



**HAL**  
open science

# Une approche basée sur le web sémantique pour l'étude de trajectoires de vie

David Noel

► **To cite this version:**

David Noel. Une approche basée sur le web sémantique pour l'étude de trajectoires de vie. Web. Université Grenoble Alpes, 2019. Français. NNT : 2019GREAM022 . tel-02307133

**HAL Id: tel-02307133**

**<https://theses.hal.science/tel-02307133>**

Submitted on 7 Oct 2019

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

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



## THÈSE

Pour obtenir le grade de

## DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

**David NOEL**

Thèse dirigée par **Jérôme GENSEL**  
et codirigée par **PIERRE LE QUEAU**, Communauté Université  
Grenoble Alpes  
et **Marlène VILLANOVA OLIVER**, MCF, UGA

préparée au sein du **Laboratoire d'Informatique de Grenoble**  
dans **l'École Doctorale Mathématiques, Sciences et  
technologies de l'information, Informatique**

## Une approche basée sur le web sémantique pour l'étude de trajectoires de vie

## A semantic web based approach to study life trajectories

Thèse soutenue publiquement le **17 juin 2019**,  
devant le jury composé de :

**Monsieur JÉRÔME GENSEL**  
PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Directeur de thèse

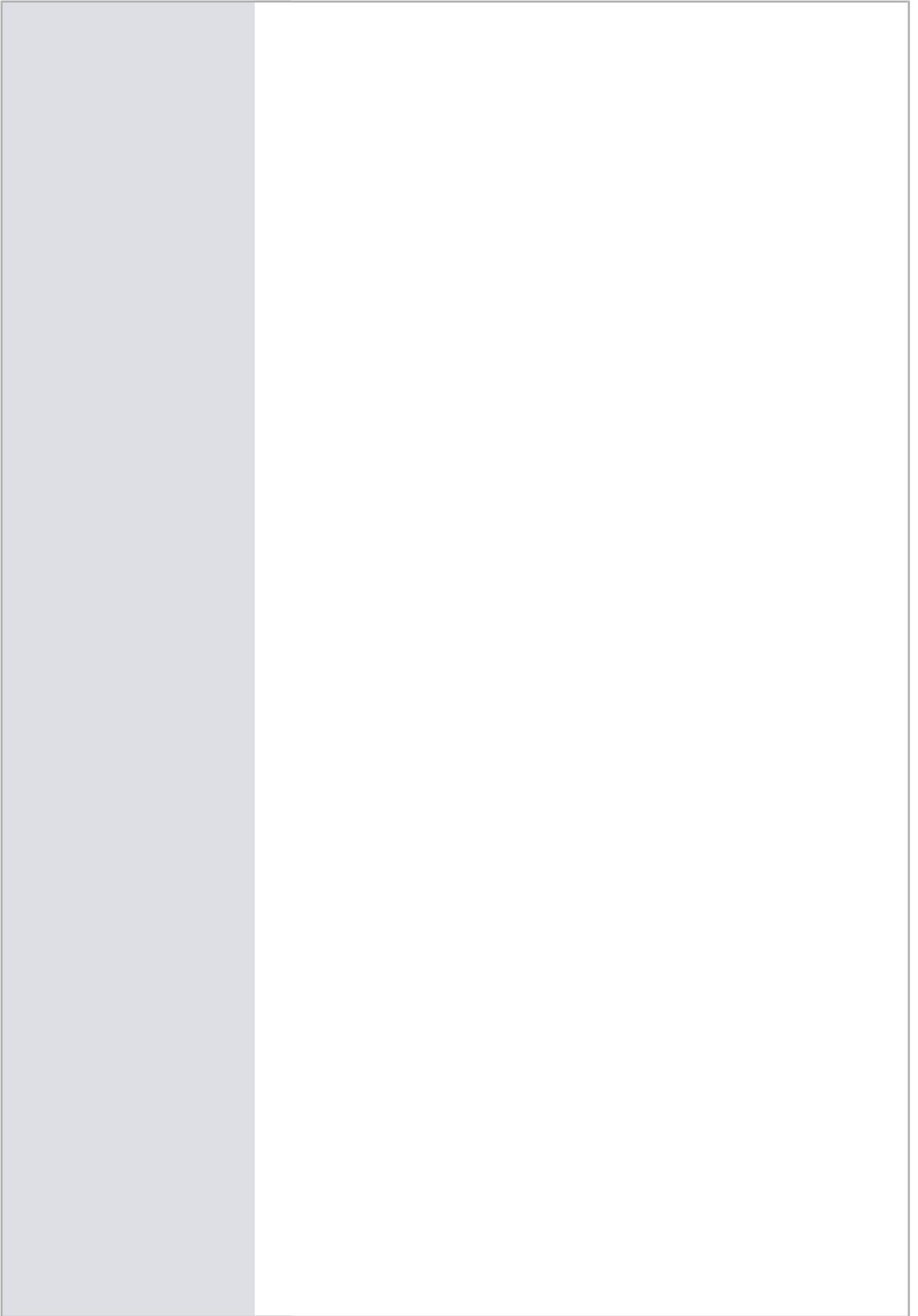
**Monsieur GILLES FALQUET**  
PROFESSEUR ASSOCIE, UNIVERSITE DE GENEVE - SUISSE,  
Rapporteur

**Monsieur ALAIN BOUJU**  
MAITRE DE CONFERENCES, UNIVERSITE DE LA ROCHELLE,  
Rapporteur

**Monsieur DIDIER JOSSELIN**  
DIRECTEUR DE RECHERCHE, CNRS DELEGATION PROVENCE ET  
CORSE, Président

**Monsieur PIERRE LE QUEAU**  
MAITRE DE CONFERENCES, UNIVERSITE GRENOBLE ALPES, Co-  
directeur de thèse

**Madame MARLENE VILLANOVA-OLIVER**  
MAITRE DE CONFERENCES, UNIVERSITE GRENOBLE ALPES,  
Examineur



---

## **A Semantic Web based approach to study life trajectories**

**Abstract :** The notion of trajectory is the subject of many works in computer science. The life trajectory has several peculiarities which distinguish it from the trajectories usually considered in these works. It is first of all its temporal hold, which is the life, the existence of the observed subject. It is then its thematic hold, this one potentially concerning multiple aspects of the life of an object or an individual. Finally, it is the metaphorical use of the term trajectory, which refers more to the meaning of the trajectory than to the description of a simple evolution in time and space. The life trajectory is used by the expert (sociologist, urban planner...) who wishes to put in perspective the information on individuals to better understand their choices. The motivations for studying the life trajectory are depending on the application and themes considered : the relation to work and employment, family life, social life, health, residential trajectory... We propose a Semantic Web based approach to study life trajectories, which allows their modeling, collection and analysis. This approach is embodied by a software architecture whose components are configurable for each application case. This architecture is based on a life trajectory ontology design pattern, as well as a model of explanatory factors for life events. To operationalize the proposed modeling, we designed algorithms that allow the creation of a life trajectory ontology by exploiting the previous pattern and model. For data collection, we developed APIs to facilitate i) the construction of a model-compliant data collection interface ; and ii) the insertion of the collected data into a Triple Store. Our approach allows the representation, and hence the collection and exploitation of multi-granular information, whether spatial, temporal or thematic. Finally, to allow the analysis of the trajectories, we propose generic functions, which are implemented by extending the SPARQL language. The methodological approach and the proposed tools are validated on a case study on residential choices of individuals in the Grenoble metropolitan area by highlighting the characteristics of their residential trajectory and the explanatory elements of it, including from their personal and professional trajectories.

**Keywords :** ...

---

---

## Une approche basée sur le Web sémantique pour l'étude de trajectoires de vie

**Résumé :** La notion de trajectoire fait l'objet de nombreux travaux en informatique. La trajectoire de vie a plusieurs particularités qui la distinguent des trajectoires habituellement considérées dans ces travaux. C'est d'abord son emprise *temporelle*, qui est celle de la vie, de l'existence du sujet observé. C'est ensuite son emprise *thématique*, celle-ci pouvant potentiellement concerner de multiples aspects de la vie d'un objet ou d'un individu. C'est enfin, l'utilisation *métaphorique* du terme de trajectoire, qui renvoie davantage au *sens* de la trajectoire qu'à la description d'une simple évolution dans le temps et l'espace. La trajectoire de vie est utilisée par l'expert (sociologue, urbaniste...) qui souhaite mettre en perspective dans le temps les informations sur les individus pour mieux les comprendre. Les motivations d'étude de la trajectoire de vie sont différentes selon les cas d'application et les thématiques considérées : le rapport au travail, la vie familiale, la vie sociale, la santé, la trajectoire résidentielle... Nous proposons une approche d'étude des trajectoires de vie, basée sur le Web Sémantique, qui en permet la modélisation, la collecte et l'analyse. Cette approche est concrétisée par une architecture logicielle. Cette architecture repose sur un patron de conception d'ontologie de trajectoire de vie, ainsi que sur un modèle de facteurs explicatifs d'événements de vie. Pour la modélisation, nous avons conçu des algorithmes qui permettent la création d'une ontologie de trajectoire de vie en exploitant le patron et le modèle précédents. Pour la collecte de données, nous avons développé des APIs permettant de faciliter *i)* la construction d'une interface de collecte de données conforme aux modèles ; *ii)* l'insertion des données collectées dans un Triple Store. Notre approche permet la représentation, et, par suite, la collecte et l'exploitation d'informations multi-granulaires, qu'elles soient spatiales, temporelles ou thématiques. Enfin, pour permettre l'analyse des trajectoires, nous proposons des fonctions génériques, implémentées en étendant le langage SPARQL. L'approche méthodologique et les outils proposés sont validés sur un cas d'application visant à étudier les choix résidentiels d'individus en mettant en exergue les caractéristiques de leur trajectoire résidentielle et les éléments explicatifs de celle-ci.

**Mots-clés :** ...

---

## Remerciements

Remerciements.



# Table des matières

Publications . . . . .	1
<b>Introduction</b>	<b>7</b>
I Introduction . . . . .	9
1 Motivation . . . . .	9
2 Description du problème . . . . .	11
3 Organisation du manuscrit . . . . .	16
3.1 Partie I : état de l'art . . . . .	17
3.2 Partie II : contributions . . . . .	17
<b>État de l'art</b>	<b>19</b>
II Trajectoires sémantiques . . . . .	21
1 Introduction . . . . .	21
2 Origines des trajectoires sémantiques . . . . .	22
2.1 Des trajectoires spatio-temporelles brutes aux approches basées sur la sémantique . . . . .	22
2.2 La time-geography . . . . .	24
2.3 Conclusion . . . . .	26
3 Modèles de trajectoires sémantiques . . . . .	26
3.1 Introduction . . . . .	26
3.2 Un modèle basé sur une structuration en déplacements et arrêts . . . . .	27
3.3 Autres modèles conceptuels de trajectoires . . . . .	29
3.4 Conclusion . . . . .	31
4 Web sémantique et ontologies de trajectoires . . . . .	31
4.1 Introduction . . . . .	31
4.2 Technologies du web sémantique et ontologies spatio-temporelles	32
4.3 Ontologies de trajectoires sémantiques . . . . .	39
4.4 Conclusion . . . . .	41
5 Conclusion . . . . .	42
III Multiples points de vue et trajectoires . . . . .	43
1 Introduction . . . . .	43
2 Information multi-point de vue . . . . .	44
2.1 La notion de point de vue . . . . .	44
2.2 La multi-représentation . . . . .	45
2.3 Conclusion . . . . .	45

3	Information multi-granulaire . . . . .	46
3.1	La granularité . . . . .	46
3.2	Approches utilisées pour la modélisation multi-granulaire .	47
3.3	Conclusion . . . . .	49
4	Modéliser des trajectoires multi-points de vue et multi-granulaires	49
4.1	Introduction . . . . .	49
4.2	Trajectoires sémantiques multi-points de vue et multi-granulaires	50
4.3	Trajectoires de vie de Thériault . . . . .	52
4.4	Conclusion . . . . .	55
5	Conclusion . . . . .	55
IV	Synthèse et positionnement . . . . .	57
1	Introduction . . . . .	57
2	Modéliser des trajectoires de vie . . . . .	57
2.1	Limites des approches de modélisation et positionnement .	58
2.2	Analyse comparative des modèles conceptuels . . . . .	60
2.3	Conclusion . . . . .	63
3	Approches de modélisation, d'acquisition et d'exploitation des données . . . . .	63
3.1	Introduction . . . . .	63
3.2	Limites des approches présentés dans l'état de l'art . . . . .	64
3.3	Vue générale de notre approche . . . . .	65
3.4	Conclusion . . . . .	66
4	Conclusion . . . . .	66
	<b>Contributions</b>	<b>69</b>
V	Patron et modèle socles l'approche . . . . .	71
1	Introduction . . . . .	71
2	Patron de conception de trajectoire de vie . . . . .	72
2.1	Notions d'épisodes et d'événements . . . . .	72
2.2	Trajectoire de vie métaphorique . . . . .	75
2.3	Trajectoire de vie géographique . . . . .	76
2.4	Conclusion . . . . .	77
3	Modèle de facteurs explicatifs . . . . .	77
3.1	La notion de facteurs explicatifs . . . . .	77
3.2	Typologie des facteurs explicatifs . . . . .	81
3.3	Conclusion . . . . .	82
4	Conclusion . . . . .	82
VI	Méthodologie et architecture pour l'étude des trajectoires de vie .	85
1	Introduction . . . . .	85

