifs. Prenons par exemple la relation *Contient* établie entre des parcelles. Si une parcelle A contient une parcelle B , et que cette parcelle B contient une parcelle C alors on peut en déduire que la parcelle A contient la parcelle C ce qui signifie que la relation *Contient* est une relation transitive. En logique de description, cela se décrit comme suit : $Tr(\text{Contient})$.

B.1.5/ LE CONSTRUCTEUR D'ÉNUMÉRATION D'INDIVIDUS \mathcal{O}

Le constructeur \mathcal{O} se distingue des autres constructeurs en cela qu'il permet de désigner des individus dans une terminologie ce qui est normalement réservé à la ABox. Prenons la liste des Départements et Régions d'Outre-Mer (DROM) français. Pour exprimer le concept de *MembreDesDROM*, le meilleur moyen d'y parvenir est d'explicitement la liste des départements et régions qui en font partis. Habituellement, il est impossible de définir des individus au sein de la TBox, c'est pourquoi il est nécessaire d'introduire un nouveau constructeur : l'énumération symbolisée par la lettre \mathcal{O} . Ainsi, l'énumération prend la forme de $\{a_1, \dots, a_n\}$ où a_1, \dots, a_n sont des noms d'individus. La liste des membres des DROM se décrit par l'assertion suivante :

$$\text{MembreDesDROM} \equiv \{\text{Guadeloupe}, \text{Martinique}, \text{Guyane}, \text{La Réunion}, \text{Mayotte}\} \quad (\text{B.7})$$

B.1.6/ LE CONSTRUCTEUR D'UNION \mathcal{U}

Le symbole \mathcal{U} permet de représenter l'union de concepts. En considérant deux concepts quelconques C et D , l'union de concepts permet de représenter l'ensemble des individus appartenant soit à la classe C , soit à la classe D via la description suivante :

$$C \sqcup D \quad (\text{B.8})$$

A l'aide de cet opérateur, il est par exemple possible de représenter un territoire ne comportant que des zones industrielles ou des zones commerciales à l'aide de la définition

suivante :

$$\begin{aligned} \text{TerritoireIndustrielCommercial} \equiv \\ \text{Territoire} \sqcap \forall \text{comporteZone} . (\text{ZoneIndustrielle} \sqcup \text{ZoneCommerciale}) \end{aligned} \quad (\text{B.9})$$

Ainsi pour appartenir au concept *TerritoireIndustrielCommercial*, un individu doit appartenir au concept *Territoire* tel que si ce territoire comporte des zones alors ces zones sont de type *ZoneIndustrielle* ou *ZoneCommerciale*.

B.1.7/ LE CONSTRUCTEUR DE QUANTIFICATION EXISTENTIELLE COMPLÈTE \exists

Comme nous l'avons étudié précédemment, le quantificateur existentiel existe au sein du langage \mathcal{AL} pour permettre de définir qu'une entité doit avoir au moins une relation avec un autre objet. Cependant, il est impossible de spécifier la classe de cet autre objet, c'est pourquoi le langage \mathcal{AL} nécessite d'utiliser le concept universel \top à chaque usage du quantificateur existentiel. Ceci limite l'expressivité du langage. Ainsi, le constructeur de quantification existentielle complète dépasse cette limitation pour permettre des définitions de la forme $\exists R.C$ avec R une relation et C un concept pouvant être différent du concept universel. Dès lors, il est possible de représenter la classe des territoires qui possèdent au moins une zone commerciale :

$$\exists \text{comporteZone} . \text{ZoneCommerciale} \quad (\text{B.10})$$

B.1.8/ LE CONSTRUCTEUR DE FONCTION \mathcal{F}

Le constructeur \mathcal{F} permet de spécifier qu'une relation est fonctionnelle. Une relation fonctionnelle caractérise une relation pour laquelle aucune entité ne peut être liée plus d'une fois à une autre par cette relation. Prenons l'exemple de la relation *estCapitaleDe* s'appliquant entre un individu de la classe *Ville* et un individu de la classe *Région*. Cette propriété doit être définie comme fonctionnelle car une capitale ne peut être contenue dans plus d'une région.

$$\text{CapitaleRégion} \equiv \text{Ville} \sqcap \exists \text{estCapitaleDe} . \top \quad (\text{B.11})$$

La définition ci-dessus n'empêche pas une instance de la classe *CapitaleRégion* d'être liée à plus d'une *région* par la relation *estCapitaleDe*. Ainsi, il est nécessaire de définir en supplément un axiome de la forme $\text{Fun}(R)$ afin de définir qu'un rôle R est fonctionnel : $\text{Fun}(\text{estCapitaleDe})$.

B.1.9/ LE CONSTRUCTEUR D'INVERSE \mathcal{I}

Le constructeur d'inversion \mathcal{I} permet de définir qu'un rôle est l'inverse d'un autre rôle. Par exemple, si la tour Eiffel se situe à Paris alors Paris contient la tour Eiffel. Ainsi, en considérant deux relations *seSitueDans* et *Contient* alors toute assertion de la forme $\text{seSitueDans}(x, y)$ implique $\text{Contient}(y, x)$. L'axiome suivant permet d'exprimer que les deux relations sont inverses :

$$seSitueDans \equiv Contient^- \quad (B.12)$$

B.1.10/ LES RESTRICTIONS DE CARDINALITÉ \mathcal{N}

Le constructeur \mathcal{N} apporte la restriction sur les cardinalités permettant de représenter des concepts tels que l'ensemble de ceux qui possèdent au moins deux lacs ou bien au plus cinq lacs à l'aide de définition de la forme $\leq nR$ et $\geq nR$. En combinant les deux opérateurs, il est possible de définir l'ensemble des villes possédant exactement trois lacs par exemple :

$$Ville \sqcap \leq 3possèdeLac \sqcap \geq 3possèdeLac \quad (B.13)$$

En outre, en utilisant des valeurs de cardinalité 0 ou 1, il est alors possible d'exprimer des logiques \mathcal{ALCF} , c'est-à-dire d'exprimer des propriétés fonctionnelles en offrant un autre formalisme. En effet, pour le constructeur \mathcal{F} , nous avons défini un axiome de la forme $Fun(R)$. A présent, en utilisant le constructeur \mathcal{N} , pour spécifier qu'une relation est fonctionnelle, il est possible d'utiliser l'axiome suivant :

$$\top \sqsubseteq \leq 1R \quad (B.14)$$

Cet axiome signifie que quel que soit, l'entité décrite, elle ne peut être liée à plus d'une entité par la relation R ce qui correspond effectivement à la définition d'une relation fonctionnelle. Ainsi en reprenant notre exemple de la sous-section B.1.8 l'axiome $Fun(estCapitaleDe)$ est équivalent à $\top \sqsubseteq \leq 1estCapitaleDe$. Toutefois, cet axiome doit être défini en dehors de la définition impliquant la relation fonctionnelle. Prenons, par exemple, la définition suivante :

$$CapitaleRégion \equiv Ville \sqcap \exists estCapitaleDe. \top \sqcap \leq 1estCapitaleDe \quad (B.15)$$

La définition ci-dessus indique qu'une capitale est une ville qui est définie comme la capitale exactement d'une seule Région. Ainsi, la définition ne définit pas la propriété $estCapitaleDe$ comme fonctionnelle mais décrit seulement une restriction. En effet, considérons cette fois la définition d'un concept $CapitalePays$ définissant une ville comme capitale d'un pays au sein de la même base de connaissance et à l'aide de la même propriété $estCapitaleDe$ avec l'axiome suivant :

$$CapitalePays \equiv Ville \sqcap \exists estCapitaleDe. Pays \quad (B.16)$$

L'axiome ci-dessus utilise bien la même propriété que pour l'axiome B.15 pourtant rien n'interdit ici de lier plus d'un pays à une même ville via la relation $estCapitaleDe$. Ainsi une restriction ne s'applique qu'à la définition d'un concept particulier tandis qu'une propriété définie comme fonctionnelle concerne chaque entité liée à d'autres entités via cette propriété fonctionnelle au sein de la base de connaissance.

B.1.11/ LES RESTRICTIONS DE CARDINALITÉ QUALIFIÉE Q

Tout comme le quantificateur existentiel du langage \mathcal{AL} ne peut s'appliquer sur d'autres concepts que le concept universel \top , les restrictions de cardinalité du constructeur \mathcal{N} ne peuvent s'appliquer sur des classes spécifiques de la base de connaissance. Par exemple, en logique \mathcal{ALN} , il n'est pas possible de définir l'ensemble des villes comportant deux individus ou plus via la relation *possède* :

$$\text{Ville} \sqcap \leq 2 \text{ possède } . \text{Lac} \sqcap \geq 3 \text{ possède } . \text{Lac} \quad (\text{B.17})$$

Le constructeur de restrictions de cardinalité qualifiées Q permet de définir des axiomes de la forme $\leq nR.C$ et $\geq nR.C$. Ainsi, il est désormais possible de définir l'axiome B.17 à l'aide de ces nouveaux opérateurs.

B.1.12/ LA NÉGATION SANS RESTRICTION C

Le langage de base \mathcal{AL} autorisait la négation mais uniquement sur des concepts atomiques. Le constructeur C (complément) permet d'étendre le langage \mathcal{AL} en ajoutant la possibilité d'utiliser la négation sans aucune restriction. Dès lors, il est possible de décrire la classe des territoires ne comportant pas de zones commerciales.

$$\text{Territoire} \sqcap \neg \exists \text{ comporte } \text{Zone} . \text{ZoneCommerciale} \quad (\text{B.18})$$

Il est également possible de représenter la classe des territoire qui ne comporte pas de zones industrielles adjacentes à une forêt :

$$\text{Territoire} \sqcap \forall \text{ comporte } \text{Zone} . \neg (\text{ZoneIndustrielle} \sqcap \text{ adjacent } A . \text{Forêt}) \quad (\text{B.19})$$

Cependant, il est important de noter que le constructeur C ne le rend pas le langage de base plus expressif car il est toujours possible de ramener l'axiome à la négation du langage \mathcal{AL} :

$$\begin{aligned} & \text{Territoire} \sqcap \forall \text{ comporte } \text{Zone} . \neg \text{ZoneCommerciale} \\ & \text{Territoire} \sqcap \forall \text{ comporte } \text{Zone} . (\neg \text{ZoneIndustrielle} \sqcup \forall \text{ adjacent } A . \neg \text{Forêt}) \end{aligned} \quad (\text{B.20})$$

Littéralement, cela pourrait se traduire par : "Quelles que soient les zones d'un territoire, ces zones ne sont pas des zones commerciales" et "Quelles que soient les zones d'un territoire, ces zones ne sont pas des zones industrielles pour lesquelles quel que soit le voisinage, il ne s'agit pas d'une forêt".

B.1.13/ DISCUSSION

En étendant le langage \mathcal{AL} avec l'un des constructeur, on obtient des langages de la famille \mathcal{AL} de la forme :

$$\mathcal{AL}[\mathcal{U}], [\mathcal{E}], [\mathcal{C}], [\mathcal{N}] \quad (\text{B.21})$$

Constructeurs	Syntaxe	Description
\mathcal{F}	$\leq 1R$ ou $Fun(R)$	Rôles fonctionnels
\mathcal{E}	$\exists R.C$	Quantification existentielle complète
\mathcal{U}	$C_1 \sqcup C_2$	Union de rôles
\mathcal{C}	$\neg C$	Négation complète
\mathcal{H}	$R_1 \sqsubseteq R_2$	Hiérarchie de rôles
\mathcal{O}		Énumération d'individus
\mathcal{I}	R^-	Rôles inverses
\mathcal{N}	$\geq nR$	Restriction de cardinalité
	$\leq nR$	
	$= nR$	
\mathcal{Q}	$\geq nR.C$	Restriction de cardinalité qualifiée
	$\leq nR.C$	
	$= nR.C$	
\mathcal{R}	$R_1 \sqcap R_2$	Conjonction de rôles
\mathcal{R}^+	R^+	Transitivité des rôles

TABLE B.1 – Récapitulatif des différents constructeurs en logique de description

Chaque lettre correspond à un constructeur et permet d'étendre l'expressivité du langage de base. Par exemple, \mathcal{ALEN} est une extension de \mathcal{AL} par le quantificateur existentiel complet et la restriction de cardinalité. D'un point de vue sémantique, certains des langages sont équivalents. En effet, en tenant compte des équivalences suivantes : $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$ et $\exists R.C \equiv \neg \forall R.\neg C$, il est intéressant de noter que l'union et le quantificateur existentiel complet peuvent être exprimés à l'aide de la négation. Ou inversement, la combinaison de l'union et du quantificateur existentiel complet nous donne la possibilité de décrire la négation des concepts. Dès lors, il est possible de considérer que l'union et le quantificateur existentiel complet sont présents dans tous les langages qui contiennent la négation et inversement. De ce fait, il n'est pas nécessaire de distinguer des langages \mathcal{AL} possédant la négation de ceux possédant la combinaison de l'union et du quantificateur existentiel complet. Par convention, la lettre C est utilisée à la place des lettres \mathcal{UE} . Par exemple, nous écrirons \mathcal{ALC} à la place de \mathcal{ALUE} et \mathcal{ALCN} à la place de \mathcal{ALUEN} .

Les langages de la famille \mathcal{AL} autorisent une description détaillée des concepts de la base de connaissance mais restent cependant limités dans la description des rôles. Afin de dépasser cette limitation viennent alors les langages de la famille \mathcal{SH} . Le constructeur \mathcal{S} est équivalent à la logique \mathcal{ALC} (ou \mathcal{ALUE}) augmentée par \mathcal{R}^+ .

La figure B.1 est issue de [Staab et al., 2013], [Baader et al., 2003] et illustre les différentes familles des logiques descriptives et le mode de construction des langages. Il est intéressant de noter qu'il n'est pas nécessaire de cumuler les constructeurs \mathcal{F} , \mathcal{N} , \mathcal{Q} car \mathcal{Q} inclut les capacités de \mathcal{N} qui lui-même inclut les capacités de \mathcal{F} . Pour s'en convaincre, reprenons les éléments suivants du tableau B.1 :

Constructeurs	Syntaxe	Description
\mathcal{F}	$\leq 1R$	Rôles fonctionnels
	$\geq nR$	
\mathcal{N}	$\leq nR$	Restriction de cardinalité
	$= nR$	
	$\geq nR.C$	
\mathcal{Q}	$\leq nR.C$	Restriction de cardinalité qualifiée
	$= nR.C$	
	$= nR.C$	

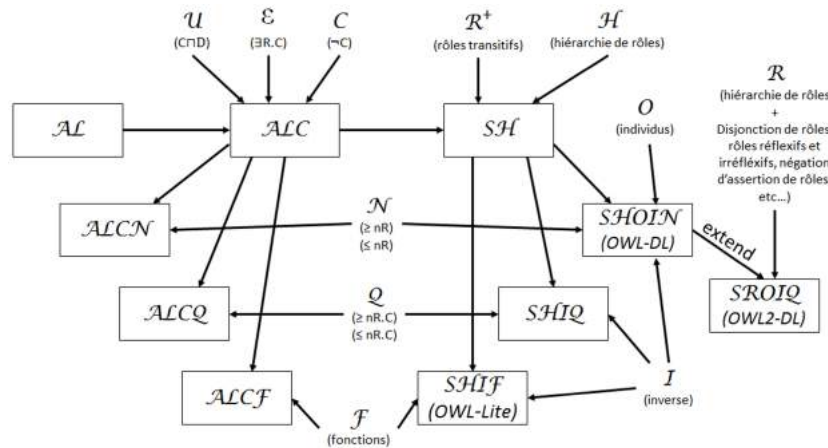


FIGURE B.1 – Famille des logiques descriptives

Dès lors, il est aisé de comprendre que toutes les propriétés valides pour le constructeur \mathcal{F} le sont également pour \mathcal{N} et \mathcal{Q} . L'inverse n'est bien évidemment pas vrai. Par exemple, la propriété fonctionnelle *estMariéA* peut être définie avec \mathcal{F} , \mathcal{N} et \mathcal{Q} :

$$\leq 1 \text{estMariéA} \tag{B.22}$$

En revanche, la restriction de cardinalité de la propriété *possèdeRoue* où le nombre de roue est de 4, peut être définie avec \mathcal{N} ou \mathcal{Q} mais ne peut l'être avec \mathcal{F} :

$$= 4 \text{possèdeRoue} \tag{B.23}$$

De même, si cette restriction précise que cette propriété ne s'applique qu'à des instances du concept *Voiture* :

$$= 4 \text{possèdeRoue.Voiture} \tag{B.24}$$

alors seul le constructeur \mathcal{Q} est valable pour définir la restriction de cette propriété. En définitive, il est possible d'établir que : $\mathcal{F} \subseteq \mathcal{N} \subseteq \mathcal{Q}$.

Nous venons de voir les logiques de description qui représentent une famille de langage. Mais alors parmi tous ces langages, pourquoi faire un choix ? Pourquoi ne pas tout simplement prendre le plus expressif pour être sûr de pouvoir tout représenter à notre convenance ? Pour répondre à ces questions, il faut se souvenir qu'une base de connaissances permet effectivement de décrire les connaissances d'un domaine spécialisé, mais peu aussi et surtout permettre de raisonner sur ces connaissances. Toutefois, dans le respect des performances des outils actuels, les temps de calcul pour un raisonnement donné peuvent être extrêmement longs. La complexité du raisonnement, et par conséquent, le temps de calcul et la quantité de mémoire nécessaires à sa réalisation, dépendent de la taille de la base de connaissances ainsi que de son niveau d'expressivité. Dès lors, réduire l'expressivité d'une base de connaissances permet de garantir que les mécanismes de raisonnement seront décidables et auront une complexité maximale. Ainsi, des choix doivent être faits afin de trouver un compromis acceptable entre expressivité et complexité pour une base de connaissances donnée. Dans la prochaine section, nous étudierons les différents types de raisonnement applicables sur une base

de connaissances puis nous discuterons de leur complexité en fonction de la taille de la base de connaissances et de la logique de description utilisée.

B.2/ LES RAISONNEMENTS EN LOGIQUES DE DESCRIPTION

Le raisonnement est un processus qui permet de produire de nouveaux résultats ou de vérifier des faits. Cependant, il existe différents types de raisonnements selon qu'ils soient appliqués sur la TBox ou la ABox. Dans cette section, nous étudierons les différents types de raisonnements propres à la logique de description. Dans un second temps, nous discuterons de la complexité des raisonnements.

B.2.1/ RAISONNEMENT SUR LA TBOX

La TBox (\mathcal{T}) décrit les connaissances générales d'un domaine au travers d'un ensemble de classes et propriétés. Il existe quatre principaux problèmes d'inférences au niveau terminologique [Baader et al., 2003] :

- **Satisfaisabilité** : Un concept C d'une terminologie \mathcal{T} est satisfiable si et seulement si il existe un modèle \mathcal{I} de \mathcal{T} tel que $C^{\mathcal{I}} \neq \emptyset$. Autrement dit, un concept est satisfiable si au moins un individu peut appartenir à l'ensemble décrit par ce concept. Par exemple, le concept $Forêt \sqcap \neg Forêt$ est insatisfiable puisqu'il est impossible d'incarner un concept et son contraire. Par conséquent, aucun individu ne peut appartenir à la classe représentant l'intersection du concept $Forêt$ avec le concept $\neg Forêt$.
- **Subsomption** : Un concept C est subsumé par un concept D pour une terminologie si et seulement si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ pour tout modèle \mathcal{I} de \mathcal{T} . En utilisant la syntaxe de la logique de description, on écrira $\mathcal{T} \models C \sqsubseteq D$. Ainsi, en reprenant la définition suivante d'une zone industrielle : $ZoneIndustrielle \equiv ZoneActivité \sqcap \exists \text{possède.ParcIndustriel} \sqcap \exists \text{possède.Infrastructures}$, il est possible de démontrer que $ZoneIndustrielle \sqsubseteq ZoneActivité$. En effet, une zone industrielle est une zone d'activité possédant des caractéristiques particulières permettant distinguer les deux concepts. En d'autres termes, une zone industrielle est un type particulier de zone d'activité. Ainsi, si un individu appartient au concept $ZoneIndustrielle$ alors il appartient également au concept $ZoneActivité$. Il est également possible d'effectuer des raisonnements en définissant des concepts tels que : $TissuUrbainContinu \sqsubseteq ZoneUrbanisée$ et $ZoneUrbanisée \sqsubseteq TerritoireArtificialisé$. Dès lors, il est possible d'inférer que $TissuUrbainContinu \sqsubseteq TerritoireArtificialisé$. D'autres exemples impliquant l'inférence d'une relation de subsomption existent mais ne seront pas détaillés dans cette section.
- **Équivalence** : Un concept C est équivalent à un concept D pour une terminologie si et seulement si $C^{\mathcal{I}} = D^{\mathcal{I}}$ pour chaque modèle \mathcal{I} de \mathcal{T} . En utilisant la syntaxe de la logique de description, on écrira $\mathcal{T} \models C \equiv D$.
- **Disjonction** : Des concepts C et D sont disjoints par rapport terminologie si et seulement si $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ pour chaque modèle \mathcal{I} de \mathcal{T} . En utilisant la syntaxe

de la logique de description, on écrira $\mathcal{T} \models C \sqcap D \sqsubseteq \perp$. Prenons les concepts *Ville* et *Campagne* avec $Ville \equiv \neg Campagne$ alors $Ville \sqcap Campagne \sqsubseteq \perp$. Ainsi, il est intéressant de noter que le complément d'un concept implique la disjonction entre les deux concepts tandis que la disjonction n'implique pas nécessairement le complément. Par exemple, imaginons les concepts *ZoneIndustrielle* et *ZoneAgricole* définis comme disjoints. Cela n'implique aucunement que *ZoneIndustrielle* soit le complément de *ZoneAgricole*. Dans ce cas précis, ça n'est d'ailleurs pas le cas car ces deux concepts sont des sous-concepts de *ZoneActivité* qui regroupent également les concepts de *ZoneArtisanale*, *ZonePortuaire*, *ZoneTechnologique*, etc.