---

2	Vue générale de l'approche et de l'architecture . . . . .	86
3	Modélisation et paramétrage de l'architecture . . . . .	87
3.1	Introduction . . . . .	87
3.2	Principales étapes de la démarche de modélisation . . . . .	87
3.3	Fonctionnement du module Ontology Manager pour la modélisation . . . . .	90
3.4	Conclusion . . . . .	97
4	Acquisition de données . . . . .	97
4.1	Conclusion . . . . .	100
5	Exploitation des données . . . . .	100
6	Conclusion . . . . .	112
VII	Application de la méthodologie à l'étude des trajectoires résidentielles . . . . .	115
1	Introduction . . . . .	115
2	Application du patron de conception pour la production d'un modèle cible . . . . .	115
2.1	Choix des points de vue pertinents . . . . .	116
2.2	Création de l'ontologie de trajectoire . . . . .	117
3	Collecte de trajectoires de vie . . . . .	121
4	Analyse des trajectoires de vie collectées . . . . .	124
5	Conclusion . . . . .	138
	<b>Conclusion</b> . . . . .	<b>139</b>
VIII	Conclusion et perspectives . . . . .	141
1	Conclusion . . . . .	141
2	Perspectives . . . . .	142
2.1	Validation de l'approche et évaluation des opérateurs . . . . .	142
2.2	Création d'une interface d'étude des trajectoires de vie . . . . .	143
2.3	Modélisation de trajectoires . . . . .	143
2.4	Acquisition des données . . . . .	144
2.5	Exploitation des trajectoires . . . . .	145
	<b>Annexes</b> . . . . .	<b>147</b>
A	Le patron de conception . . . . .	149
B	Le modèle de facteur explicatif . . . . .	153
C	L'implémentation des fonctions SPARQL avec l'API Jena . . . . .	157
1	Fonction ltofn:before . . . . .	157
2	Fonction ltofn:after . . . . .	158
3	Fonction ltofn:overlaps . . . . .	159

4	Fonction ltofn:isFactorOutFrom . . . . .	160
5	Fonction ltofn:isFactorEnterIn . . . . .	161
6	Fonction ltofn:isFactorOfExplainingEventType . . . . .	162
7	Fonction ltofn:isFactorOfExplainedEventType . . . . .	163
8	Fonction ltofn:IsFactorOfExplainedTrajectoryType . . . . .	164
9	Fonction ltofn:IsFactorOfExplainingTrajectoryType . . . . .	165
10	Fonction ltofn:episodeBeforeFactor . . . . .	166
11	Fonction ltofn:episodeAfterFactor . . . . .	167
12	Fonction ltofn:episodeAtFactorDate . . . . .	168
D	L'ontologie LTO pour l'étude du choix résidentiels . . . . .	169
E	Vocabulaires contrôlés utilisés . . . . .	181
1	Vocabulaire contrôlé des types de logement . . . . .	181
2	Vocabulaire contrôlé des types de logement . . . . .	182
3	Vocabulaire contrôlé des intervalles de loyers . . . . .	183
4	Vocabulaire contrôlé des types de contrat . . . . .	184
5	Vocabulaire contrôlé des intervalles de salaires . . . . .	186
6	Vocabulaire contrôlé des statuts maritaux . . . . .	187
7	Vocabulaire contrôlé des événements résidentiels . . . . .	188
8	Vocabulaire contrôlé des événements professionnel . . . . .	189
9	Vocabulaire contrôlé des événements familiaux . . . . .	190
F	Patrons de conception de requête d'insertion des données . . . . .	193
1	Requête d'insertion d'un épisode d'une trajectoire géographique . . . . .	193
G	Interfaces de collecte de trajectoire de vie . . . . .	195
1	Collecte des épisodes et événements professionnels . . . . .	196
2	Collecte des épisodes et événements familiaux . . . . .	197
	Bibliographie . . . . .	199

# Liste de Publications

Les articles suivants ont été publiés lors de cette thèse :

- **Marlène Villanova-Oliver, David Noël, Jérôme Gensel, Pierre Le Quéau.** A framework for describing and analysing lifelines : Taking a step towards studying residential migration factors, in (eds) Jean-Christophe Dissart and Natacha Seigneuret, *Local Resources and Well-Being : A Multidisciplinary Perspective*. (à paraître en 2019).
- **David Noël, Marlène Villanova-Oliver, Jérôme Gensel, Pierre Le Quéau.** Design patterns for modelling life trajectories in the semantic web. In *International Symposium on Web and Wireless Geographical Information Systems*, pages 51-65. Springer, 2017.
- **David Noël, Marlène Villanova-Oliver, Jérôme Gensel, Pierre Le Quéau.** Collecter et analyser des trajectoires de vie, *ATELIER Analyse de trajectoires spatio-temporelles, SAGEO Spatial Analysis and GEomatics*, Rouen, France, 2017.
- **David Noël.** Metropo-lifeline : Participatory description and analysis of the migration of residents within a metropolitan area. In *European Knowledge Acquisition Workshop*, pages 269-277. Springer, 2016.
- **David Noël, Marlène Villanova-Oliver, Jérôme Gensel, Pierre Le Quéau.** Modélisation de trajectoires sémantiques intégrant perspectives multiples et facteurs explicatifs : application aux trajectoires de vie. *Revue Internationale de Géomatique*, 26(4) pages 491-510, 2016.
- **David Noël, Marlène Villanova-Oliver, Jérôme Gensel, Pierre Le Quéau.** Modélisation de trajectoires sémantiques intégrant perspectives multiples et facteurs explicatifs : Application aux trajectoires de vie. *SAGEO Spatial Analysis and GEomatics*, Hammamet, Tunisie, 2015.
- **David Noël, Marlène Villanova-Oliver, Jérôme Gensel, Pierre Le Quéau.** Modeling semantic trajectories including multiple viewpoints and explanatory factors : application to life trajectories. In *Proceedings of the 1st International ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*, pages 107-113. ACM, 2015.



# Table des figures

1.1	Extrait d'une trajectoire de vie centrée sur l'aspect résidentiel. Représentation inspirée de <a href="#">Thériault et al. [1999]</a> . . . . .	12
2.1	De la trajectoire brute à la trajectoire sémantique d'après <a href="#">Yan and Spaccapietra [2009]</a> . . . . .	23
2.2	Visualisation de trois trajectoires de vie dans le cube espace-temps [ <a href="#">Sinha and Mark, 2005</a> ] . . . . .	25
2.3	Le design pattern proposé par <a href="#">Spaccapietra et al. [2008]</a> . . . . .	28
2.4	Modèle conceptuel de trajectoire basé sur des événements [ <a href="#">Andrienko et al., 2011</a> ] . . . . .	30
2.5	Les technologies du Web sémantique . . . . .	32
2.6	Le Linked Open Data Cloud en 2014 [ <a href="#">Schmachtenberg et al., 2014</a> ] . . . . .	34
2.7	La classe spatiale SpatialObject, racine de l'ontologie GeoSPARQL. Source : <a href="http://www.opengeospatial.org/blog/1673">http://www.opengeospatial.org/blog/1673</a> . . . . .	35
2.8	Exemple d'utilisation de Geosparql pour décrire un lieu et sa géométrie. . . . .	36
2.9	Le modèle d'entité temporelle utilisé dans OWL-Time [ <a href="#">Hobbs and Pan, 2006a</a> ] . . . . .	37
2.10	L'ontologie <i>Event Core</i> [ <a href="#">Krisnadhi and Hitzler, 2017</a> ] . . . . .	38
2.11	Ontologie d'événements [ <a href="#">Raimond and Abdallah, 2007</a> ] . . . . .	39
2.12	Le <i>design pattern</i> proposé par <a href="#">Baglioni et al. [2008]</a> . . . . .	40
2.13	Le design pattern d'ontologie proposé par [ <a href="#">Hu et al., 2013</a> ] . . . . .	41
3.1	Les différents niveaux de détail d'un prédicat représentés sur un digramme de Hasse [ <a href="#">Silva et al., 2015</a> ] . . . . .	48
3.2	Le modèle CONSTAnT [ <a href="#">Bogorny et al., 2014</a> ] prend en compte de multiples aspects pour une trajectoire . . . . .	50
3.3	Représentation de l'ontologie Baquara [ <a href="#">Fileto et al., 2015</a> ] . . . . .	52
3.4	La trajectoire de vie de <a href="#">Thériault et al. [2002]</a> . . . . .	54
4.1	Les différents composants nécessaires à la gestion de données dans une approche d'étude des trajectoires par <i>design pattern</i> dans les travaux existants . . . . .	64

4.2	Les différents composants nécessaires à la gestion de données dans une approche d'étude des trajectoires par <i>design pattern</i> intégrant la collecte de données . . . . .	65
4.3	Vue générale de notre approche et de l'outillage mis en place . . . . .	65
5.1	extrait d'une trajectoire de vie centrée sur l'aspect résidentiel. Représentation inspirée de Thériault [Thériault et al., 1999] . . . . .	73
5.2	Le <i>design pattern</i> de trajectoires de vie métaphorique . . . . .	75
5.3	Le design pattern de trajectoires de vie . . . . .	77
5.4	Exemples de facteurs explicatifs . . . . .	79
5.5	Modèle de facteurs explicatifs de trajectoires . . . . .	80
6.1	Architecture pour la collecte et l'analyse de trajectoires de vie [Villanova-Oliver et al., 2018] . . . . .	87
6.2	Représentation des étapes de la démarche de modélisation [Villanova-Oliver et al., 2018] . . . . .	88
6.3	Représentation de l'association de deux trajectoires thématiques A et B . . . . .	93
6.4	Requêtes génériques et requêtes dédiées . . . . .	94
6.5	Un exemple de trajectoire de vie . . . . .	105
7.1	Différents points de vue pertinents pour l'étude du choix résidentiel . . . . .	116
7.2	Ontologie de la trajectoire résidentielle . . . . .	118
7.3	Ontologie de la trajectoire professionnelle . . . . .	119
7.4	Ontologie de la trajectoire familiale . . . . .	120
7.5	Interface de collecte d'un épisode résidentiel . . . . .	122
7.6	Un exemple d'instance de trajectoire résidentielle . . . . .	124
7.7	Résultat de la requête <i>1a</i> dans Graphdb . . . . .	126
7.8	Extrait du résultat de la requête <i>1b</i> dans Graphdb . . . . .	126
7.9	Extrait du résultat de la requête <i>2</i> dans Graphdb . . . . .	127
7.10	Extrait du résultat de la requête <i>4</i> dans Graphdb . . . . .	129
7.11	Extrait du résultat de la requête <i>5</i> dans Graphdb . . . . .	130
7.12	Extrait du résultat de la requête de création de la grille de lecture dans Graphdb . . . . .	133
3.1	Capture d'écran Eclipse de la Classe BeforePropertyFunction . . . . .	157
3.2	Capture d'écran Eclipse de la Classe AfterPropertyFunction . . . . .	158
3.3	Capture d'écran Eclipse de la Classe OverlapsPropertyFunction . . . . .	159

---

3.4	Capture d'écran Eclipse de la Classe IsFactorOutFromPropertyFunction . . . . .	160
3.5	Capture d'écran Eclipse de la Classe IsFactorEnterInPropertyFunction . . . . .	161
3.6	Capture d'écran Eclipse de la Classe IsFactorOfExplainingEventPropertyFunction . . . . .	162
3.7	Capture d'écran Eclipse de la Classe IsFactorOfExplainedEventPropertyFunction . . . . .	163
3.8	Capture d'écran Eclipse de la Classe IsFactorOfExplainedTrajectoryPropertyFunction . . . . .	164
3.9	Capture d'écran Eclipse de la Classe IsFactorOfExplainingTrajectoryPropertyFunction . . . . .	165
3.10	Capture d'écran Eclipse de la Classe EpisodeBeforeFactorPropertyFunction . . . . .	166
3.11	Capture d'écran Eclipse de la Classe EpisodeAfterFactorPropertyFunction . . . . .	167
3.12	Capture d'écran Eclipse de la Classe EpisodeAtFactorDatePropertyFunction . . . . .	168
7.1	Interface de collecte d'un épisode professionnel . . . . .	196
7.2	Interface de collecte d'un épisode familial . . . . .	197



# Introduction



# Introduction

---

## 1 Motivation

**L'étude des trajectoires.** La notion de trajectoire fait l'objet de nombreux travaux en informatique, notamment depuis l'essor des technologies à base de localisation par GPS d'objets mobiles [Güting and Schneider, 2005] de toutes sortes (véhicules, humains, animaux, *etc.*) qui a pour conséquence la collecte et le stockage de quantités de données spatiales toujours plus importantes. Lorsque les objets mobiles étudiés sont des personnes, l'un des principaux enjeux est de modéliser leurs activités ou de les reconnaître à partir de données GPS brutes. La plupart du temps, on s'intéresse à la notion de déplacement, effectué le plus souvent sur une courte période, par exemple les déplacements journaliers ou hebdomadaires [Sidonie et al., 2010]. Dans ce cadre, la détection d'activités [Laube and Purves, 2006] consiste à mettre en œuvre des méthodes statistiques ou d'analyse spatiale, ainsi que des techniques de visualisation scientifique, afin d'extraire des patrons ou *patterns* d'activités à partir de trajectoires spatio-temporelles brutes, c'est-à-dire d'une succession de points (des localisations spatio-temporelles) issus de capteurs de type GPS.