B.2.2/ RAISONNEMENT SUR LA ABOX

Après que la TBox ait été définie et des mécanismes de raisonnement appliqués dessus afin de vérifier que les concepts sont satisfiables ou encore que les relations de sub-somption inférées respectent la hiérarchie de concepts définis, la ABox peut alors être remplie avec des instances. Pour rappel, il existe deux types d'instances que sont les instances de concept de la forme $C(a)$ et les instances de rôles de la forme $R(a, b)$. L'un des raisonnements typiques de la ABox consiste à vérifier la consistance des connaissances insérées dans la ABox. Par exemple, si la ABox possède les axiomes suivants : *Région(Bourgogne)* et *Département(Bourgogne)* et si la TBox définit que *Région* et *Département* sont des concepts disjoints alors ces deux axiomes sont inconsistants car la Bourgogne ne peut être à la fois un département et une région. Littéralement, la consistance d'une ABox consiste à s'assurer que celle-ci ne contienne aucune assertion $C(a)$ telle qu'il est impossible que a appartienne à C . De façon plus formelle, la définition de la consistance est la suivante : une ABox (\mathcal{A}) est consistante dans le respect de la TBox (\mathcal{T}), s'il existe une interprétation qui satisfait à la fois \mathcal{A} et \mathcal{T} . Dès lors, il est intéressant de noter que pour une TBox vide, toutes les assertions sont consistantes dans le respect de la TBox puisqu'aucune restriction n'est établie. Ainsi, si les concepts *Région* et *Département* n'étaient pas définis comme disjoints dans la TBox, les axiomes *Région(Bourgogne)* et *Département(Bourgogne)* auraient été consistants. Le même principe s'applique sur les instances de rôles, ainsi la vérification de la consistance d'une base de connaissances regroupe deux problèmes d'inférences que sont :

- **La vérification d'instance** : Vérifier par inférence si une assertion $C(a)$ est vraie pour tout modèle \mathcal{I} d'une ABox \mathcal{A} et d'une TBox \mathcal{T} .
- **La vérification de rôle** : Vérifier par inférence si une assertion $R(a, b)$ est vraie pour tout modèle \mathcal{I} d'une ABox \mathcal{A} et d'une TBox \mathcal{T} .

Un autre type d'inférence pouvant être appliqué sur une ABox consiste à déterminer si un individu appartient à une certaine classe. Ce problème est connu sous le nom de problème de récupération et défini de la façon suivante :

- **Le problème de récupération** : Pour une ABox \mathcal{A} , un concept C d'une terminologie \mathcal{T} , inférer les individus $a_1^{\mathcal{I}} \dots a_n^{\mathcal{I}} \in C^{\mathcal{I}}$ pour tout modèle \mathcal{I} de \mathcal{T} .

Ainsi $\mathcal{A} \models C(a)$ si à partir d'une ABox \mathcal{A} , on peut déduire que a appartient au concept C . Si la ABox contient directement l'assertion $C(a)$, l'assertion est triviale, cependant il existe des cas plus complexes et plus intéressants. Reprenons l'exemple suivant permettant de définir le concept de *ZoneIndustrielle* dans la TBox \mathcal{T} :

$$\text{ZoneIndustrielle} \equiv \text{ZoneActivité} \sqcap \text{ParcIndustriel} \sqcap \text{Infrastructures} \quad (\text{B.25})$$

Considérons, à présent, la ABox \mathcal{A} contenant l'assertion $\text{ZoneIndustrielle}(\text{TroyesZI})$, il est alors possible d'inférer l'assertion $\text{ZoneActivité}(\text{TroyesZI})$. Ce type d'inférence existe sous de nombreuses formes au sein d'une base de connaissances. Prenons, par exemple une TBox contenant les assertions suivantes :

$$\begin{aligned} \exists \text{estCapitaleDe} &\sqsubseteq \text{Ville} \\ \top &\sqsubseteq \forall \text{estCapitaleDe}.\text{Pays} \end{aligned} \quad (\text{B.26})$$

Le premier axiome signifie que la relation estCapitaleDe ne peut s'appliquer qu'à des individus de la classe Ville . Le deuxième axiome définit que tous les individus de domaine de la relation estCapitaleDe appartiennent à la classe Pays . En d'autres termes, la relation estCapitaleDe ne s'applique qu'entre un individu de la classe Ville et un individu de la classe Pays avec le concept Ville comme domaine de la relation et le concept Pays comme image de la relation. En considérant, également une ABox contenant l'assertion suivante : $\text{estCapitaleDe}(\text{Paris}, \text{France})$, il est alors possible d'inférer les assertions suivantes : $\text{Ville}(\text{Paris})$ et $\text{Pays}(\text{France})$.

B.2.3/ COMPLEXITÉ DE L'INFÉRENCE

Comme étudié précédemment, la complexité du raisonnement est directement dépendante du niveau d'expressivité de la base de connaissances. La complexité d'un raisonnement se définit selon plusieurs classes en temps ou en espace. La complexité en temps correspond à l'évaluation du temps d'exécution de l'algorithme tandis que la complexité en espace correspond à l'espace mémoire occupé par l'exécution de l'algorithme. Pour gagner du temps de calcul, il faut donc utiliser davantage d'espace mémoire, toutefois on s'intéresse davantage à la complexité en temps. D'autre part, les classes de complexité se divisent selon deux types de machine de Turing : les machines de Turing déterministes ou non-déterministes. Une machine de Turing non déterministe est une machine de Turing habituelle, c'est-à-dire déterministe, mais qui peut avoir plusieurs transitions activables, pour un état donné. En conséquence les calculs d'une machine de Turing déterministe forment une suite tandis que ceux d'une machine de Turing non déterministe forment un arbre, dans lequel chaque chemin correspond à une suite de calculs possibles. La complexité des calculs diffère entre les deux approches. Ainsi la complexité d'une classe sur une machine non-déterministe contient la classe correspondante sur une machine déterministe. parmi lesquelles nous citerons :

- **P** : la classe des problèmes qui peuvent être résolus en temps polynomial sur une machine déterministe. Ces problèmes sont souvent considérés comme ceux pour lesquels il existe un algorithme efficace. La complexité de ces problèmes est $O(n^k)$ pour un k donné.
- **NP** : la classe des problèmes qui peuvent être résolus en temps polynomial sur une machine non déterministe.
- **PSpace** : la classe des problèmes qui peuvent être résolus en espace polynomial sur une machine déterministe.

- **ExpTime** : la classe des problèmes qui peuvent être résolus en temps exponentiel sur une machine déterministe.
- **NExpTime** : la classe des problèmes qui peuvent être résolus en temps exponentiel sur une machine non-déterministe.

Type d'inférence	Langages				
	<i>ALC</i>	<i>S</i>	<i>SH - SHIF</i>	<i>SHOIN</i>	<i>SROIQ</i>
Satisfiabilité de concept	PSpace Complet	PSpace Complet	ExpTime Complet	NExpTime Complet	NExpTime Complet
Consistance de la ABox	PSpace Complet	-	ExpTime Complet	NExpTime Complet	NExpTime Complet

VÉRIFICATION DE CONTRAINTES D'INTÉGRITÉ

Sommaire

C.1 Les contraintes de subsomption	177
C.2 Les contraintes domain-range	178
C.3 Les contraintes de participation	179
C.4 Les contraintes de cardinalité	181
C.5 Les contraintes de propriétés	184
C.6 Les contraintes complexes	185

C.1/ LES CONTRAINTES DE SUBSOMPTION

Ce genre de contrainte garantit que certaines relations de superclasse et de sous-classe existent entre instances.

Une capitale doit être une ville :

Contrainte

```
:Capital rdfs:subClassOf :City
```

Base de données A : **invalide**

```
:Paris a :Capital .
```

Base de données B : **valide**

```
:Paris a :Capital , :City .
```

Cette contrainte indique que si un individu RDF est une instance de Capitale, il doit aussi être une instance de Ville. Dans le cas de A, la seule instance de Capitale, à savoir Paris, n'est pas une instance de Ville ; par conséquent, A est non valide. Dans le cas de B, Paris est une instance de Capitale et également de Ville, B est donc valide.

C.2/ LES CONTRAINTES DOMAIN-RANGE

Ces contraintes contrôlent les types de domaine et range des propriétés.

Seules des villes peuvent être la capitale d'un état :

Contrainte

```
:isCapitalOf rdfs:range :States
:isCapitalOf rdfs:domain :City
```

Base de données A : **invalide**

```
:Paris :isCapitalOf :France .
:France a :States .
```

Base de données B : **invalide**

```
:Paris a :City;
      :isCapitalOf :France .
```

Base de données C : **valide**

```
:Paris a :City ;
      :isCapitalOf :France.
:France a :States .
```

Cette contrainte indique que, si deux instances de RDF sont liées entre elles par l'intermédiaire de la propriété *est_Capitale_De*, alors l'instance de range doit être qu'une instance d'Etat et l'instance du domaine doivent être une instance de Ville. Dans la base de données A, il n'y a qu'une seule paire d'individus liés par l'intermédiaire de la relation *est_Capitale_De*, à savoir (Paris, France) avec France une instance d'Etat ; mais Paris n'est pas une instance de Ville. Par conséquent, A est non valide. B, Paris est une instance de Ville, mais France n'est pas une instance d'Etat, B n'est donc pas valide. Dans C, Paris est une instance de Ville, et France est une instance d'Etat, C est donc valide.

Seuls les timeslices peuvent être estampillés :

Contrainte

```
:hasTime rdfs:domain :Timeslice
```

Base de données A : **invalide**

```
:Portugal_TS_1 :hasTime "1990" .
```

Base de données B : **valide**

```
: Portugal_TS_1 a :Timeslice ;
      :hasTime "1990" .
```

Cette contrainte définit que, si une instance de RDF i a une assertion de données via la propriété hasTime, alors i doit être une instance de la classe Timeslice. Dans la base de

données A, *Portugal_TS_1* n'est pas une instance de *Timeslice* mais possède la propriété *hasTime* ; par conséquent, A est non valide. Dans B, *Portugal_TS_1* est une instance de *Timeslice* ; B est donc valide.

La date d'estapillage doit être une date :

Contrainte

```
:hasTime rdfs:range xsd:date
```

Base de données A : **invalide**

```
:Italy_TS_1 :hasTime "1970-01-01" .
```

Base de données B : **valide**

```
: Italy_TS_1 :hasTime "1970-01-01"^^xsd:date
```

Cette contrainte définit que si une instance de RDF *i* est liée à un littéral *l* par l'intermédiaire de la propriété *hasTime*, *l* doit avoir le type de schéma XML XSD *:date*. Dans la base de données A, *Italy_TS_1* est lié à un littéral non typé « 1970-01-01 » par l'intermédiaire de *hasTime*, donc A est non valide. Dans B, le littéral « 1970-01-01 » est correctement typé, donc B est valide.

C.3/ LES CONTRAINTES DE PARTICIPATION

Ces contraintes contrôlent si une instance de RDF participe à une relation spécifiée.

Une ville touristique doit posséder au moins un aéroport :

Contrainte

```
# Constraint in Terp syntax:
# :Touristic_City rdfs:subClassOf (:hasAirport some :Airport)

:Touristic_City rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty :hasAirport ;
    owl:someValuesFrom :Airport
  ] .
```

Base de données A : **valide**

```
: Paris a owl:Thing .
```

Base de données B : **invalide**

```
: Paris a : Touristic_City.
```

Base de données C : **invalide**

```
: Paris a : Touristic_City;
  : hasAirport: Airport_Charles_De_Gaulles.
```


Base de données D : **valide**

```
: Paris a : Touristic_City;
   : hasAirport: Airport_Charles_De_Gaulles.
: Airport_Charles_De_Gaulles a : Airport
```

Cette contrainte définit que si une instance de RDF *i* est de type *Ville_Touristique*, alors *i* doit être associé à un individu *j* via la propriété *possede_aeroport* et aussi que *j* doit être une instance de *Aéroport*. Dans la base de données A, *Ville_Touristique* n'a pas d'instances; par conséquent, A est trivialement valide. Dans B, la seule instance de *Ville_Touristique*, à savoir Paris, n'est liée à aucun individu; par conséquent, B n'est pas valide. Dans C, Paris est liée à *Aeroport_Charles_De_Gaulles* via *possede_aeroport*, mais *Aeroport_Charles_De_Gaulles* n'est pas une instance de *Aéroport*, C est donc invalide. Dans D, Paris est liée à *Aeroport_Charles_De_Gaulles* via *possede_aeroport* et *Aeroport_Charles_De_Gaulles* est une instance de *Aéroport*, D est donc valide.

Chaque ville doit avoir un nombre valide d'habitants :

Contrainte

```
# Constraint in Terp syntax:
# :Village rdfs:subClassOf (:numberCitizen some xsd:integer[>= 0, < 10000])

:Village rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty :numberCitizen ;
    owl:someValuesFrom
      [ a rdfs:Datatype ;
        owl:onDatatype xsd:integer ;
        owl:withRestrictions ([xsd:minInclusive 0] [ xsd:maxExclusive 10000])
      ]
  ] .
```

Base de données A : **valide**

```
:Marcigny a owl:Thing .
```

Base de données B : **invalide**

```
: Marcigny a :Village
```

Base de données C : **invalide**

```
: Marcigny a : Village;
   : numberCitizen "2456" .
```

Base de données D : **invalide**

```
: Marcigny a : Village;
   : numberCitizen "11598"^^xsd:integer .
```

Base de données E : **valide**

```
: Marcigny a : Village;
   : numberCitizen "2456"^^xsd:integer .
```

Cette contrainte définit que si une instance de RDF *i* est de type Village, alors *i* doit être relié via la propriété *nombre_habitant* à un entier compris entre 0 et 10000 (inclusivement), c'est-à-dire les villages possèdent un nombre d'habitants compris dans une certaine gamme. Dans la base de données A, l'individu Marcigny n'est pas connu comme étant une instance de la classe Village donc la contrainte ne s'applique pas et A est valide. Dans B, Marcigny est une instance de Village mais ne possède pas d'assertion via la propriété *nombre_habitant* donc A est non valide. Dans C, Marcigny possède une assertion pour la propriété *nombre_habitant* mais le littéral « 2456 » n'est pas typé, par conséquent, ce n'est pas un nombre entier donc, C est non valide. Dans D, Marcigny est liée à un nombre entier par l'intermédiaire de la propriété *nombre_habitant*, mais il est hors de l'intervalle défini, donc D n'est pas valide. Enfin, en E, Marcigny est lié à l'entier 2456, qui est dans l'intervalle [0,10000], E est valide.

C.4/ LES CONTRAINTES DE CARDINALITÉ

Ces contraintes contrôlent le nombre de certaines relations ou des valeurs de propriété.

Une ville ne peut pas être contenue dans plus d'un pays :

Contrainte

```
# Constraint in Terp syntax:
# :City rdfs:subClassOf (:within max 1 :States)

:City rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty :within;
    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onClass :States
  ] .
```

Base de données A : **valide**

```
:Paris a owl:Thing.
```

Base de données B : **valide**

```
:Paris a :City ;
  :within :France .
:France a :States .
```

Base de données C : **invalide**

```
:Paris a :City ;
  :within :France , :United-States.
: France a :States.
: United-States a :States.
```

Si une instance de RDF *i* est une ville, alors *i* ne doit pas être liée par l'intermédiaire de la propriété *contenue_dans* à plus de 1 instance de la classe Pays. Dans la base de données A, Paris n'est pas connue comme une instance de la classe Ville donc la contrainte ne s'applique pas, par conséquent, A est valide. Dans B, Paris est une instance de Ville et est connue pour être contenue dans un seul pays, à savoir la France, donc B est valide. Dans C, Paris est liée à deux pays par l'intermédiaire de la propriété *contenue_dans*.

Une région doit contenir au moins 2 départements :

Contrainte

```
# Constraint in Terp syntax:
# :Region rdfs:subClassOf (contains min 2 :Department)

:Region rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty :contains ;
    owl:minQualifiedCardinality "2"^^xsd:nonNegativeInteger ;
    owl:onClass :Department
  ] .
```

Base de données A : **valide**

```
:Burgundy a owl:NamedIndividual .
```

Base de données B : **invalide**

```
:Burgundy a :Region ;
:contains : Côte_d'or.
:Côte_d'or a :Department .
```

Base de données C : **valide**

```
:Burgundy a :Region ;
:contains : Côte_d'or;
:contains : Côte_d'or.
:Côte_d'or a :Department.
:Yonne a :Department.
```

Cette contrainte définit que si une instance de RDF *i* est une Région, alors il doit exister au moins 2 instances *j* et *k* de la classe Département qui sont liées à *i* par l'intermédiaire de la propriété Contient. Dans la base de données A, Bourgogne n'est pas connue comme une instance de la classe Région donc la contrainte ne s'applique pas. Dans B, Bourgogne est une instance de Bourgogne, mais une seule instance de Département, à savoir *Cote_d'Or*, est connue comme contenue dans Bourgogne, donc B n'est pas valide. Dans C, Bourgogne est liée aux individus *Cote_d'Or* et Yonne, qui sont deux instances de Département et sont considérées comme distinctes, donc C est valide.

Un timeslice doit être représenté par exactement une géométrie :

Contrainte

```
# Constraint in Terp syntax:
# :Timeslice rdfs:subClassOf (:hasGeometry exactly 1 :Geometry)

:Timeslice rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty :hasGeometry ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onClass :Geometry
  ] .
```

Base de données A : **valide**

```
Individual: Madrid_TS1
```

Base de données B : **invalide**

```
: Madrid_TS1 a :Timeslice .
```

Base de données C : **invalide**

```
: Madrid_TS1 a :Timeslice;
  :hasGeometry :Geom1 .
```

Base de données D : **valide**

```
: Madrid_TS1 a :Timeslice;
  :hasGeometry :Geom1.
:MyDepartment a :Geometry .
```

Base de données E : **invalide**

```
: Madrid_TS1 a :Timeslice;
  :hasGeometry :Geom1, :Geom2.
:Geom1 a :Geometry .
:Geom2 a :Geometry.
```

Cette contrainte définit que si une instance de RDF, *i* est un *Timeslice*, alors elle doit se rapporter à exactement 1 instance de la classe *Géométrie* via la propriété *possede_Geometrie*. Dans la base de données A, l'individu *Madrid_TS1* n'est pas connu comme étant une instance de *Timeslice* donc la contrainte ne s'applique pas et A est valide. Dans B, *Madrid_TS1* est une instance de *Timeslice*, mais n'est lié à aucune *Géométrie* donc B n'est pas valide. Dans C, *Madrid_TS1* est liée à l'individu *Geom1* via la propriété *possede_Geometrie* mais *Geom1* n'est pas connue pour être une instance de la classe *Géométrie*, donc C est non valide. Dans D, *Madrid_TS1* est liée à exactement une seule instance de *Géométrie*, à savoir *Geom1*, donc D est valide. Enfin, dans E, *Madrid_TS1* est liée à deux instances distinctes de la classe *Géométrie*, à savoir *Geom1* et *Geom2*, donc E n'est pas valide.

Les entités ne peuvent pas avoir plus d'une relation de continuation :

Contrainte

```
:name a owl:FunctionalProperty .
```

Base de données A : **valide**

```
:Madrid_TS1 a owl:Thing .
```

Base de données B : **valide**

```
: Madrid_TS1 :hasContinuation Madrid_TS2 .
```

Base de données C : **invalide**

```
: Madrid_TS1 :hasContinuation Madrid_TS2, Madrid_TS2_bis.
```

Cette contrainte définit qu'aucune instance de RDF *i* ne peut avoir plus de 1 assertion via la relation *hasContinuation*. Dans la base de données A, *Madrid_TS1* ne possède pas d'assertions pour la propriété *hasContinuation* donc A est valide. Dans B, *Madrid_TS1* possède une seule assertion pour la propriété, l'ontologie est donc aussi valable. Dans C, *Madrid_TS1* possède deux fois la relation de continuation, donc C est non valide.

C.5/ LES CONTRAINTES DE PROPRIÉTÉS

Ces contraintes contrôlent comment les instances sont liées à un autre via des propriétés.

Un timeslice lié dans le temps par une relation de filiation à un autre timeslice dont il est issu doit posséder une partie de sa géométrie qui s'intersecte avec son prédécesseur : cette intersection se traduit par une relation de filiation spatiale. Une relation de filiation spatiale s'établit entre deux timeslices qui s'intersectent à différentes époques, toutefois cette relation n'a pas valeur de vérité dans tous les cas de filiation :

Contrainte

```
:hasFiliation rdfs:subPropertyOf :hasSpatialFiliation .
```

Base de données A : **invalide**

```
:Madrid_TS1 :hasFiliation :Madrid_TS2
```

Base de données B : **valide**

```
:Madrid_TS1 :hasSpatialFiliation :Madrid_TS2;
  :hasFiliation :Madrid_TS2.
```

Cette contrainte définit que si une instance de RDF *i* est liée à *j* via la propriété *hasSpatialFiliation*, alors *i* doit aussi être associé à *j* via la propriété *hasFiliation*. Dans la base de données A, *Madrid_TS1* est lié à *Madrid_TS2* via la propriété *hasSpatialFiliation* mais pas par l'intermédiaire de *hasFiliation*, donc A est non valide. Dans B, *Madrid_TS1* est lié à *Madrid_TS2* via les deux propriétés, donc B est valide.

Le conseil général doit s'occuper de l'entretien des routes départementales de son département :

Contrainte

```
:maintain owl:propertyChainAxiom (:manages [owl:inverseOf :within]) .
```

Base de données A : **invalide**

```
:General_Council_Côte_d'Or :manages :Côte_d'Or ;
  :maintain :D28 .
:D28 :within :Côte_d'Or.
:D14 :within :Côte_d'Or.
```

Base de données B : **valide**

```
:General_Council_Côte_d'Or :manages :Côte_d'Or ;
  :maintain :D28, D14 .
:D28 :within :Côte_d'Or.
:D14 :within :Côte_d'Or.
```

Cette contrainte définit que si une instance de RDF *i* est liée à *j* via la propriété *Gère* et *k* est lié à *j* via la propriété *est_contenue_dans*, alors *i* doit être reliée à *k* par l'intermédiaire de la propriété *Entretien*. Dans la base de données A, *Conseil_General_Cote_d'Or* est lié à *Cote_d'Or* via *Gère*, *D14* est liée à *Cote_d'Or* via *est_contenue_dans*, mais

Conseil_General_Cote_d'Or n'est pas lié à D14 par l'intermédiaire d'aucune propriété, donc A est non valide. Dans B, *Conseil_General_Cote_d'Or* est lié à D28 et D14 — qui sont tous deux contenus dans *Cote_d'Or* via *est_contenue_dans* — via la propriété *Entretien*, donc B est valide.

C.6/ LES CONTRAINTES COMPLEXES

Ces contraintes sont plus complexes, incluant souvent plusieurs conditions.