**Les trajectoires de vie.** La notion de *trajectoire de vie* est utilisée en sciences sociales pour analyser les trajectoires des individus dans leur contexte. La trajectoire de vie traduit l'utilisation d'une approche biographique : il s'agit de mettre en perspective dans le temps les informations sur les individus pour mieux les comprendre. Pour Hélardot [2006], la trajectoire de vie, ou parcours de vie, "peut être considéré comme un entrecroisement de multiples lignes biographiques plus ou moins autonomes ou dépendantes les unes des autres." Chacune de ces lignes biographiques correspond à un domaine de l'existence, par exemple, "le parcours scolaire, le rapport au travail et à l'emploi, la vie familiale, la vie sociale, la santé, la trajectoire résidentielle, l'itinéraire politique, religieux ou spirituel, *etc.*". Ainsi, les thématiques portées par la trajectoire de vie, et les motivations pour l'étudier, dépendent d'un domaine d'application.

Une des applications de l'étude des trajectoires de vie concerne l'étude du choix résidentiel.

**Le cas de l'étude de la trajectoire résidentielle.** Les métropoles doivent faire face à plusieurs défis liés aux choix résidentiels de leurs habitants. Ainsi, les couples avec enfants privilégient généralement un habitat éloigné des centres métropolitains, tandis que ceux-ci attirent en plus grande proportion à la fois les jeunes nouvellement indépendants et les retraités [Robette et al., 2012]. Ce constat est la source de quelques-uns des problèmes que les villes métropolitaines doivent résoudre. Cet exode des couples actifs avec enfants vers les zones périurbaines entraîne pour les villes qui les accueillent des difficultés à fournir les aménités nécessaires (par exemple, des crèches et des écoles), mais aussi une augmentation du temps de trajet domicile-travail, générateur de pollution et de congestion du trafic. La diversification de la population des centres villes et de la périphérie des métropoles est donc un enjeu important, et il est essentiel pour les décideurs politiques et les spécialistes de l'aménagement urbain d'acquérir une meilleure connaissance des dynamiques résidentielles.

De nombreux géographes ont abordé cette problématique. Le cadre de vie, notamment ce qui est lié aux caractéristiques de la résidence et de son voisinage, ainsi que d'autres éléments relevant des conditions économiques doivent également être envisagés [Lelévrier, 2014]. Il s'agit notamment de tenir compte de la perception qu'ont les individus de ces différents éléments et de leur impact sur les choix résidentiels : un cadre de vie perçu comme anxiogène, des conditions économiques ressenties comme favorables à l'investissement immobilier dans un autre quartier, *etc.* Dans le texte introductif du séminaire ESO [Madoré and Pihan, 2003] trois critères déterminants sont présentés comme essentiels dans le choix résidentiel : le statut d'occupation, le type d'habitat et la localisation. Pour ce dernier point en particulier, ces chercheurs invitent à porter une attention particulière aux différents cycles de la vie, notamment les cycles longs (professionnels, familiaux, ...). Cette idée de mettre en perspective le choix résidentiel a pris une place importante dans les travaux récents sur les stratégies résidentielles. Les raisons du choix résidentiel d'un individu ou d'un ménage donné doivent être appréhendées en prenant en compte les aspects familiaux et professionnels, mais aussi ceux relevant de la sphère des loisirs et de tous les aspects de la vie qui sont déterminants dans ce choix. Le choix résidentiel doit donc être étudié dans le contexte d'une approche globale des trajectoires de vie des individus.

**Les trajectoires de vie en informatique** En informatique, les travaux initiés par Marius Thériault font référence dans le domaine de la modélisation de trajectoires de vie. Le modèle spatio-temporel pour l’analyse des trajectoires de vie [Thériault et al., 1999, 2002], dédié à l’étude des trajectoires résidentielles, repose sur trois trajectoires chacune caractérisant une thématique différente (résidentielle, professionnelle et familiale). Il n’existe cependant pas, à notre connaissance, d’approche d’étude générique, au sens de indépendante d’un cas d’application, permettant la modélisation, l’acquisition et l’analyse de trajectoires de vie selon des dimensions pouvant être définies en fonction des besoins.

**Exploiter de nouveaux paradigmes** En informatique, parmi les récents bouleversements qu’ont permis l’avènement des technologies mobiles, Internet ou encore les données satellitaires, la *Volunteered Geographic Information* (VGI ou géographie participative), terme introduit par Goodchild [2007], marque le début d’une profonde transformation dans la manière selon laquelle l’information géographique, c’est-à-dire toute information référencée dans l’espace (par exemple sur Terre par des coordonnées en latitude et longitude), et la connaissance qui en découle, est produite et partagée. Pour s’en convaincre, il suffit de considérer le succès d’un projet de VGI comme Open Street Map (OSM) dont le but est de cartographier le monde et d’en proposer des représentations cartographiques libres de droit et exploitables par tous. De nombreux travaux montrent que la VGI ou géographie participative est déjà largement utilisée à des fins de recueil d’informations, notamment depuis les citoyens à destinations des pouvoirs publics chargés de l’administration d’un territoire [Noël et al., 2014].

Un autre paradigme d’intérêt pour l’étude des trajectoires de vie est le Web sémantique. Ce web de données ouvertes et connectées laisse entrevoir de nouvelles possibilités en termes de modélisation et d’analyse de trajectoires. Celui-ci peut en effet être utile tant comme une source de connaissances, par la réutilisation d’ontologies, que comme une source pour l’enrichissement des données, par les croisements d’informations qu’il permet d’envisager.

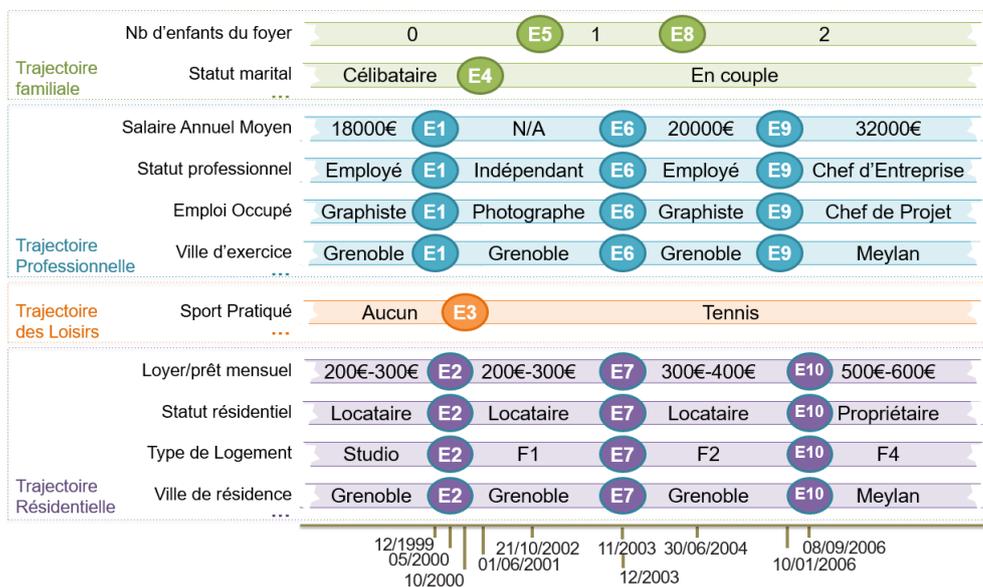
## 2 Description du problème

L’étude des trajectoires de vie pose certain nombre de défis : leurs thématiques multiples, le fait qu’elles soient des trajectoires *métaphoriques* [Spaccapietra et al., 2008], *géographiques* ou *agéographique*, leur nature *discrète* [Yattaw, 1999a], le fait qu’elles aient un *sens* [Authier et al., 2012] ainsi que

leurs multiples niveaux de granularité, temporel, spatial et thématique. Nous décrivons ici ces particularités et les défis qui y sont liés pour la modélisation, l'acquisition et l'analyse de données les décrivant.

**Des trajectoires multi-thématiques** La Figure 1.1 représente un extrait de la trajectoire de vie d'un individu, inspiré par les travaux de Thériault et al. [1999]. La trajectoire est ici représentée comme une succession de statuts (*status* pour Thériault et al. [1999]) correspondant à des attributs décrivant la "vie" de cet individu. Ces statuts sont ici regroupés selon les divers aspects qu'on souhaite observer de la vie d'une personne (tels que résidentiel, familial, professionnel, loisirs, etc.)

Ces différents regroupements forment différentes trajectoires ciblant une thématique particulière. La trajectoire de vie est donc *multidimensionnelle*, chaque dimension correspondant à un focus *thématique* sur cette trajectoire de vie. On parlera ainsi d'un ensemble de *trajectoires thématiques*, et, par exemple, de trajectoire résidentielle représentée par les statuts *est propriétaire*, *loyer/prêt*, *type d'habitation* et *ville* (voir figure 1.1). Tout statut, et, par suite, trajectoire, est intrinsèquement temporel(le) car lié(e) à une ligne de temps.



**Figure 1.1** – Extrait d'une trajectoire de vie centrée sur l'aspect résidentiel. Représentation inspirée de Thériault et al. [1999]

Les thématiques d'intérêt sont différentes selon le cas d'application et dépendent du choix du spécialiste de domaine qui souhaite étudier une trajectoire particulière en fonction de son objectif. Le défi est ici de caractériser

*génériquement* des trajectoires thématiques observables selon différents points de vue, ceci dans l'objectif de répondre à des cas d'application divers. Dans cette optique, cette généralité est nécessaire, non seulement pour le *modèle* de trajectoire de vie, mais aussi pour l'ensemble d'une approche d'étude des trajectoires, c'est-à-dire également pour l'acquisition des données et leur exploitation.

**Des trajectoires métaphoriques** Dans l'expression *trajectoire de vie*, et par extension, pour chacune des *trajectoires* thématiques qui la composent, le terme de *trajectoire* est utilisé dans un sens métaphorique [Spaccapietra et al., 2008]. Pour Spaccapietra et al. [2008], une trajectoire métaphorique représente l'évolution de la position d'objet dans un *espace abstrait* au cours du temps. Une trajectoire métaphorique très simple est représentée sous la forme d'un attribut dont la valeur varie au cours du temps. Par exemple, la valeur de l'attribut *situation professionnelle* pourrait représenter la trajectoire d'un individu dans l'espace abstrait *monde du travail* (dans lequel la position dans le monde du travail serait réduite à la situation professionnelle) [Spaccapietra et al., 2008]. Cependant, ce modèle de trajectoire devient vite insuffisant si la trajectoire se complexifie. En effet, un seul attribut est insuffisant pour modéliser une thématique. De plus, la trajectoire de vie est composée de multiples trajectoires thématiques.