Chaque ville possède au moins une gare, ou se situe à proximité d'une ville qui possède au moins une gare, ou possède un aéroport :

Contrainte

```
# Constraint in Terp syntax:
# :City rdfs:subClassOf (:hasRailwayStation some (:RailwayStation or (:isLocatedNearby some (:City and
(:hasRailwayStation some :RailwayStation))) or (:hasAirport some :Airport)))

:City rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty :hasRailwayStation ;
    owl:someValuesFrom
      [ owl:unionOf (:RailwayStation
        [ a owl:Restriction ;
          owl:onProperty :isLocatedNearby ;
          owl:someValuesFrom
            [ owl:intersectionOf (:City
              [ a owl:Restriction ;
                owl:onProperty :hasRailwayStation;
                owl:someValuesFrom :RailwayStation;
              ])
            ])
          ]
        ]
      ]
    ]
  [ a owl:Restriction ;
    owl:onProperty :hasAirport ;
    owl:someValuesFrom :Airport
  ]
] .
```

Base de données A : **invalide**

```
:Paris a :City .
```

Base de données B : **invalide**

```
:Nozay a :City ;
  :isLocatedNearby :Nozay .
:Paris a :City .
```

Base de données C : **valide**

```
:Nozay a :City ;
  :isLocatedNearby :Paris .
:Paris a :City ;
  :hasRailwayStation :Gare_de_Lyon .
:Gare_de_Lyon a :RailwayStation.
```

Base de données D : **valide**

```
:Paris a :City ;
    :hasAirport :Airport .
:Airport_Charles_De_Gaulles a :Airport .
```

Base de données E : **valide**

```
:Paris a :City ;
    :hasAirport :Airport_Charles_De_Gaulles ;
    :hasRailwayStation :Gare_de_Lyon .
:Airport_Charles_De_Gaulles a :Airport .
:Gare_de_Lyon a :RailwayStation .
```

Cette contrainte définit que si un individu *i* est une instance de la classe Ville, alors au moins une des trois conditions doit être remplie :

- Il est lié à une instance de la classe *Gare_ferroviaire* via la propriété *hasRailwayStation*,
- il est lié à une instance *j* par l'intermédiaire de la propriété *isLocatedNearby* et *j* est une instance de la classe Ville qui est associée à une instance de *Gare_ferroviaire* via la propriété *hasRailwayStation*,
- il est lié à une instance de la classe Aéroport via la propriété *hasAirport*.

A et B ne sont pas valides car aucune des conditions n'est remplie. C répond à la deuxième condition : Nozay (qui est une ville) est reliée à Paris par l'intermédiaire de la relation *isLocatedNearby*, avec Paris défini comme une Ville associée à *Gare_De_Lyon*, qui est une gare ferroviaire, par l'intermédiaire de la propriété *hasRailwayStation*. D répond à la troisième condition : Paris est liée à une instance de la classe Aéroport, à savoir *Airport_Charles_De_Gaulles*, par l'intermédiaire de la propriété *hasAirport*. Enfin, E réunit la première et la troisième condition parce qu'en plus de posséder un aéroport, Paris est également liée une instance de la gare ferroviaire, à savoir *Gare_De_Lyon*, par l'intermédiaire de la propriété *hasRailwayStation*.

Seules des personnes étant des citoyens européens peuvent circuler librement sur un territoire qui se situe en Europe :

Contrainte

```
# Constraint in Terp syntax:
# :Country and (:within some :European_States)) rdfs:subClassOf (inverse :circulate_freely
only (:People and (:nationality value "UE")))

[ owl:intersectionOf (:Country
    [ a owl:Restriction ;
      owl:onProperty :within ;
      owl:someValuesFrom :European_States
    ]) .
] rdfs:subClassOf
    [ a owl:Restriction ;
      owl:onProperty [owl:inverseOf :circulate_freely] ;
      owl:allValuesFrom [ owl:intersectionOf (:People
          [ a owl:Restriction ;
            owl:hasValue "UE" ;
            owl:onProperty :nationality
          ])
        ] .
    ] .
```

Base de données A : **valide**

```
:Rome a :Country ;
    :within :Italy .
:Italy a :European_States
```

Base de données B : **invalide**

```
:Rome a :Country ;
    :within :Italy .
:Italy a :European_States
:Mario a :People ;
    :circulate_freely :Rome .
```

Base de données C : **valide**

```
:Rome a :Country ;
    :within :Italy .
:Italy a :European_States
:Mario a :People ;
    :circulate_freely :Rome ;
    :nationality "UE" .
```

Base de données D : **invalide**

```
:Rome a :Country ;
    :within :Italy .
:Italy a :European_States
:Mario a :People ;
    :circulate_freely :Rome ;
    :nationality "UE" .
:Françoise a :Citizen ;
    :circulate_freely :Rome ;
    :nationality "UE" .
```

Base de données E : **valide**

```
:Rome a :Country ;
    :within :Italy .
:Italy a :European_States
:Mario a :People ;
    :circulate_freely :Rome ;
    :nationality "UE" .
:Françoise a :Citizen;
    :circulate_freely :Rome ;
    :nationality "UE" .
:Citizen rdfs:subClassOf :People .
    SubClassOf:People
```

Cette contrainte indique que si une personne *i* est une instance de *Country* et qu'elle est liée à une instance de *European_States* par l'intermédiaire de la propriété *Within*, alors tout individu *j* qui est lié à *i* par l'intermédiaire de la propriété *circulate_freely* doit satisfaire à deux conditions :

- il doit être une instance de *Country*,
- il ne doit pas être lié à n'importe quel littéral autre que "UE" par l'intermédiaire de la propriété *nationality*.

A est valide car qu'il n'y a aucun individu lié à *Country* par *Within*, donc la contrainte est trivialement satisfaite. B n'est pas valide car *Mario* est lié à *Rome* par l'intermédiaire de la propriété *circulate_freely*, *Rome* est une instance de *Country* et est liée à une instance de *European_States*, à savoir l'Italie, via la relation *Within*, mais *Mario* n'a aucune assertion concernant la propriété *Nationality*. C est valide parce que les deux conditions sont remplies. D n'est pas valide parce que *Françoise* a violé la première condition : elle est liée à *Rome* par l'intermédiaire de *circulate_freely*, mais n'est pas connue comme étant une instance de *People*. Enfin, ce problème est corrigé dans E, au moyen d'un axiome OWL, qui stipule que chaque instance de *Citizen* est une instance de *People*, donc *Françoise* est inférée comme étant une instance de *People*, en conséquence, E est valide.

RAISONNEMENTS D'INFÉRENCES EN OWL

Ce chapitre présente différents exemples d'inférence trouvant leur application dans le domaine de l'information géo-spatiale. Ces exemples sont décrits en OWL 2 DL et traduit en logique de description $\mathcal{SROIQ}(\mathcal{D})$. Les concepts de la logique de description sont assimilés à des classes en OWL, de même que les rôles sont assimilés à des propriétés en OWL. Les raisonnements abordés dans ce rapport concernent uniquement les inférences permettant de générer de l'information, les inférences liées à la détection d'inconsistance, incohérence ou insatisfiabilité ne sont pas traitées ici.

Sommaire

D.1	Présentation de OWL 2 DL	189
D.2	Les règles d'inférence de OWL Horst	191
D.2.1	Règles RDFS	191
D.2.2	Règles OWL	192
D.3	Raisonnements sur la TBOX	192
D.3.1	Inférence sur les classes	194
D.3.2	Inférence sur les propriétés	195
D.4	Raisonnements sur la ABOX	196
D.4.1	Inférence d'instance de classe	197
D.4.2	Inférence d'instances de propriétés	200

D.1/ PRÉSENTATION DE OWL 2 DL

OWL 2 DL est très proche de la logique de description $\mathcal{SROIQ}(\mathcal{D})$. D'autre part, la logique de description $\mathcal{SHOIN}(\mathcal{D})$, propre à OWL DL, s'avère être un support difficile de raisonnement pour des problèmes d'inférence standard tels que la vérification de la consistance de l'ontologie, la satisfiabilité de classes, l'expression de la subsumption au sein d'une hiérarchie de classe ou encore la vérification d'instance. Tous ces tests d'inférence correspondent à une complexité $2NExpTime$. Afin de traiter ces problèmes de complexité pour les différentes inférences énoncées, trois profils OWL ont été définis possédant chacun de meilleures propriétés de calculs et des complexités moindres pour résoudre les problèmes d'inférence :

- **OWL 2 - EL (Existential Languages)** : Ce fragment de langage est basé sur $\mathcal{EL}++$, une logique de description pour laquelle les problèmes de raisonnement standard peuvent être résolus en temps polynomial en fonction de la taille de l'ontologie. Ce profil possède une expressivité particulièrement utile dans des applications utilisant des ontologies contenant un grand nombre de propriétés et/ou de classes. Contrairement aux deux autres profils, *EL* possède l'avantage d'être symétrique au sens où les restrictions s'appliquent aussi bien sur la TBox que la ABox. Ce profil est ainsi très simple à définir et utiliser. Cependant, les restrictions de classe excluent l'usage du quantificateur universel, des cardinalités, de la disjonction, de la négation, de l'énumération impliquant plusieurs individus et la plupart des propriétés caractéristiques telles que les propriétés inverses, disjointes, irreflexives, fonctionnelles, inversement fonctionnelles, symétriques et asymétriques. De plus, les types de données suivants pour les propriétés « datatypes » sont également exclus de ce profil : *double*, *float*, *nonPositiveInteger*, *PositiveInteger*, *negativeInteger*, *long*, *int*, *short*, *byte*, *unsignedLong*, *unsignedInt*, *unsignedshort*, *unsignedByte*, *language* et *boolean*. Enfin, le profil *EL* interdit l'utilisation d'individus anonymes. Plusieurs raisonneurs sont disponibles pour le profil *EL* parmi lesquels CB [Kazakov et al., 2009], CEL [Baader et al., 2006], Snorocket [ter Horst, 2005] et Pellet (Pellet supporte tous les profils OWL 2 mais propose un raisonneur dédié au profil *EL* afin d'optimiser les raisonnements) [Sirin et al., 2007].
- **OWL 2 – QL (Query Language)** : Ce fragment est basé sur DL-Lite, une logique de description pouvant être implémentée au sein de système de gestion de base de données relationnelles conventionnelles et calculée avec une complexité *LOGSPACE* dépendante de la taille des données (individus). Ce profil est destiné aux applications utilisant un très grand volume d'instances et dont le requêtage est le principal raisonnement à appliquer. De même que en *OWL 2 EL*, des algorithmes en temps polynomial peuvent être utilisés pour vérifier la consistance de l'ontologie ou bien appliquer des raisonnements relatifs à la subsomption. L'expressivité de *OWL 2 QL* est volontairement limitée bien que les caractéristiques principales des modèles conceptuels tels que les diagrammes de classe UML ou les diagrammes entité/relation sont inclus dans ce profil. *OWL 2 QL* étant basé sur DL-Lite, et bien que les instances soient admises au sein d'une base de données relationnelles, l'hypothèse du nom unique n'est pas nécessaire dans ce profil, c'est pourquoi l'hypothèse du nom unique n'a pas de conséquences sémantiques sur l'ontologie. De plus, *OWL 2 QL* ne permet pas seulement des restrictions sur les classes mais permet également de définir des restrictions sur la sous-classe ou la superclasse d'un axiome *subClassOf*. Par conséquent, la définition précise du profil devient relativement complexe. L'ensemble des types de données supportés par le profil est le même que pour le profil *EL*, de même que l'usage d'individus anonymes est également prohibé. Enfin, *OWL 2 QL* ne permet pas la définition de l'égalité entre individus (*SameIndividual*) car il ne serait plus possible alors d'écrire les requêtes en logique du premier ordre et d'implémenter ce profil en utilisant directement une base de données relationnelle. Plusieurs raisonneurs sont disponibles pour ce profil parmi lesquels OwlGres (la version de Pellet dédiée au profil *QL*) et QuOnto. Ces deux raisonneurs sont basées sur des techniques de réécriture de requête qui transforment une requête conjonctive sur l'ontologie en un ensemble de requêtes sur les individus (la ABox) puis, dans un dernier temps (via des mappages de noms de classe et de propriétés de ontologie avec des requêtes SQL) en des requêtes

SQL sur une base de données où les données d'instance sont stockées.

- **OWL 2 – RL (Rule Language)** : Ce profil est basé sur la logique de description programmable (Description Logic Programs) [Grosz et al., 2003] est destiné aux applications nécessitant un passage à l'échelle des capacités de raisonnement sans sacrifier une trop grande expressivité au sein de l'ontologie. Ce profil tente de concilier à la fois les applications OWL 2 qui souhaitent profiter de toute l'expressivité du langage avec une certaine efficacité et pour les applications RDF(S) nécessitant d'augmenter l'expressivité en utilisant certains axiomes OWL 2. Pour cela, un sous-ensemble syntaxique de *OWL 2*, soumis à l'usage de technologies basées sur les règles, a été défini. Comme *OWL 2 QL*, la syntaxe de *RL* est asymétrique en ce sens que différentes contraintes s'appliquent aux expressions de classe selon l'endroit où ils se produisent sur les axiomes de l'ontologie. De manière générale, cela signifie que sont autorisés : les énumérations (*ObjectOneOf*), les intersections (*ObjectIntersectionOf*), les unions (*ObjectUnionOf*) et les restrictions existentielles dans la position de la sous-classe d'un axiome *subClassOf*, et l'intersection (*ObjectIntersectionOf*), la négation (*ObjectComplementOf*), les restrictions universelles (*ObjectAllValuesFrom* et *DataAllValuesFrom*), les restrictions existentielles utilisant un individu ou une valeur de donnée (*ObjectHasValue* et *DataHasValue*) et les restrictions de cardinalité « au plus » 0/1 (*ObjectMaxCardinality* et *DataMaxCardinality* avec les valeurs 0 ou 1) dans la position d'une superclasse d'un axiome *subClassOf*. Plusieurs raisonneurs sont disponibles pour le profil *OWL 2 RL* parmi lesquels Elly, Jena et Oracle Database 11g OWL Reasoner. Toutes ces implémentations sont basées sur des règles s'appliquant sur des triple stores et des bases de données relationnelles. Ces règles calculent toutes les inférences pertinentes et les matérialisent au sein de la base de données ou du triple store. Le calcul d'inférence peut nécessiter beaucoup de temps et d'espace de stockage, cependant dans des conditions acceptables de calcul, il est possible, après ce calcul d'inférence, de fournir une réponse aux requêtes simplement en interrogeant le triple store ou la base de données. L'exhaustivité de la réponse est, toutefois, dépendante de l'ensemble des déductions matérialisées et de la requête à satisfaire car seules les requêtes s'appliquant sur les ontologies satisfaisant les restrictions syntaxiques du profil *RL* garantissent un résultat correct.

D.2/ LES RÈGLES D'INFÉRENCE DE OWL HORST

Les exemples présentés dans la suite de ce rapport sont issus d'un ensemble de règles d'implications définies pour RDFS et OWL. Ces ensembles de règles sont identifiés dans [ter Horst, 2005]. Certaines règles ne présentent pas ou peu de pertinence dans le cadre des travaux de modélisation spatiale. En reprenant les tableaux récapitulatifs des règles de Horst fourni par Rafael Peixoto (doctorant dans l'équipe Checksem) dans son rapport *Web-Scale Reasoning*.

D.2.1/ RÈGLES RDFS

L'inférence en RDFS est possible au travers de règles d'implication. L'implication est la relation entre deux déclarations où la vérité d'une déclaration (A) nécessite la vérité de la

déclaration (B). Le tableau 1 montre les règles d'implication RDFS identifiés par un nom et les deux phrases de la règle d'implication.

TABLE D.1 – Règle d'implication RDFS

Nom	Si	alors
1	$s \text{ p o}$ (if o is a literal)	$o \text{ rdf:type rdfs:Literal}$
2	$p \text{ rdfs:domain } x \ \& \ s \text{ p o}$	$s \text{ rdf:type } x$
3	$p \text{ rdfs:range } x \ \& \ s \text{ p o}$	$o \text{ rdf:type } x$
4a	$s \text{ p o}$	$s \text{ rdf:type rdfs:Resource}$
4b	$s \text{ p o}$	$o \text{ rdf:type rdfs:Resource}$
5	$p \text{ rdfs:subPropertyOf } q \ \& \ q \text{ rdfs:subPropertyOf } r$	$p \text{ rdfs:subPropertyOf } r$
6	$p \text{ rdf:type rdf:Property}$	$p \text{ rdfs:subPropertyOf } p$
7	$s \text{ p o} \ \& \ p \text{ rdfs:subPropertyOf } q$	$s \text{ q o}$
8	$s \text{ rdf:type rdfs:Class}$	$s \text{ rdfs:subClassOf rdfs:Resource}$
9	$s \text{ rdf:type } x \ \& \ x \text{ rdfs:subClassOf } y$	$s \text{ rdf:type } y$
10	$s \text{ rdf:type rdfs:Class}$	$s \text{ rdfs:subClassOf } s$
11	$x \text{ rdfs:subClassOf } y \ \& \ y \text{ rdfs:subClassOf } z$	$x \text{ rdfs:subClassOf } z$
12	$p \text{ rdf:type rdfs:ContainerMembershipProperty}$	$p \text{ rdfs:subPropertyOf rdfs:member}$
13	$o \text{ rdf:type rdfs:Datatype}$	$o \text{ rdfs:subClassOf rdfs:Literal}$

D.2.2/ RÈGLES OWL

OWL-Horst est un fragment d'OWL qui définit des extensions sémantiques de RDFS impliquant des types de données et un sous-ensemble du vocabulaire OWL qui comprend un vocabulaire lié aux propriétés, des comparaisons et des restrictions de valeur. L'ensemble de règles d'implication OWL Horst comprend les règles RDFS présentées dans le tableau D.1 et les règles définies dans [ter Horst, 2005] et présentées dans le tableau D.2.

D.3/ RAISONNEMENTS SUR LA TBOX

La TBox décrit les connaissances générales d'un domaine au travers d'un ensemble de classes et propriétés. Il existe quatre principaux problèmes d'inférences au niveau terminologique :

- **Satisfiabilité** : Un concept C d'une terminologie \mathcal{T} est satisfiable si et seulement si il existe un modèle I de \mathcal{T} tel que $C^I \neq \emptyset$.
- **Subsumption** : Un concept C est subsumé par un concept D pour une terminologie si et seulement si $C^I \subseteq D^I$ pour tout modèle I de \mathcal{T} .
- **Équivalence** : Un concept C est équivalent à un concept D pour une terminologie et seulement si $C^I = D^I$ pour chaque modèle I de \mathcal{T} .
- **Disjonction** : Des concepts C et D sont disjoints par rapport terminologie si et seulement si $C^I \cap D^I = \emptyset$; pour chaque modèle I de \mathcal{T} .

Dans la suite de cette section, les relations inférées sont logiquement des relations de subsumption, équivalence ou disjonction entre des classes ou des propriétés. Le test de satisfiabilité étant lié à la détection d'erreur.

TABLE D.2 – Ensemble de règles d'implication OWL Horst

Nom	Si	Alors	Notes
1	p rdf:type owl:FunctionalProperty . u p v . u p w .	v owl:sameAs w .	Functional property
2	p rdf:type owl:InverseFunctionalProperty . u p w . v p w .	u owl:sameAs v	Inverse functional property
3	p rdf:type owl:SymmetricProperty . v p w .	w p v .	Symmetric property
4	p rdf:type owl:TransitiveProperty . u p v . v p w .	u p w .	Transitive property
5a	v p w .	v owl:sameAs v .	Self-evident sameAs
5b	v p w .	w owl:sameAs w .	Self-evident sameAs
6	v owl:sameAs w .	w owl:sameAs v .	Reflexive sameAs
7	u owl:sameAs v . v owl:sameAs w .	u owl:sameAs w .	Transitive sameAs
8ax	p owl:inverseOf q . v p w .	w q v .	Inverse of property
8bx	p owl:inverseOf q . v q w .	w p v .	Inverse of property
9	v rdf:type owl:Class . v owl:sameAs w .	v rdfs:subClassOf w .	Subsumption on same classes
10	p rdf:type owl:Property . p owl:sameAs q .	p rdfs:subPropertyOf q .	subsumption on same properties
11	u p v .	u' p v' .	Property extension through sameAs
12a	u owl:sameAs u' . v owl:sameAs v' . v owl:equivalentClass w .	v rdfs:subClassOf w .	Subsumption on Equivalent-Class
12b	v owl:equivalentClass w .	w rdfs:subClassOf v .	Equivalency from subsumption classes
12c	v rdfs:subClassOf w .	v rdfs:equivalentClass w .	Equivalency from subsumption classes
13a	w rdfs:subClassOf v . v owl:equivalentProperty w .	v rdfs:subPropertyOf w .	Subsumption on Equivalent property
13b	v owl:equivalentProperty w .	w rdfs:subPropertyOf v .	Subsumption on Equivalent property
13c	v rdfs:subPropertyOf w .	v rdfs:equivalentProperty w .	Equivalency from subsumption properties
14a	w rdfs:subPropertyOf v . v owl:hasValue w . v owl:onProperty p . u p w .	u rdf:type v .	Filler restriction entailment
14bx	v owl:hasValue w . v owl:onProperty p . u rdf:type v .	u p w .	HasValue definition
15	v owl:someValuesFrom w . v owl:onProperty p . u p x .	u rdf:type v .	full existential restriction entailment
16	x rdf:type w . v owl:allValuesFrom w . v owl:onProperty p . u rdf:type v . u p x .	x rdf:type w .	allvaluesfrom definition

D.3.1/ INFÉRENCE SUR LES CLASSES

D.3.1.1/ HÉRITAGE DE LA DISJONCTION AVEC LE COMPLÉMENT D'UNE SUPER-CLASSE

Toutes les sous-classes d'une classe donnée sont disjointes avec le complément de la classe.

```
(rdfs:subClassOf Downtown Urban)
(owl:complementOf Urban Non-Urban)
Implique que
(owl:disjointWith Downtown Non-urban)
```

Logique de description :

TBox	ABox	Implication
$(Downtown \sqsubseteq Urban)$ $(Urban \equiv \neg Non - Urban)$		$(Downtown \sqcap Non - Urban \sqsubseteq \perp)$

D.3.1.2/ HÉRITAGE DE LA CONTRAINTE DE DISJONCTION

Etant donné une classe A définie comme disjointe d'une classe B alors toutes les sous-classes de B héritent de la disjonction avec la classe A. Par exemple, si la classe « Urbain » est définie comme disjointe de la classe « Rural » et la classe « Territoire Agricole » est définie comme sous-classe de « Rural » alors il est possible d'inférer que la classe « urbain » est disjointe de la classe « Territoire Agricole ».

```
(owl:disjointWith Urban CountrySide)
(rdfs:subClassOf Agricultural_Land CountrySide)
Implique que
(owl:disjointWith Urban Agricultural_Land)
```

Logique de description :

TBox	ABox	Implication
$(Urban \sqcap CountrySide \sqsubseteq \perp)$ $(Agricultural_Land \sqsubseteq CountrySide)$		$(Urban \sqcap CountrySide \sqsubseteq \perp)$

D.3.1.3/ OWL :EQUIVALENTCLASS

Quand la propriété owl :sameAs est définie ou inférée entre deux entités qui sont connues comme étant des classes, la relation owl :equivalentClass est inférée entre ces classes.

```
(rdf:type City rdfs:Class)
(rdf:type Town rdfs:Class)
(owl:sameAs City Town)
Implique que
(owl:EquivalentClass City Town)
```