Une trajectoire thématique peut être une trajectoire *géographique*. C'est le cas de la trajectoire résidentielle (voir figure 1.1) car elle est constituée d'un attribut contenant une information spatiale, ici la *Ville de résidence*, mais qui pourrait être considérée à une autre échelle spatiale, par exemple un quartier ou des coordonnées  $x, y$ . Le terme de trajectoire est ici utilisé à la fois dans un sens *géographique* (localisation des différents lieux de résidence), et aussi, (surtout) par métaphore pour désigner l'évolution de la situation résidentielle d'un individu, étant entendu que cette situation résidentielle n'est pas caractérisée que par la localisation du logement, mais aussi par d'autres éléments (montant du loyer, type de logement). D'autres aspects de la vie de l'individu contiennent une composante géographique, par exemple la trajectoire *professionnelle* peut porter notamment la localisation de son lieu de travail. Dans ces cas, une trajectoire thématique, est à la fois *métaphorique* et *géographique*. Nous parlons dans ce travail de trajectoire *métaphorique géographique*, ou simplement de trajectoire *géographique*, toute trajectoire thématique étant ici une trajectoire *métaphorique*.

Une trajectoire *métaphorique* peut également ne pas avoir de dimension correspondant à l'espace géographique. Lorsque c'est le cas, nous considé-

rons alors que la trajectoire métaphorique est *agéographiques*. C'est le cas par exemple de la trajectoire familiale (voir figure 1.1). Dans ce travail, nous parlons alors de trajectoires *métaphoriques agéographiques* ou simplement de trajectoires *agéographiques*.

L'enjeu est ici de modéliser ces *trajectoires métaphoriques agéographiques*, mais également de les modéliser de manière générique avec les *trajectoires métaphoriques géographiques* de façon à pouvoir les analyser sur un même plan.

**Des trajectoires qui ont un sens** L'utilisation métaphorique du terme trajectoire ne s'arrête pas à la référence à un espace abstrait. A ce titre, [Authier et al. \[2012\]](#) font remarquer à propos des trajectoires résidentielles que l'utilisation du terme de *trajectoire* ne fait pas référence qu'à l'aspect géographique de la vie résidentielle des individus. En effet, pour les sociologues [Grafmeyer and Authier \[2011\]](#), parler de trajectoire est plus spécifique que de décrire une simple évolution dans le temps et l'espace géographique : "l'utilisation du terme de *trajectoire* plutôt que celui d'*itinéraire* ou de *mobilité*, revient à suggérer qu'une série donnée de positions successives n'est pas le simple fait du hasard, mais s'enchaîne au contraire selon un ordre intelligible." Le degré de maîtrise d'un individu sur sa trajectoire, qui résulte de ses contraintes et ses possibilités, diffère selon chaque cas, et il convient de mettre en évidence les conditions dans lesquelles les choix sont effectués. Plus largement, il s'agit d'éliciter les facteurs ou les combinaisons de facteurs qui ont contribué à modéliser une trajectoire donnée. Il faut donc donner du sens à la trajectoire résidentielle, comprendre les raisons qui poussent les individus à changer de résidence, à quel moment de leur vie et pour quel (autre) espace.

Une approche plus générale de cette notion est faite par [Becker Howard \[1985\]](#). En effet, ce que nous cherchons ici à mettre évidence se rapproche de la notion de *carrière* telle qu'elle a été définie par cet auteur : "la carrière désigne les facteurs dont dépend la mobilité d'une position à l'autre, c'est-à-dire aussi bien les faits objectifs relevant de la structure sociale que les changements dans les perspectives, les motivations et les desseins des individus". Le terme de carrière est ici utilisé dans un sens plus large que celui de carrière professionnelle et peut s'appliquer aux différentes thématiques d'une trajectoire de vie.

Ainsi, l'enjeu est ici de donner du *sens* aux trajectoires de vie, en modélisant les facteurs qui influencent un changement de situation dans une trajectoire de vie, que ceux ci dépendent de l'environnement de la trajectoire ou de la trajectoire de l'individu elle même. Il convient également d'être capable

d'exploiter ces facteurs dans une phase d'analyse des trajectoires.

**Des trajectoires discrètes** Les trajectoires, qu'elles soient géographiques ou géographiques, peuvent être continues ou discrètes [Yattaw, 1999b].

On peut cependant remarquer qu'un mouvement dans l'espace géographique est forcément continu même si, bien sûr, du fait des dispositifs d'enregistrement notamment, les informations disponibles sur ces trajectoires sont discrètes. Cela n'est cependant pas forcément le cas pour l'*espace abstrait* d'une trajectoire métaphorique. Cet *espace abstrait*, peut être discret ou continu, la nature du mouvement étant donc elle-même discrète ou continue selon l'espace. Or, dans un espace discret, pour caractériser une évolution, il est important de modéliser les *transitions* entre deux positions de cet espace. Par exemple, sur la figure 1.1, on peut considérer que la trajectoire professionnelle est une succession de positions dans un espace abstrait caractérisé par quatre attributs (*salaire annuel moyen*, *statut professionnel*, *emploi occupé*, *Ville d'exercice*). Dans cet espace abstrait, il est nécessaire de caractériser les transitions entre deux emplois pour comprendre comment ils se sont succédés. Ces transitions correspondent à des événements de vie des individus, en l'occurrence des changements d'emploi, (sur la figure il s'agit des événements *E1*, *E2* et *E3*).

L'enjeu est donc ici de caractériser les événements de la trajectoire de vie des individus.

**Des trajectoires multi-granulaires** La notion de *granularité* [Silva et al., 2015] fait référence au fait que le monde est perçu à différents niveaux de précision. La perception d'un niveau de granularité est celle d'un *point de vue* sur le monde dont la particularité est de correspondre à un niveau de précision. Dans la lignée des travaux tels que ceux de Hornsby and Egenhofer [2002], on ne peut ignorer l'importance des niveaux de granularité temporelle, spatiale ou thématique pour la représentation d'objets mobiles. Les *niveaux de granularité* auxquels sont représentées les informations déterminent les phénomènes qui sont révélés. Les enjeux soulevés par la problématique de la granularité concernent à la fois la *modélisation*, la *collecte* et l'*exploitation* de trajectoires de vie.

Un premier enjeu est de définir la granularité temporelle adaptée à la modélisation de la trajectoire de vie. Pour [Jensen et al., 1998], un *chronon* est une durée de temps minimum, la plus petite unité temporelle considérée pour marquer temporellement un événement. Dans le cas de la trajectoire de vie, un chronon pertinent est le jour, cette unité permettant de situer des événements

marquants avec une précision qui fait sens pour un individu (comme la date d'un mariage, de la naissance d'un enfant, *etc.*). À l'échelle d'une vie, il est cependant utile de retenir un chronon d'une granularité moins fine telle que le mois ou l'année pour situer un événement pour lequel la précision en termes de *jour* est impossible (oubli de la date précise) ou non requise (par exemple, pour situer quand a débuté le projet de déménager). Le modèle doit donc supporter des degrés de précision différents dans la datation.

Un raisonnement similaire est appliqué pour déterminer le niveau de granularité auquel il convient de considérer l'information de nature géographique. En effet, selon l'intérêt du spécialiste de domaine étudiant une trajectoire de vie, l'information concernant la *localisation* géographique d'un individu peut être requise au niveau de son adresse exacte ou bien de sa ville de résidence. Le même problème se pose concernant l'information thématique (au sens de [Peuquet \[1994\]](#)). Par exemple, une information concernant l'emploi occupé par un individu peut être considérée au niveau de la profession exacte, du poste, ou de la catégorie socio-professionnelle.

Toutes ces informations, qu'elles soient spatiales, temporelles ou thématiques doivent donc être stockées/collectées/analysées à différents niveaux de granularité. Le modèle de trajectoires de vie doit donc être en mesure de caractériser ces différents niveaux. Lors de la collecte de donnée, il doit également être possible de contribuer à chacun des différents niveaux modélisés. Concernant l'exploitation des données, celle-ci doit également être possible à différents niveaux de granularité.

**Nécessite d'une approche globale de l'étude des trajectoires de vie** Il n'existe pas à notre connaissance d'approche globale permettant la collecte, le stockage, et l'analyse des trajectoires de vie. Une approche générique, indépendante d'un cas d'application, est requise. Cette approche doit donner à un expert de domaine, souhaitant étudier des trajectoires de vie *i)* la possibilité de créer un modèle prenant en compte ses thématiques d'intérêt *ii)* de collecter des données conformes à ce modèle *iii)* de les analyser selon leurs différentes dimensions (personnelles, professionnelles, socio-culturelles, *etc.*) et différents niveaux de granularité.

### 3 Organisation du manuscrit

Ce manuscrit de thèse est divisé en deux parties. La première présente, en trois chapitres, un état de l'art du domaine dans lequel nos travaux s'ins-

crivent. La seconde, comprenant également trois chapitres, présente nos contributions, qui pallient certaines des limites identifiées dans l'état de l'art.

### 3.1 Partie I : état de l'art

Après ce chapitre introduction, le chapitre II se concentre sur les travaux sur les *trajectoires sémantiques* [Alvares et al., 2007], des premiers travaux inspirés de la *time-geography* aux ontologies de trajectoires permettant une intégration dans le web sémantique.

Le chapitre III aborde la notion de point de vue et particulièrement les modèles de trajectoires prenant en compte de multiples points de vue. Les travaux sur les trajectoires de vie font partie de ces modèles.

Le chapitre IV propose un récapitulatif des limites des travaux présentés dans l'état de l'art. À partir de ce récapitulatif, nous présentons la façon dont nos travaux contribuent à lever les verrous scientifiques identifiés.

### 3.2 Partie II : contributions

Le chapitre V présente un patron de conception d'ontologie de trajectoires de vie, ainsi qu'un modèle de facteurs explicatifs pour les événements de vie. Le patron de conception permet, de créer un modèle caractérisant une information multigranulaire appartenant à de multiples thématiques. La notion de facteur explicatif permet de caractériser ce qui influence les événements de vie. C'est sur ces deux modèles que repose l'approche décrite et appliquée dans les chapitres suivants.

Le chapitre VI est consacré à la description de l'approche qui permet de modéliser, collecter et analyser des trajectoires de vie, indépendamment d'un cas d'application. Nous y décrivons également l'architecture logicielle qui concrétise notre approche. Concernant la modélisation, nous décrivons comment notre patron peut être appliqué successivement pour créer un modèle de trajectoires de vie dédié à un cas d'application grâce notamment à la réutilisation d'ontologies. Pour l'acquisition des données, notre approche constitue à la fois un support pour une insertion facilitée des données dans un *Triple Store* ainsi que pour la création d'une interface de collecte de trajectoires de vie. Enfin, pour l'exploitation des données, elle met à profit le web sémantique et propose

des solutions génériques d'exploitation des trajectoires collectées en étendant SPARQL<sup>1</sup>, le langage de requêtes du Web sémantique.

Le chapitre VII est consacré à l'application de notre approche pour l'étude des trajectoires résidentielles. Nous montrons, en appliquant notre approche : *i*) la création d'une ontologie de trajectoire de vie adaptée à ce cas d'application *ii*) la mise en place d'une collecte de donnée sur ces trajectoires *iii*) l'analyse des données collectées en utilisant les opérateurs génériques précédemment définis.

Dans le dernier chapitre de ce mémoire, nous dressons le bilan de ce travail et nous en présentons les perspectives.

---

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

# Etat de l'art



# Trajectoires sémantiques

---

## Sommaire

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>21</b>
<b>2</b>	<b>Origines des trajectoires sémantiques</b> . . . . .	<b>22</b>
2.1	Des trajectoires spatio-temporelles brutes aux approches basées sur la sémantique . . . . .	22
2.2	La time-geography . . . . .	24
2.3	Conclusion . . . . .	26
<b>3</b>	<b>Modèles de trajectoires sémantiques</b> . . . . .	<b>26</b>
3.1	Introduction . . . . .	26
3.2	Un modèle basé sur une structuration en déplacements et arrêts . . . . .	27
3.3	Autres modèles conceptuels de trajectoires . . . . .	29
3.4	Conclusion . . . . .	31
<b>4</b>	<b>Web sémantique et ontologies de trajectoires</b> . . . . .	<b>31</b>
4.1	Introduction . . . . .	31
4.2	Technologies du web sémantique et ontologies spatio-temporelles . . . . .	32
4.3	Ontologies de trajectoires sémantiques . . . . .	39
4.4	Conclusion . . . . .	41
<b>5</b>	<b>Conclusion</b> . . . . .	<b>42</b>

---

## 1 Introduction

La notion de *trajectoires sémantiques* [Alvares et al. \[2007\]](#) est utilisée pour enrichir des données de mobilité brutes qui se présentent habituellement sous forme d'une trajectoire spatio-temporelle formée par une suite de triplets (latitude, longitude, temps), avec des informations susceptibles d'y apporter du sens. La modélisation est une étape nécessaire vers une exploitation des trajectoires et la mise en place de solutions de découverte de la connaissance qui

peut en être issue [Bogorny et al., 2014]. Les premiers travaux sur l'étude des trajectoires proposent des modèles permettant le requêtage de données brutes [Bogorny et al., 2014]. Cependant, en l'absence de davantage d'informations liées aux trajectoires, les possibilités d'analyse et de réponse à des requêtes intégrant de la sémantique sont limitées. Pour combler ces lacunes, de nombreux travaux ont intégré de la sémantique dans les modèles conceptuels utilisés pour gérer ces données de mobilités [Zheni et al., 2009, Spaccapietra et al., 2008, Andrienko et al., 2011]. La *time-geography* de Hägerstraand [1970], qui propose un framework conceptuel associant le temps et l'espace, est largement utilisée pour structurer les trajectoires sémantiques.