Logique de description :

La transitivité via rdfs :subClassOf et rdfs :subPropertyOf :

TBox	ABox	Implication
(<i>City</i> \doteq <i>Town</i>) (<i>City</i> : <i>Class</i>) (<i>Town</i> : <i>Class</i>)		(<i>City</i> \equiv <i>Town</i>)

```
(rdfs:subClassOf Division Province)
(rdfs:subClassOf Province Geographical_Region)
Implique que
(rdfs:subClassOf Division Geographical_Region)
```

Logique de description :

TBox	ABox	Implication
(<i>Division</i> \sqsubseteq <i>Province</i>) (<i>Province</i> \sqsubseteq <i>Geographical_Region</i>)		(<i>Division</i> \sqsubseteq <i>Geographical_Region</i>)

D.3.2/ INFÉRENCE SUR LES PROPRIÉTÉS

D.3.2.1/ TRANSIVITÉ ET RELATION INVERSE

Si une propriété Q est owl :inverseOf d'une propriété P, et que P est une owl :TransitiveProperty, alors Q est aussi une owl :TransitiveProperty. Par exemple, la relation « contient » est définie comme l'inverse de la relation « est contenu dans ». La relation « est contenu dans » est définie comme transitive par conséquent, la relation « contient » est inférée comme transitive également.

```
(owl:inverseOf Contains Within)
(rdf:type Within owl:TransitiveProperty)
Implique que
(rdf:type Contains owl:TransitiveProperty)
```

Logique de description :

TBox	ABox	Implication
(<i>Contains</i> \equiv <i>Within</i> ⁻) <i>Tr</i> (<i>Within</i>)		<i>Tr</i> (<i>Contains</i>)

D.3.2.2/ HÉRITAGE DES PROPRIÉTÉS SUBSUMÉES

Etant donné, une propriété P1 définie comme sous-propriété d'une propriété P2, elle-même définie comme sous-propriété d'une propriété P3 alors P1 est une sous-propriété de P3. Par exemple, la relation « Touche » est une sous-propriété de la relation « Relation topologique » qui elle-même est une sous-propriété de la relation « Relation spatiale » par conséquent, la relation « Touche » est une sous-propriété de « relation spatiale ».

```
(rdfs:subPropertyOf touches TopologicalRelationship)
(rdfs:subPropertyOf TopologicalRelationship SpatialRelationship)
Implique que
(rdfs:subPropertyOf touches SpatialRelationship)
```


Logique de description :

TBox	ABox	Implication
$(\textit{touches} \sqsubseteq \textit{TopologicalRelationship})$ $(\textit{TopologicalRelationship} \sqsubseteq \textit{SpatialRelationship})$		$(\textit{touches} \sqsubseteq \textit{SpatialRelationship})$

D.3.2.3/ OWL :EQUIVALENTPROPERTY

Quand la propriété owl :sameAs est définie ou inférée entre deux entités qui sont connus comme étant des propriétés, la relation owl :equivalentProperty est inférée entre les propriétés.

```
(rdf:type Adjacent rdf:Property)
(rdf:type Neighboring rdf:Property)
(owl:sameAs Adjacent Neighboring)
Implique que
(owl:EquivalentClass Adjacent Neighboring)
```

Logique de description :

TBox	ABox	Implication
$(\textit{Adjacent} \equiv \textit{Neighboring})$ $(\textit{Adjacent} : \textit{Property})$ $(\textit{Neighboring} : \textit{Property})$		$(\textit{Adjacent} \equiv \textit{Neighboring})$

D.4/ RAISONNEMENTS SUR LA ABOX

Une ABox contient un ensemble d'assertions sur les individus : des assertions d'appartenance (Classes) et des assertions de rôle (Propriétés). Chaque ABox doit être associée à une TBox, car les assertions s'expriment en fonction des concepts et des rôles de la TBox. Le niveau factuel comprend quatre principaux problèmes d'inférence :

- **Cohérence** : Une ABox \mathcal{A} est cohérente par rapport à une TBox \mathcal{T} si et seulement s'il existe un modèle \mathcal{I} de \mathcal{A} et \mathcal{T} .
- **Vérification d'instance** : Vérifier par inférence si une assertion $C(a)$ est vraie pour tout modèle \mathcal{I} d'une ABox \mathcal{A} et d'une TBox \mathcal{T} .
- **Vérification de rôle** : Vérifier par inférence si une assertion $\mathcal{R}(a, b)$ est vraie pour tout modèle \mathcal{I} d'une ABox \mathcal{A} et d'une TBox \mathcal{T} .
- **Problème de récupération** : Pour une ABox \mathcal{A} , un concept C d'une terminologie \mathcal{T} , inférer les individus $a_{\mathcal{I}_1} \dots a_{\mathcal{I}_n} \in C^{\mathcal{I}}$ pour tout modèle \mathcal{I} de \mathcal{T} .

Les trois premiers types d'inférences factuelles énoncés étant liés au domaine de la détection d'erreur dans une base de connaissance, cette section montre uniquement des exemples liés au problème de récupération qui infère des relations d'instanciations entre des individus et des classes.

D.4.1/ INFÉRENCE D'INSTANCE DE CLASSE

Cette partie met en évidence les différents axiomes pouvant servir de support à l'inférence d'assertions d'appartenance.

D.4.1.1/ L'HÉRITAGE VIA RDFS :SUBCLASSOF

Etant donné, un individu i_1 comme instance d'une classe C_1 et C_1 la sous-classe de la classe C_2 alors i_1 est également une instance de la classe C_2 . Par exemple, si le « lac Michigan » est défini comme instance de la classe « Lac » et que « Lac » est la sous-classe de la classe « étendue d'eau » alors le « Lac Michigan » est une instance de la classe « étendue d'eau ».

```
(rdf:type Lake_Michigan Lake)
(rdfs:subClassOf Lake WaterBody)
Implique que
(rdf:type Lake_Michigan WaterBody)
```

Logique de description :

TBox	ABox	Implication
$(Lake \sqsubseteq WaterBody)$	$Lake(Lake_Michigan)$	$WaterBody(Lake_Michigan)$

D.4.1.2/ LES CONTRAINTES RDFS :DOMAIN ET RDFS :RANGE

Etant donné une relation R définie avec comme Domain une classe A et comme Range une classe B . Si $R(i_1, i_2)$ avec i_1 et i_2 des individus alors $A(i_1)$ et $B(i_2)$. Par exemple, la propriété « est capital de » est définie avec pour Domain la classe « Ville » et pour Range la classe « Pays ». En outre, « Paris » est définie comme « la capitale de » la « France ». Par conséquent, il est possible d'inférer que « Paris » est une instance de la classe « Ville » et la « France » est une instance de la classe « Pays ».

```
(rdfs:domain isTheCapitalOf City)
(rdfs:range isTheCapitalOf States)
(isTheCapitalOf City States)
Implique que
(rdf:type Paris City)
(rdf:type France States)
```

Logique de description :

TBox	ABox	Implication
$(\geq 1 \text{ isTheCapitalOf} \sqsubseteq \text{City})$ $(\top \sqsubseteq \forall \text{ isTheCapitalOf} . \text{States})$	$(\text{isTheCapitalOf}(\text{Paris}, \text{France}))$	$\text{City}(\text{Paris})$ $\text{States}(\text{France})$

D.4.1.3/ L'AXIOME OWL :ONEOF

Si une classe A est owl :oneOf d'une liste d'individus, par exemple, X, Y et Z alors chacun de ces individus est rdf :type A. Par exemple, si la classe « Continent » est owl :oneOf des individus « Europe », « Asie », « Afrique », « Océanie », « Amérique ». Alors ces individus sont inférés comme des instances de la classe « Continent ».

```
(owl:oneOf Continent Europe)
(owl:oneOf Continent Asie)
(owl:oneOf Continent Afrique)
(owl:oneOf Continent Océanie)
(owl:oneOf Continent Amérique)
Implique que
(rdf:type Europe Continent)
(rdf:type Asie Continent)
(rdf:type Afrique Continent)
(rdf:type Océanie Continent)
(rdf:type Amérique Continent)
```

Logique de description :

TBox	ABox	Implication
$(Continent \equiv Europe \sqcup Asie \sqcup Afrique \sqcup Océanie \sqcup Amérique)$		$(Continent(Europe))$ $(Continent(Asie))$ $(Continent(Afrique))$ $(Continent(Océanie))$ $(Continent(Amérique))$

D.4.1.4/ LA RESTRICTION OWL :HASVALUE

Si une restriction owl : hasValue restreint une propriété particulière à une valeur particulière, et qu'un objet a cette valeur pour cette propriété, l'objet est inféré comme une instance de cette restriction. Par exemple, si une « zone déforestée » est définie comme tout individu dont la propriété « type de surface » possède comme valeur unique « Déforesté » alors si l'individu « Forêt Amazonienne » possède la propriété « type de surface » avec pour valeur « Déforesté », il est possible d'inférer que cet individu est une instance de la classe « Zone déforestée ».

```
(owl:onProperty RestrictionDeforestedArea LandCoverType)
(owl:hasValue RestrictionDeforestedArea Deforested)
(LandCoverType Amazonian_Forest Deforested)
Implique que
(rdf:type Amazonian_Forest RestrictionDeforestedArea)
```

Logique de description :

TBox	ABox	Implication
$(ResDeforestedArea \equiv LCType : Deforested)$	$(LCType(Amazonian_Forest, Deforested))$	$(ResDeforestedArea(Amazonian_Forest))$

D.4.1.5/ LA RESTRICTION OWL :ALLVALUESFROM

Si un objet est une instance d'une restriction owl : allValuesFrom, et que l'objet a des valeurs pour la propriété spécifiée, les valeurs sont du type spécifié. Par exemple, une « ville aéroportuaire » est une ville dont toutes les valeurs de la propriété « possède aéroport » sont des instances de la classe « aéroport ». Si Paris est définie comme une « ville aéroportuaire » et comporte la propriété « possède aéroport » avec comme valeur « Charles-De-Gaulle » alors il est possible d'inférer que « Charles-De-Gaulle » est une instance de la classe « Aéroport ».

```
(owl:onProperty Airport_City hasAirport)
(owl:allValuesFrom Airport_City Airport)
(rdf:type Paris Airport_City)
(hasAirport Paris Charles-De-Gaulle)
Implique que
(rdf:type Charles-De-Gaulle Airport)
```

Logique de description :

TBox	ABox	Implication
$(Airport_City \equiv \forall hasAirport.Airport)$	$(hasAirport(Paris, Charles - De - Gaulle))$ $(Airport_City(Paris))$	$(Airport(Charles - De - Gaulle))$

D.4.1.6/ LA RESTRICTION OWL :SOMEVALUESFROM

Si une restriction owl :someValuesFrom restreint une propriété particulière à un type particulier, et si un objet a des valeurs du type spécifié pour la propriété spécifiée, alors l'objet est une instance de la restriction. Par exemple, en définissant un territoire historique comme tout individu possédant au moins une propriété « Point d'intérêt » dont la valeur est une instance de la classe « Monument historique » alors si la tour Eiffel est définie à la fois comme un point d'intérêt de la ville de Paris et une instance de la classe « Monument historique », il est possible d'inférer que Paris est une instance qui satisfait la classe « Territoire historique ».

```
(owl:onProperty Historical_Territory PointOfInterest)
(owl:someValuesFrom Historical_Territory Historical_Monument)
(rdf:type Tour_Eiffel Historical_Monument)
(PointOfInterest Paris Tour_Eiffel)
Implique que
(rdf:type Paris Historical_Territory)
```

Logique de description :

TBox	ABox	Implication
$(Historical_Territory \equiv \exists PointOfInterest.Historical_Monument)$	$(PointOfInterest(Paris, Tour_Eiffel))$ $(Historical_Monument(Tour_Eiffel))$	$(Historical_Territory(Paris))$

D.4.2/ INFÉRENCE D'INSTANCES DE PROPRIETES

Cette partie met en évidence les différents axiomes pouvant servir de support à l'inférence d'assertions de rôle.