Dans ce chapitre, nous présentons les intérêts et limites des différents travaux sur les trajectoires sémantiques pour la modélisation de trajectoires de vie. Dans la première section, nous explorons les origines des trajectoires sémantiques en présentant les fondements de ces modèles et en analysant le *framework* conceptuel de la *time-geography*. Dans la seconde section, nous présentons les principaux modèles de trajectoires sémantiques.

## 2 Origines des trajectoires sémantiques

Dans cette section, nous nous intéressons aux origines des modèles de trajectoires sémantiques et à ce qu'elles nous indiquent sur leurs capacités de modélisation. Dans la première sous-section nous évoquons les premiers travaux qui définissent cette notion. Dans la seconde sous-section, nous présentons la *time-geography* [Hägerstraand, 1970] qui fournit un framework conceptuel très utilisé pour la modélisation de trajectoires sémantiques.

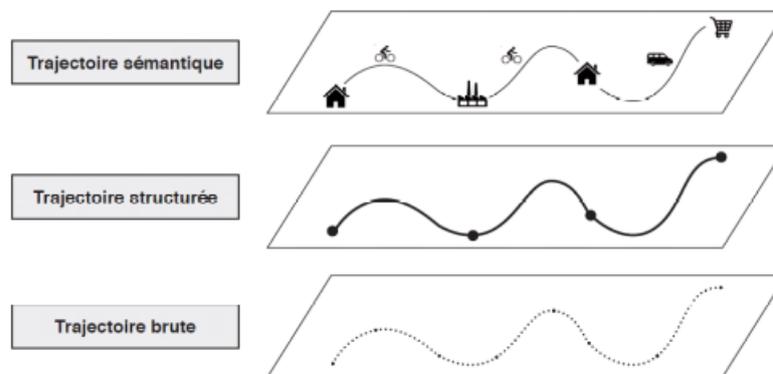
### 2.1 Des trajectoires spatio-temporelles brutes aux approches basées sur la sémantique

L'étude des trajectoires spatio-temporelles est devenue un champ de recherche important avec l'émergence des technologies de positionnement et de télécommunication [Li et al., 2006]. En effet, de plus en plus de données de localisation estampillées temporellement peuvent aujourd'hui être collectées. Ces séries spatio-temporelles forment des trajectoires, que les spécialistes de nombreux domaines souhaitent analyser.

En 2006, Li et al. [2006] désignent comme *approches basées sur la sémantique*, celles modélisant les trajectoires non plus comme un ensemble de localisations mais comme un ensemble de *points clés* (key point). Les points clés

regroupent un ensemble de localisations d'une trajectoire spatio-temporelle (par exemple des coordonnées GPS brutes) selon des critères sémantiques. De nombreuses approches utilisent les notions d'*activité* et *déplacement* pour structurer ces trajectoires (voir sous-section suivante). On passe alors de trajectoires composées de localisations (pouvant être caractérisées par des coordonnées géographiques) à des trajectoires composées de points clés, caractérisant une activité ou un déplacement, qui portent en eux une caractérisation sémantique. Cette structuration permet également de diminuer les volumes de données, ce qui réduit les problèmes de scalabilité. En effet, un point clés peut regrouper de nombreuses localisations d'une trajectoire spatio-temporelle et le stockage des différentes localisations enregistrées initialement n'est plus forcément nécessaire.

En 2007, [Alvares et al. \[2007\]](#) utilisent pour la première fois le terme de trajectoires sémantiques, qui devient le terme usuel pour désigner ce type de trajectoire. Depuis, de nombreux modèles de trajectoires sémantiques ont émergé, proposant différentes solutions pour enrichir les trajectoires spatio-temporelles. Il s'agit de passer de données GPS brutes, indiquant le déplacement d'un objet mobile, à une trajectoire contenant des informations utiles à sa description (cf. Figure 2.1).



**Figure 2.1** – De la trajectoire brute à la trajectoire sémantique d'après [Yan and Spaccapietra \[2009\]](#)

Ces informations peuvent, par exemple, être liées la nature du mode de transport ou encore porter sur les lieux traversés. Dans la figure 2.1, l'objet mobile qui nous intéresse est une personne, et grâce à la sémantique ajoutée à la trajectoire (ici portée par des symboles graphiques) on comprend qu'elle s'est rendue à son lieu de travail à bicyclette, et qu'elle est partie ensuite en voiture pour se rendre dans un magasin, des informations qui n'étaient pas contenues dans la trajectoire brute. Plus largement, toute information

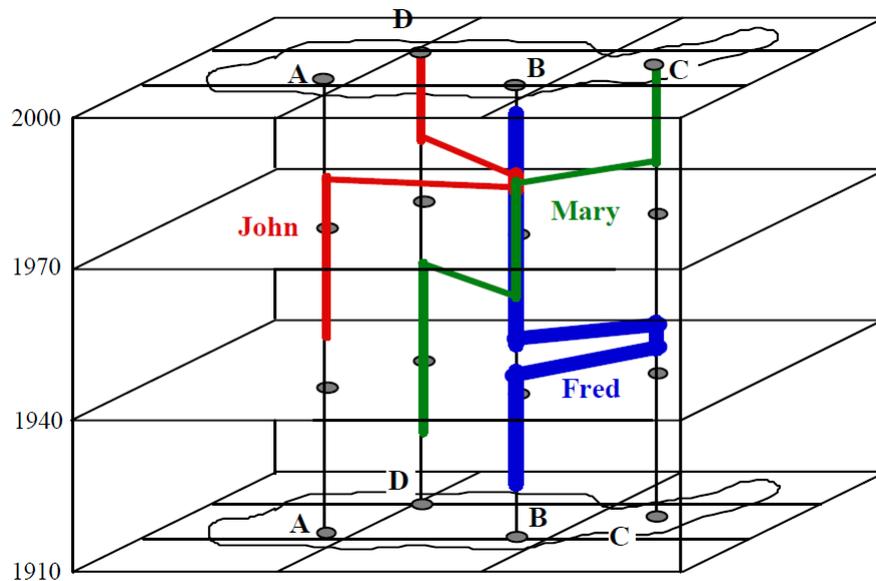
susceptible de donner du sens à une trajectoire peut être incluse dans un modèle de trajectoire sémantique.

Les approches basées sur la sémantique visent donc à associer du *sens* à des trajectoires. Ainsi, avec ces approches, on observe une translation de l'intérêt des chercheurs de l'aspect géographique des trajectoires vers leur aspect sémantique. Pour porter cette sémantique, les trajectoires géographiques sont structurées en épisodes spatio-temporels, selon une approche qui s'inspire de la *time-geography*.

## 2.2 La time-geography

Les travaux de Hägerstraand [1970] sur la *time-geography* permettent de mettre en relation le temps et l'espace. Celle-ci offre des solutions conceptuelles pour caractériser la mobilité à différentes échelles, par exemple à l'échelle journalière ou à celle de la vie. La représentation dite du *cube espace-temps*, ou *aquarium spatio-temporel* (voir figure 2.2) en est issue. Dans cette représentation, l'espace figure sur la dimension horizontale, par exemple sous la forme d'une carte, et le temps sur la dimension verticale, ce qui permet de représenter le mouvement dans l'espace géographique. Cette méthode est le plus souvent utilisée dans l'étude des trajectoires quotidiennes [Banos et al., 2005], mais certains travaux ont également exploré le potentiel de la *time-geography* pour étudier des trajectoires à l'échelle de la vie.

Dans une application au domaine de la santé, Sinha and Mark [2005] utilisent le concept de *geospatial lifeline* pour modéliser des trajectoires à l'échelle de la vie. Une *geospatial lifeline* [Mark, 1998] est ainsi définie comme un ensemble continu de positions occupées par un objet dans l'espace géographique durant une période de temps. Ce concept correspond à celui de trajectoire spatio-temporelle. Ces travaux sont les premiers à utiliser le *cube espace-temps* (voir figure 2.2) pour représenter des trajectoires à l'échelle de la vie. La localisation des résidences successives des individus est utilisée pour la dimension spatiale.



**Figure 2.2** – Visualisation de trois trajectoires de vie dans le cube espace-temps [Sinha and Mark, 2005]

Cette approche a l'intérêt d'explorer le potentiel de l'aquarium spatio-temporel, pour la représentation des trajectoires de vie, cette représentation étant habituellement réservée à des échelles plus fines. Mais cette représentation, et le modèle issu de la *time-geography* sur lequel elle repose, ont également leurs limites. Tout d'abord, le modèle de trajectoire de vie est réduit à une seule dimension spatiale : la localisation des différentes résidences. La seule sémantique liée à cette localisation est la notion de *résidence*, et, de plus, celle-ci n'est pas décrite autrement que par sa localisation. Une autre limite de ces travaux est liée au caractère discret [Yattaw, 1999b] de la trajectoire de vie. En effet, les trajectoires de mobilité quotidienne, typiquement caractérisées par les modèles issus de la *time-geography* sont des trajectoires continues. Ainsi, même si le modèle est discret, il est possible de relier deux points pour approximer de manière relativement précise des positions spatio-temporelles, selon la précision de l'étalonnage. Cependant, lorsque le phénomène à modéliser est discret, comme c'est le cas de la trajectoire résidentielle modélisée dans ces travaux, se pose la question de la transition entre deux épisodes. À ce titre, la représentation donnée (Figure 2.2) illustre ce qui constitue une faiblesse du modèle : les traits reliant les différentes résidences des individus dont la trajectoire est représentée (par exemple, pour Fred de B à C puis de C à B) ne représente pas une information pertinente d'un point de vue spatial : contrairement aux traits verticaux, chaque point de ces traits ne correspond pas à une position que Fred a effectivement occupé. Ceci renvoie à la question

de la représentation des transitions entre les différentes positions dans une trajectoire discrète, et dans une étape précédente, à celle de leur modélisation. La *time-geography* a influencée de nombreux modèles de trajectoires sémantiques, et ces approches, qui reposent sur la modélisation d'épisodes successifs ou non, ne proposent pas de modélisation pour ces transitions. La sémantique liée à ces transitions, et notamment les facteurs qui pourraient les influencer, n'est donc pas non plus caractérisée. Dans cet exemple, il s'agirait d'explications liées aux changements de résidence.

### 2.3 Conclusion

Les modèles de trajectoires sémantiques sont la conséquence d'un déplacement de l'intérêt des chercheurs d'une information géographique (des trajectoires spatio-temporelles brutes) vers une information sémantique. Ces modèles découpent les trajectoires spatio-temporelles, ce qui permet de les structurer en périodes de déplacements et d'activités en reposant sur la *time-geography* de Hägerstraand [1970]. Cependant, ils ne permettent pas de caractériser des transitions entre épisodes, ce qui est pertinent pour la modélisation de trajectoires discrètes telles que les trajectoires de vie. Dans le cas des trajectoires de vie, ces transitions correspondent à des événements de la vie de l'individu. Cela nous amène à nous intéresser dans cet état de l'art à la modélisation d'événements (voir sous-section 4.2 de chapitre).

Au préalable, dans une seconde section, nous présentons différents modèles conceptuels de trajectoires sémantiques, ainsi que les structurations sur lesquelles ils reposent. Nous étudions ce que ces modèles peuvent apporter à la modélisation de trajectoires de vie, tant sur le plan conceptuel que en terme d'approche de modélisation.