D.4.2.1/ RESTRICTION OWL :HASVALUE

Si un objet est défini comme une instance (`rdf:type`) d'une restriction owl : `hasValue`, l'objet a la valeur spécifiée pour la propriété spécifiée dans la restriction. Par exemple, si une « zone déforestée » est définie comme la classe regroupant les individus dont la propriété « type de surface » comporte la valeur unique « déforestée » alors si la « forêt amazonienne » est définie comme une instance de cette classe, cette instance doit posséder la propriété « type de surface » avec pour valeur unique « déforestée ».

```
(owl:onProperty RestrictionDeforestedArea LandCoverType)
(owl:hasValue RestrictionDeforestedArea Deforested)
(rdf:type Amazonian_Forest Deforested)
Implique que
(LandCoverType Amazonian_Forest Deforested)
```

Logique de description :

TBox	ABox	Implication
$(\text{RestrictionDeforestedArea} \equiv \text{LandCoverType} : \text{Deforested})$	$(\text{ResDeforestedArea}(\text{Amazonian_Forest}))$	$(\text{LCType}(\text{Amazonian_Forest}, \text{Deforested}))$

D.4.2.2/ LA TRANSITIVITÉ A L'AIDE DE OWL :TRANSITIVEPROPERTY

La propriété de transitivité appliquée à une relation R entre un individu i1 et i2, et entre un individu i2 et i3 permet de définir que si R(i1, i2) et R(i2, i3) alors R(i3, i1). Par exemple, étant donné une relation Contient définie comme transitive et si la base de connaissance comporte l'assertion définissant que la France contient l'île de France et que l'île de France contient Paris alors la France contient Paris.

```
(rdf:type Contains owl:TransitiveProperty)
(Contains France Ile_De_France)
(Contains Ile_De_France Paris)
Implique que
(Contains Paris France)
```

Logique de description :

TBox	ABox	Implication
$(\text{Tr}(\text{Contains}))$	$(\text{Contains}(\text{France}, \text{Ile_De_France}))$ $(\text{Contains}(\text{Ile_De_France}, \text{Paris}))$	$(\text{Contains}(\text{Paris}, \text{France}))$

D.4.2.3/ LA SYMÉTRIE À L'AIDE DE OWL :SYMMETRICPROPERTY

La propriété de symétrie appliquée à une relation R entre un individu i1 et i2 permet de définir que si R(i1, i2) alors R(i2, i1). Par exemple, étant donné une relation *Touches* définie comme symétrique et si la base de connaissances comporte l'assertion définissant que la France touche l'Italie alors l'Italie touche également la France.

```
(rdf:type Touches owl:SymmetricProperty)
(Touches France Italy)
Implique que
(Touches Italy France)
```

Logique de description :

TBox	ABox	Implication
(<i>Touches</i> \equiv <i>Touches</i> ⁻)	(<i>Touches</i> (France, Italy))	(<i>Touches</i> (Italy, France))

D.4.2.4/ LA PROPRIÉTÉ INVERSE À L'AIDE DE OWL :INVERSEOF

La propriété inverse définie entre une relation R1 et une relation R2 et appliquée entre un individu i1 et i2 permet de définir que si R1(i1, i2) alors R2(i2, i1). Par exemple, étant donné une relation *Contient* définie comme inverse d'une relation *estContenuDans* et si la base de connaissance comporte l'assertion définissant que l'Europe contient la France alors la France est contenue dans l'Europe.

```
(owl:inverseOf Contains Within)
(Contains Europe France)
Implique que
(Within France Europe)
```

Logique de description :

TBox	ABox	Implication
(<i>Contains</i> \equiv <i>Within</i> ⁻)	(<i>Contains</i> (Europe, France))	(<i>Within</i> (France, Europe))

D.4.2.5/ OWL :ALLDIFFERENT ET OWL :DIFFERENTFROM

Tous les éléments d'un owl :AllDifferent sont owl :differentFrom les uns des autres.

```
(rdf:type bkn1 owl:AllDifferent)
(owl:distinctMembers bkn1 Urban)
(owl:distinctMembers bkn1 Forest)
(owl:distinctMembers bkn1 Agricultural)
Implique que
(owl:differentFrom Urban Forest)
(owl:differentFrom Forest Agricultural)
(owl:differentFrom Agricultural Urban)
```

Logique de description :

TBox	ABox	Implication
$(bkn1 : AllDifferent)$ $(distinctMembers(bkn1, Urban))$ $(distinctMembers(bkn1, Forest))$ $(distinctMembers(bkn1, Agricultural))$		$(differentFrom(Urban, Forest))$ $(differentFrom(Forest, Agricultural))$ $(differentFrom(Agricultural, Urban))$

D.4.2.6/ PROPRIÉTÉS FONCTIONNELLES

La propriété fonctionnelle définie sur une relation R et appliquée entre un individu $i1$ et $i2$ permet de définir que si $R(i1, i2)$ alors $i2$ est l'unique valeur acceptée pour $i1$. Par exemple, dans le domaine de la modélisation spatio-temporelle, l'évolution d'une entité entre deux instants de temps consécutifs peut être soit une relation de continuation si l'identité de l'entité n'a pas été affectée par le changement, soit une relation de dérivation si l'identité a été affectée. Une entité peut subir plusieurs relations de dérivation lorsqu'elle se subdivise en deux ou plusieurs entités entre deux instants de temps. En revanche, une seule relation de continuation n'est autorisée à chaque évolution, c'est pourquoi la relation de continuation doit être définie comme fonctionnelle. Etant donné une relation de continuation définie comme fonctionnelle et si la base de connaissance comporte l'assertion définissant que l'Amazonie a subi deux relations de continuation lors d'une division entre un instant t_i et t_{i+1} alors il est possible d'inférer que les deux valeurs correspondants au *range* de la relation de continuation font référence au même individu. Dans le cas contraire, une inconsistance devra être levée.

```
(rdf:type Continuation owl:FunctionalProperty)
(Continuation Amazonia_TS1 Amazonia_TS2)
(Continuation Amazonia_TS1 Amazonia_TS2)
Implique que
(owl:sameAs Amazonia_TS2 Amazonia_TS3)
```

Logique de description :

TBox	ABox	Implication
$(\top \sqsubseteq \leq 1 Continuation)$	$(Continuation(Amazonia_TS1, Amazonia_TS2))$ $(Continuation(Amazonia_TS1, Amazonia_TS3))$	$(Amazonia_TS2 \doteq Amazonia_TS3)$

D.4.2.7/ PROPRIÉTÉS INVERSEMENT FONCTIONNELLES

La propriété inversement fonctionnelle définie sur une relation R et appliquée entre un individu $i1$ et $i2$ permet de définir que si $R(i1, i2)$ alors $i1$ est l'unique valeur acceptée pour $i2$. Nous reprenons l'exemple de la relation de continuation pour illustrer notre propos. Dans l'exemple précédent, nous avons mis en évidence le cas de la division d'une entité. Un autre motif d'évolution couramment observé dans la littérature de la modélisation spatio-temporelle concerne la fusion d'entité. Dans le cas d'une fusion, une entité peut être le résultat de plusieurs dérivations, en revanche, une seule continuation n'est autorisée c'est pourquoi la relation de continuation doit être définie à la fois comme fonctionnelle et inversement fonctionnelle. Etant donné une relation de continuation définie comme inversement fonctionnelle et si la base de connaissance comporte l'assertion définissant que l'Amazonie a subi deux relations de continuation lors d'une fusion entre un instant t_i et t_{i+1} alors il est possible d'inférer que les deux valeurs correspondants au *domain* de

la relation de continuation font référence au même individu. Dans le cas contraire, une inconsistance devra être levée.

```
(rdf:type Continuation owl:InverseFunctionalProperty)
(Continuation Amazonia_TS1 Amazonia_TS3)
(Continuation Amazonia_TS2 Amazonia_TS2)
Implique que
(owl:sameAs Amazonia_TS1 Amazonia_TS2)
```

Logique de description :

TBox	ABox	Implication
$(\top \sqsubseteq \leq 1\text{Continuation}^{\cdot}\top)$	$(\text{Continuation}(\text{Amazonia_TS1}, \text{Amazonia_TS3}))$ $(\text{Continuation}(\text{Amazonia_TS2}, \text{Amazonia_TS3}))$	$(\text{Amazonia_TS1} \doteq \text{Amazonia_TS2})$

Résumé :

Les gérants de territoires souhaitent avoir un aperçu des connaissances en cours et de l'évolution de certaines caractéristiques du territoire. Pour ce faire, les outils de télédétection enregistrent une grande quantité d'informations liées à la couverture territoriale permettant l'étude de processus dynamiques. La littérature du domaine de la modélisation spatio-temporelle est vaste et a donné lieu à de nombreux modèles. Chacun révélant une certaine aptitude pour capturer l'évolution des différentes caractéristiques d'un environnement géospatial. Toutefois, les données nécessitent de nouveaux outils pour identifier des motifs et extraire de la connaissance.

Nous proposons un modèle capable de découvrir la connaissance sur des données parcellaires et permettant l'analyse des phénomènes dynamiques à l'aide de données temporelles, spatiales et thématiques. Le modèle est appelé Continuum et se base sur les technologies du Web Sémantique pour permettre une représentation accrue du contexte de l'environnement géospatial et fournir des résultats d'analyses proches de ceux des experts du domaine via des opérations de raisonnement automatiques et valides. En définitive, ce modèle permet d'améliorer notre compréhension de la dynamique des territoires.

Mots-clés : Système d'Information Géographique, dynamiques spatio-temporelles, analyse géospatiale, ontologie, modélisation et raisonnement spatial qualitatif, système d'aide à la décision

Abstract:

There is a need for decision-makers to be provided with both an overview of existing knowledge, and information which is as complete and up-to-date as possible on changes in certain features of the biosphere. Another objective is to bring together all the many attempts which have been made over the years at various levels (international, Community, national and regional) to obtain more information on the environment and the way it is changing. As a result, remote sensing tools monitor large amount of land cover informations enabling study of dynamic processes. However the size of the dataset require new tools to identify pattern and extract knowledge. We propose a model to discover knowledge on parcel data allowing analysis of dynamic geospatial phenomena using time, spatial and thematic data. The model is called Continuum and is able to track the evolution of spatial entities along time. Based on semantic web technologies, the model allows users to specify and to query spatio-temporal informations based on semantic definitions. The semantic of spatial relationships are of interest to qualify filiation relationships. The result of this process permit to identify evolutive patterns as a basis for studying the dynamics of the geospatial environment. To this end, we use CORINE datasets to study changes in a specific part of France. In our approach, we consider entities as having several representations during their lifecycle. Each representation includes identity, spatial and descriptives properties that evolve over time.

Keywords: Geographic Information System, spatio-temporal dynamics, geospatial analysis, ontology, qualitative spatial modeling and reasoning, decision-support systems

The logo for SPIM (École doctorale SPIM) features the letters 'S', 'P', 'I', and 'M' in a large, white, sans-serif font. The 'S' is stylized with a thick orange horizontal bar to its left.