## 3 Modèles de trajectoires sémantiques

### 3.1 Introduction

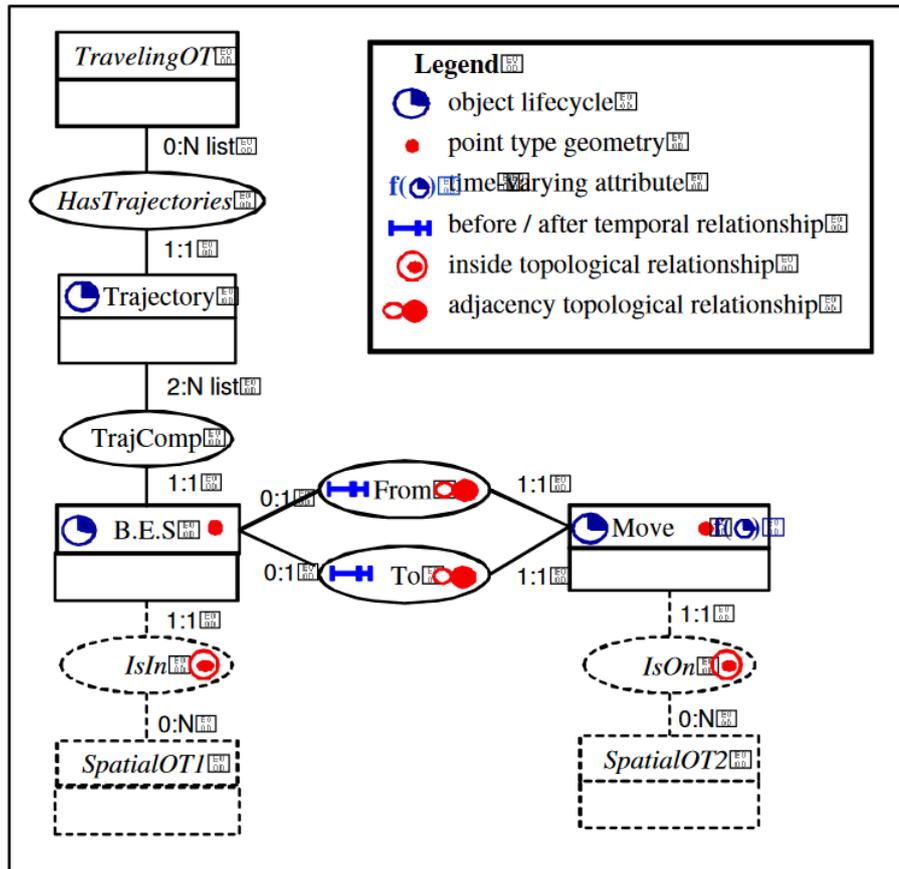
Nous présentons des modèles de trajectoires sémantiques et les approches de modélisation dans lesquels ils s'inscrivent. Pour chaque approche présentée, nous nous intéressons au méta-modèle utilisé et à sa puissance de caractérisation, aux caractéristiques du modèle lui-même, et à l'approche préconisée par les auteurs pour l'utilisation du modèle pour un cas d'application particulier. Dans la première sous-section, nous étudions un modèle [Spaccapietra et al., 2008] qui a été largement repris, ainsi qu'à la vue générale que donnent

les auteurs de ce modèle sur les trajectoires sémantiques. Nous nous intéressons également à la capacité de ce modèle à caractériser des *trajectoires métaphoriques*. Dans la seconde sous-section, nous étudions d'autres modèles conceptuels, proposant des structurations et des approches de modélisation différentes.

### 3.2 Un modèle basé sur une structuration en déplacements et arrêts

Le modèle de Spaccapietra et al. [2008] est fondateur dans le domaine de l'étude des trajectoires sémantiques. Ce modèle (présenté sur la figure 2.3) repose sur la notion de *move*, un intervalle temporel durant lequel la position de l'objet change, et de *stop* un intervalle temporel durant lequel elle reste fixe. La proposition des auteurs prend la forme d'un *design pattern* de trajectoires (voir figure 2.3) basé sur ces concepts. Ce *design pattern* utilise le modèle entité-relation, et plus particulièrement le langage MADS [Parent et al., 2006a] qui permet de représenter l'information spatio-temporelle et particulièrement les relations (topologiques et temporelles) entre les entités, à l'aide de pictogrammes. Dans ce pattern, une trajectoire est associée à une liste comprenant au moins deux entités de types B.E.S (Begin End Stop), c'est-à-dire comprenant au moins deux *stops* qui marquent respectivement le début et la fin de la trajectoire. Un *stop* est ainsi relié à 1 ou 0 *move* par une relation *from*. Un *stop* est également relié à zéro ou un *move* par une relation *to*. Ces deux relations caractérisent une double adjacence, spatiale et temporelle, entre les épisodes de la trajectoire qu'elles relient.

L'approche par *design pattern* permet à la fois une représentation claire du modèle, ainsi qu'une liberté d'application pour le modélisateur souhaitant l'étendre pour modéliser un cas particulier. En effet, les concepts (*stops*, *moves*...), leurs relations et leurs attributs spatio-temporels étant déjà définis, le modélisateur peut se concentrer uniquement sur la sémantique en choisissant les attributs à attacher aux concepts.



**Figure 2.3** – Le design pattern proposé par Spaccapietra et al. [2008]

Cette approche n’aborde pas la notion de *granularité* et la notion d’*épi-sode* (*stop* ou *move*) ne porte pas explicitement de niveau de granularité. Le problème de la caractérisation de la granularité de l’attribut est ainsi relégué à l’étape d’implémentation du modèle et, par conséquent, tributaire des capacités des Systèmes de Gestion de Bases de Données (SGBD) relationnels.

Ce modèle ne permet pas non plus de caractériser génériquement trajectoires géographiques et trajectoires *agéographiques*, car les concepts définis ici (*stops* et *moves*) sont géographiques. À ce sujet, après avoir introduit le *pattern* de la figure 2.3 les auteurs pointent également que (en anglais) : *”le terme de trajectoire est parfois utilisé dans un sens métaphorique pour décrire une évolution, bien que celle-ci ne soit pas reliée à un mouvement physique”*. Ces trajectoires sont alors appelées *trajectoires métaphoriques*, en référence à *”l’idée d’un objet (e.g. une personne) se déplaçant dans un espace abstrait”*. Ceci correspond au concept de trajectoire *métaphorique agéographique* que nous utilisons dans cette thèse. Les auteurs proposent de modéliser ces trajectoires métaphoriques en utilisant leur modèle conceptuel, basé sur les notions

de *stop* et de *move*, en transposant ces notions dans l'*espace abstrait*. Dans ce cas, une position donnée est caractérisée par un *stop* dans la trajectoire. Cependant, il semble que le concept de *move* a un sens limité dans une telle trajectoire : le mouvement caractérisé par un *move* est difficile à traduire dans l'*espace abstrait* (e.g. que signifie se déplacer d'un emploi à un autre d'un point de vue spatial et temporel?). Ainsi, dans le cas où la notion de *stops* serait transposée dans l'espace abstrait, il subsisterait la nécessité de caractériser les événements de vie qui constituent les transitions dans une trajectoire de vie.

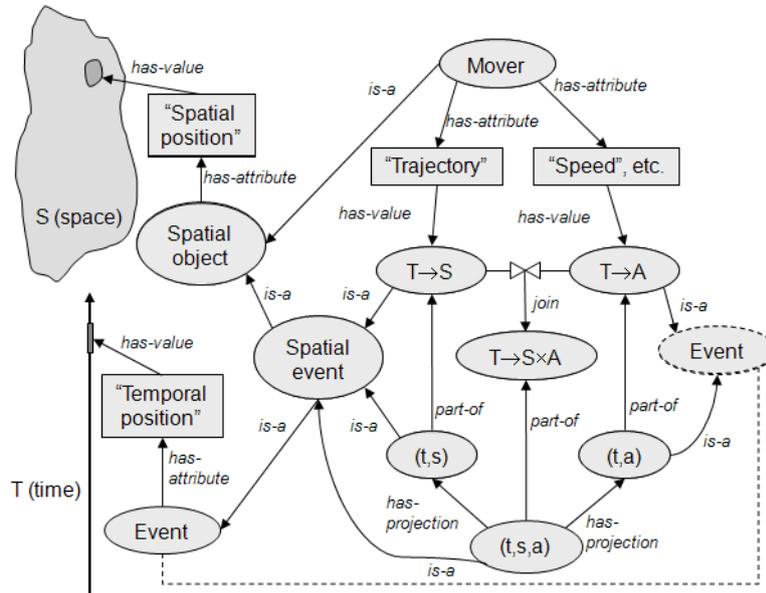
### 3.3 Autres modèles conceptuels de trajectoires

D'autres chercheurs ont utilisé les apports conceptuels de la *time-geography* pour modéliser les trajectoires sémantiques. De récents travaux visent par exemple, l'utilisation de ce *framework* dans les systèmes de gestion de base de données [Frihida et al., 2009]. Ces auteurs proposent un type de données abstrait (Abstract Data Type, ou ADT) permettant d'encapsuler une trajectoire sémantique, ainsi qu'un ensemble d'opérations permettant de les manipuler. Cette approche, une fois implémentée, permet à des bases de données d'intégrer des objets comportant des informations spatiales, temporelles et sémantiques portant sur des trajectoires. Un gestionnaire de base de données pourra ainsi simplement créer des instances de trajectoires calquées sur le modèle proposé. Le modèle conceptuel utilisé ici est basé sur la notion d'activité (*activity*) et de déplacement (*trip*). Une activité est caractérisée par sa localisation, sa position temporelle de départ, celle d'arrivée ainsi que par un attribut nommé *purpose* qui décrit la nature de l'activité et sur lequel repose la description de la sémantique de chaque partie de la trajectoire.

Cette approche peut être étendue pour proposer des objets *trajectoire* plus complexes ou correspondant à une application particulière, mais le problème de la granularité reste là aussi tributaire des capacités de caractérisation des systèmes de gestion de bases de données. De plus, les notions de *déplacement*, et d'*activité* ont ici une forte connotation géographique.

Andrienko et al. [2011] proposent un modèle permettant de caractériser des trajectoires sémantiques spatio-temporelles comme une composition d'événements (voir figure 2.4). Le mouvement est ainsi considéré comme une combinaison d'événements spatio-temporels dont l'étendue dans le temps et l'espace est variable. L'entité *Mover*, celle dont la trajectoire est décrite, a un attribut *Trajectory* qui caractérise une trajectoire. Une *trajectoire* est une fonction de  $T \rightarrow S$  où  $S$  est un ensemble de localisations spatiales et  $T$  un ensemble de localisations temporelles. Une trajectoire correspond donc à un ensemble paires

$(t,s)$  avec  $t$  appartient à  $T$  et  $s$  appartient à  $S$ .



**Figure 2.4** – Modèle conceptuel de trajectoire basé sur des événements [Andrienko et al., 2011]

L'entité *Mover* possède aussi des attributs, appelés *attributs thématiques*. La valeur d'un attribut thématique est elle aussi caractérisée par une fonction de  $T \rightarrow A$ , où  $A$  correspond aux valeurs possibles d'un attribut. Ces valeurs correspondent donc également à un ensemble de paires  $(t,a)$ . Une jointure de ces deux fonctions peut être effectuée en se basant sur la dimension temporelle et ceux-ci peuvent alors être vus comme un ensemble de triplets  $(t,s,a)$ .

Le modèle est ensuite appliqué aux mouvements d'animaux sauvages. Les événements spatio-temporels sont alors des extraits de trajectoires de ces animaux. Cette méthode est particulièrement adaptée pour des trajectoires spatio-temporelles discrètes. Le modèle permet de mettre en évidence la relation des animaux avec des localisations ayant des propriétés particulières, choisies par des experts du domaine (par exemple, des aires non couvertes par la forêt). L'approche permet donc d'extraire des relations entre le mouvement et son contexte spatio-temporel.

Celle-ci n'est cependant pas adaptée pour caractériser des trajectoires agéographiques. Si ce modèle est original, la notion d'*événement*, dotée d'une empreinte spatiale et d'une empreinte temporelle, est finalement assez proche de celle d'*épisode* et ne permet pas de caractériser des transitions ou des trajectoires métaphoriques agéographiques.

### 3.4 Conclusion

Dans cette section, nous avons présenté différentes approches de modélisation de trajectoires sémantiques. Ces modèles diffèrent tant sur le plan conceptuel (nature des structurations proposées) qu’au niveau des techniques de modélisation utilisées. Ces modèles, qui sont adaptés pour la modélisation de trajectoires géographiques quotidiennes, ne permettent pas de caractériser des trajectoires géographiques. Aucun ne permet non plus de caractériser une granularité à la fois spatiale, temporelle et thématique. Spaccapietra et al. [2008], Frihida et al. [2009], Andrienko et al. [2011] présentent cependant des approches de modélisation de haut niveau et indépendante d’un cas d’application. L’approche par *design pattern* se distingue par la liberté qu’elle donne d’adapter le modèle à un cas d’application particulier, ce qui permet d’ajouter une sémantique dépendante du cas d’application. Cette approche est donc idéale pour modéliser des thématiques différentes, et elle paraît particulièrement pertinente pour une modélisation de trajectoires de vie qui sont elles-mêmes porteuses de multiples thématiques.

D’autres chercheurs ont proposés des modèles de trajectoires sémantiques prenant la forme d’ontologies afin de tirer avantage du Web sémantique.

## 4 Web sémantique et ontologies de trajectoires

### 4.1 Introduction

L’utilisation d’ontologies pour la modélisation de trajectoires sémantiques répond à la fois au besoin de caractériser une sémantique de plus en plus riche et à celui de croiser les données de trajectoires avec des données externes afin d’en tirer de nouvelles connaissances. De plus en plus d’ontologies et de données sont publiées par des organismes publics ou privés [Shadbolt et al., 2006]. En France, on peut par exemple citer l’Institut National de la Statistique et des Études Économiques (INSEE) ou l’Institut National de l’Information Géographique et Forestière (IGN) qui publient respectivement la nomenclature des catégories socio-professionnelles<sup>1</sup> et l’ontologie geofla décrivant les unités administratives, que nous utilisons dans ces travaux.

Dans la première sous-section, nous présentons en quoi le Web sémantique peut être utilisé pour la modélisation et l’analyse de trajectoires. Dans la seconde sous-section nous examinons les intérêts et les limites des ontologies de trajectoires existantes.

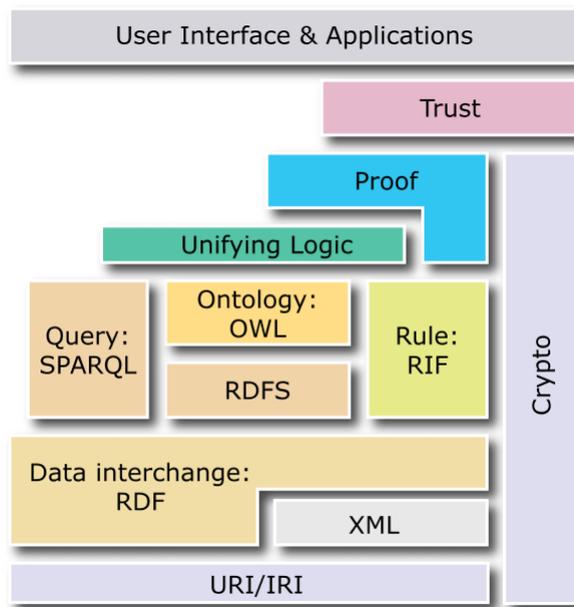
---

1. <https://www.insee.fr/fr/metadonnees/definition/c1493>

## 4.2 Technologies du web sémantique et ontologies spatio-temporelles

**Technologies du web sémantique et Linked Open Data Cloud** Le terme de web sémantique est introduit en 2001 par [Berners-Lee et al. \[2001\]](#). Il désigne, selon le W3C<sup>2</sup>, un *framework* conceptuel permettant la mise en place sur le web de données liées, partagées et réutilisables aussi bien par des machines que par des humains. Dans cette sous-section, nous présentons le Web sémantique en insistant sur deux points particulièrement importants pour la modélisation de trajectoire : le Linked Open Data Cloud (LOD Cloud) et les ontologies décrivant des informations spatio-temporelles.

Dans le Web sémantique, les ressources, qui caractérisent toutes sortes d'objets ou de concepts, sont identifiées par des URIs [[Shadbolt et al., 2006](#)]. L'ensemble des technologies (parfois appelé *semantic web stack*) à mettre en œuvre pour permettre le fonctionnement à grande échelle du Web sémantique est présenté sur la figure 2.5.



**Figure 2.5** – Les technologies du Web sémantique

Certaines briques sont déjà fonctionnelles, tandis que d'autres sont encore en phase de conception. Les *URIs* permettent de désigner les *ressources*, composantes de base et support de l'information dans le Web sémantique. Cet identifiant unique permet aux ressources une accessibilité via la protocole HTTP<sup>3</sup> (*Hypertext Transfer Protocol*). Ces ressources sont décrites dans

2. <https://www.w3.org/standards/semanticweb/>

3. <https://www.w3.org/Protocols/>

un langage, par exemple RDF<sup>4</sup> (*Ressource Description Framework*), qui peut être sérialisé en XML (*Extensible Markup Language*). Les langages RDFS<sup>5</sup> et OWL<sup>6</sup> (*Web Ontology Language*) sont utilisés pour la description des données et la définition de modèles qui les caractérisent. Ces modèles sont appelés *ontologies*. Le langage OWL est basé sur des logiques de description. Les ontologies, librement accessibles permettent l'utilisation des données qu'elles décrivent. Des requêtes peuvent, par exemple, être effectuées sur celles-ci par grâce au protocole et langage de requête SPARQL<sup>7</sup> (SPARQL Protocol and RDF Query Language).

Les ontologies peuvent également être réutilisées. C'est là un élément clé du Web sémantique qui permet d'uniformiser la publication des données et de faciliter le travail des modélisateurs, qui peuvent utiliser tout ou partie d'une ontologie déjà créée dans leur propre modèle.

Schmachtenberg et al. [2014] décrit l'état du Linked Open data Cloud (LOD Cloud, nuage de données ouvertes et liées (voir figure 2.6)). Cette implémentation du web sémantique contenait alors plus d'un millier de jeux de données rattachées à des domaines variés (données publiques, médias, réseaux sociaux...).

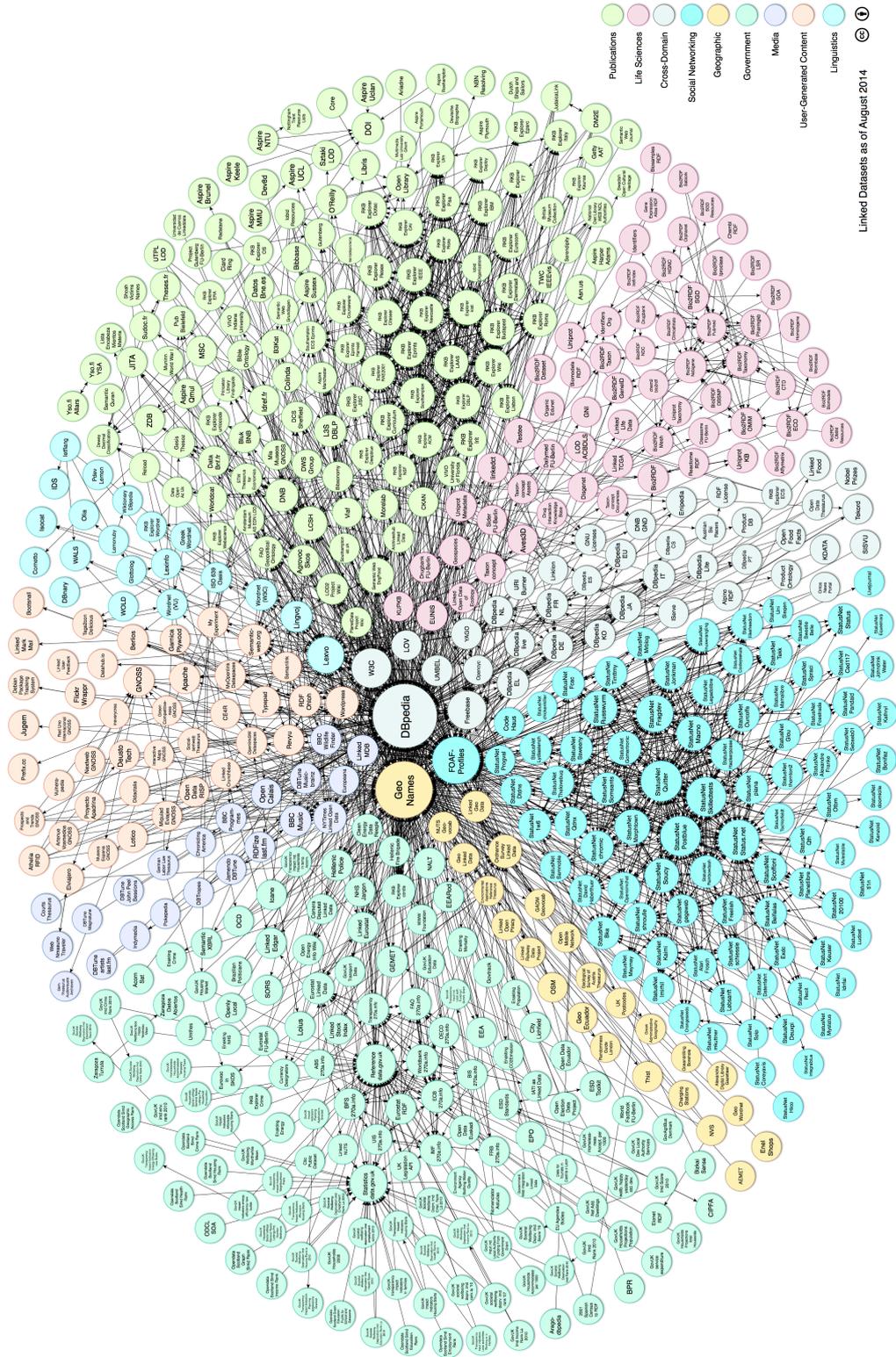
---

4. <https://www.w3.org/RDF/>

5. <https://www.w3.org/TR/rdf-schema/>

6. <https://www.w3.org/OWL/>

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

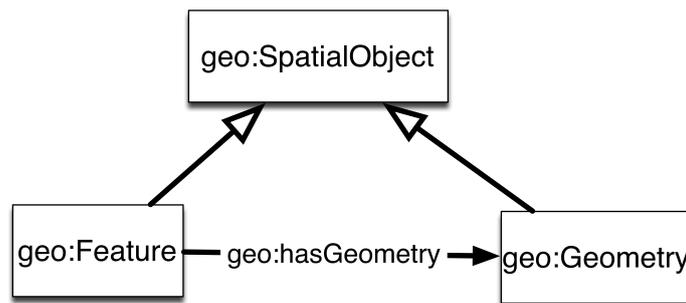


Linked Datasets as of August 2014

Figure 2.6 – Le Linked Open Data Cloud en 2014 [Schmachtenberg et al., 2014]

**Ontologies pour décrire l'espace et le temps** Il existe des ontologies de référence pour décrire l'espace et temps. Ces ontologies sont donc réutilisées dans des ontologies de trajectoires ce qui permet de caractériser les dimensions spatiales et temporelles des trajectoires et d'assurer une interconnexion avec des données externes dont les composantes spatiales et temporelles sont décrites avec les même ontologies.

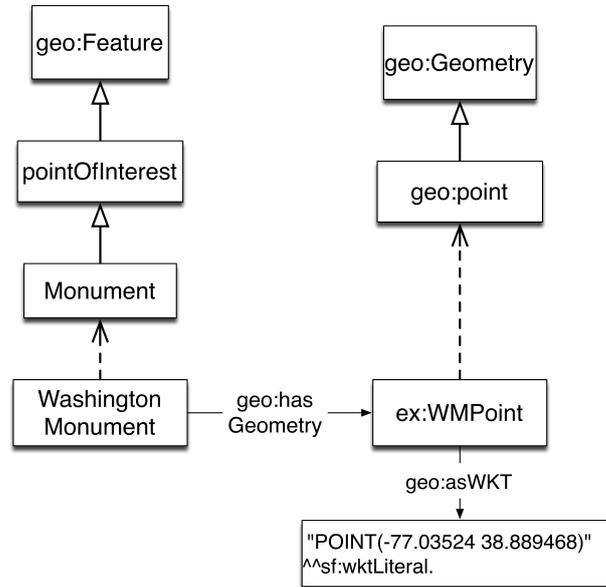
**GeoSPARQL** GeoSPARQL<sup>8</sup> est un standard défini par l'OGC (Open Geospatial Consortium). GeoSPARQL désigne à la fois une ontologie et un langage de requête, conçu comme une extension au langage SPARQL. Les deux classes principales de l'ontologie Geosparql sont représentées sur la figure 2.7. Il s'agit des classes *Feature* et *Geometry*, qui héritent toutes deux de la classe *SpatialObject*.



**Figure 2.7** – La classe spatiale *SpatialObject*, racine de l'ontologie GeoSPARQL. Source : <http://www.opengeospatial.org/blog/1673>

Selon les spécifications de GeoSPARQL, la classe *Feature* est la superclasse de toutes les classes caractérisant des objets pouvant avoir une caractéristique spatiale (par exemple, la classe *PointOfInterest* de la figure 2.8). Cette classe est équivalente à la classe *GFI Feature* de la norme ISO 19156 :2011. La classe *Geometry* est la superclasse de toutes les géométries parmi lesquelles des points, des courbes ou des surfaces. Elle est équivalente à la classe *GM Object* définie dans la norme ISO 19107. L'existence de différentes géométries spatiales peut permettre de modéliser des objets spatiaux à différents niveaux de granularité. La propriété *hasGeometry* relie une *Feature* à sa géométrie (classe *Geometry*).

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



**Figure 2.8** – Exemple d’utilisation de Geosparql pour décrire un lieu et sa géométrie.

La figure 2.8<sup>9</sup> présente un exemple d’utilisation de ces deux classes pour décrire un point d’intérêt (un monument) et sa géométrie. *Washington Monument* est une instance de la classe *Monument* qui est liée par la propriété *hasGeometry* à une instance de la classe *Point* qui est un des types de géométrie pouvant être caractérisés grâce à GeoSPARQL. Ce point a une propriété de donnée (*data property*) *asWKT* qui caractérise ses coordonnées géographiques sous la forme d’un littéral au format WKT, format de représentation des objets géométriques vectoriels défini par l’OGC<sup>10</sup>.

L’ontologie dispose également de propriétés d’objet (*object property*) permettant de caractériser des relations entre les *SpatialObject* : égalité, disjonction, intersection, adjacence, chevauchement, traversée et inclusion.

**OWL-Time** L’ontologie OWL-Time<sup>11</sup> [Hobbs and Pan, 2006a] permet de décrire des concepts temporels. L’ontologie utilise la norme ISO 8601<sup>12</sup>, qui permet la représentation des informations temporelles dans les systèmes informatiques. Elle permet la caractérisation d’instant et d’intervalles [Claramunt et al., 1997a] en utilisant comme système de référence le calendrier grégorien

9. <http://www.opengeospatial.org/blog/1673>

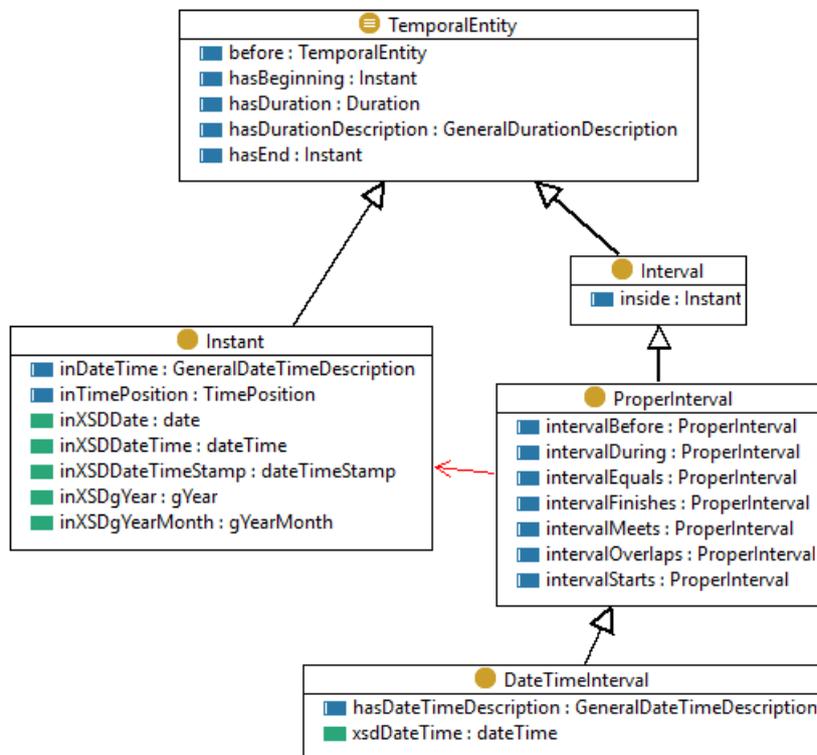
10. <http://www.opengeospatial.org/>

11. <https://www.w3.org/TR/owl-time/>

12. <https://www.iso.org/standard/40874.html>

pour la date, et le système horaire de 24 heures pour l'heure. Cette norme supporte ainsi six niveaux de granularité dans la représentation de l'information temporelle : année, mois, jour, heure, minutes et secondes. La représentation de la sémantique associée à l'information temporelle n'est pas prise en charge par cette norme.

La figure 2.9 détaille ce modèle. Les entités *Instant* et *Interval*, qui héritent de la classe *TemporalEntity*, servent à caractériser les instants et les intervalles conformément à la norme précitée. L'entité *ProperInterval* désigne un intervalle dont les instants qui en marquent le début et la fin sont différents. Cette notion est disjointe de celle d'instant.



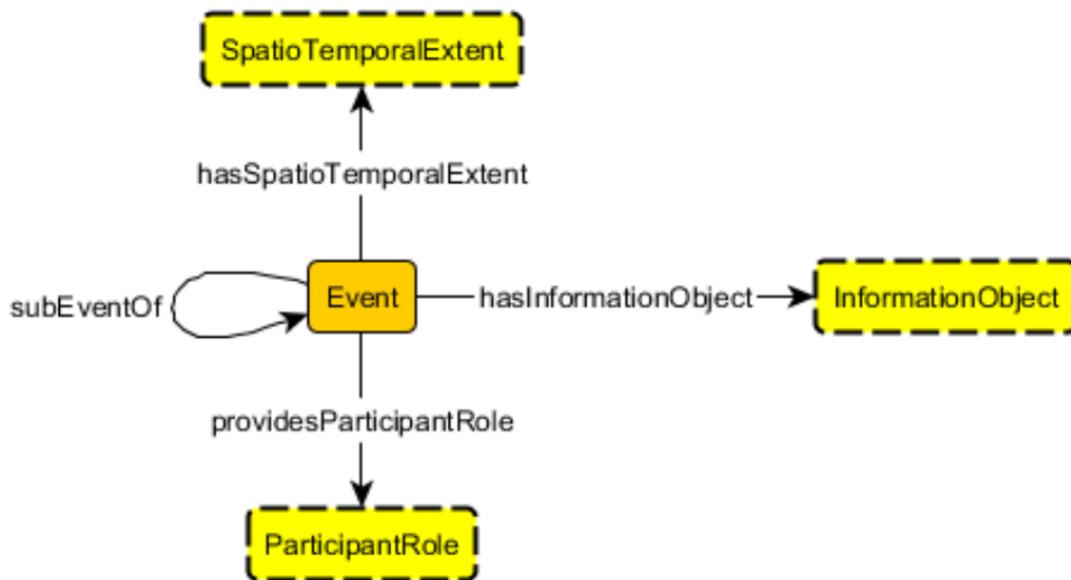
**Figure 2.9** – Le modèle d'entité temporelle utilisé dans OWL-Time [Hobbs and Pan, 2006a]

L'ontologie possède des propriétés qui reprennent l'algèbre d'Allen [Allen, 1984] et permettent de décrire les relations entre les intervalles temporels.

**Ontologies de description d'événements** Nous présentons des ontologies permettant de caractériser des événements. Nous nous intéressons particulière-

ment à la façon dont sont caractérisés les facteurs qui influencent les événements qui surviennent.

Krisnadhi and Hitzler [2017] proposent un *design pattern* d'ontologie, *Event Core*, minimaliste et permettant la représentation de tout événement. Un événement est situé dans le temps et dans un lieu, implique des participants, et est associé avec d'autres informations "non essentielles", telles que des noms ou des descriptions textuelles. Les illustrations données sont principalement des événements culturels ou sportifs. Ici, la signification d'*événement* ne correspond pas exactement à ce que nous avons besoin de représenter dans le cadre des trajectoires de vie, qui tient davantage d'une transition entre deux états d'une entité (un individu).

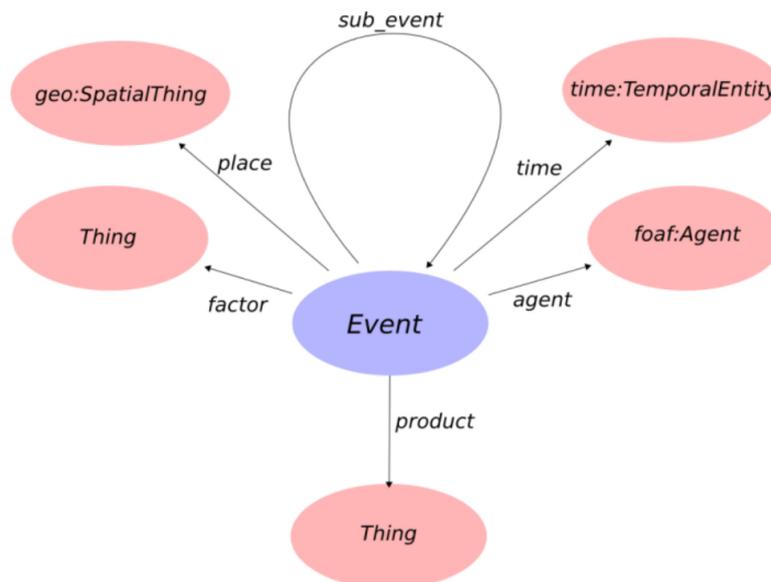


**Figure 2.10** – L'ontologie *Event Core* [Krisnadhi and Hitzler, 2017]

L'ontologie *Event Ontology*<sup>13</sup> (EO) est une ontologie de haut niveau utilisée pour décrire des événements à un niveau générique, de façon similaire à Krisnadhi and Hitzler [2017]. Un point intéressant est qu'elle inclut une propriété *facteur* qui "relie un événement à un facteur passif" (dans cette ontologie, un élément est considéré passif dans le cas où il n'est pas un agent qui prend part à l'événement). Inversement, une propriété appelée *produit* relie un événement avec ce qui peut être considéré comme une conséquence de cet événement. Ces propriétés, qui sont utilisées pour représenter les causes et les

13. <http://motools.sourceforge.net/event/event.html>

effets, sont de très haut niveau : elles n'ont pas de *range*, ce qui signifie qu'elles peuvent relier un événement à une instance de n'importe quelle classe en tant que facteur. Cette approche peut être utile pour modéliser les causes et les conséquences d'événement de vie, mais des propriétés plus précises doivent être définies pour permettre une caractérisation fine des influences et des facteurs explicatifs qui parsèment une trajectoire de vie.



**Figure 2.11** – Ontologie d'événements [Raimond and Abdallah, 2007]

L'ontologie LODE permet également de représenter des événements dans le Web de données [Troncy et al., 2010]. Cette ontologie utilise l'ontologie OWL-Time pour décrire la dimension temporelle de l'événement. Elle définit une propriété *involved* permettant de caractériser la participation d'une chose arbitraire à un événement. Elle ne contient pas de propriété permettant de représenter les facteurs qui influencent les événements.

L'ontologie BIO<sup>14</sup> permet de décrire des informations biographiques sur les individus. Elle donne la possibilité de caractériser un nombre limité d'événements de vie (par exemple *Birth*, *Mariage*, *Death*), mais n'est pas un modèle d'événement générique. Cette ontologie ne présente pas non plus de propriété susceptible de caractériser les facteurs qui influencent les événements.

### 4.3 Ontologies de trajectoires sémantiques

Baglioni et al. [2008] introduisent un *design pattern* d'ontologie (voir figure

14. [www.vocab.org/bio](http://www.vocab.org/bio)

2.12) suivant le modèle précédemment évoqué de Spaccapietra et al. [2008]. L'ontologie présentée en reprend les principaux concepts (les notions de *stop* et de *move*), en adaptant les relations qui les associent. Ce modèle, qui est l'un des premiers à mettre à profit le potentiel du Web sémantique pour la modélisation de trajectoire, permet l'utilisation d'ontologies de domaine, ainsi que des capacités de raisonnement du langage OWL dans un contexte d'enrichissement sémantique des trajectoires. Sur le plan conceptuel, il a les mêmes limites que celui de [Spaccapietra et al., 2008] en ce qui concerne la modélisation de trajectoires de vie : impossibilité de modéliser des trajectoires métaphoriques agéographiques et des transitions dans le cas de trajectoires discrètes.

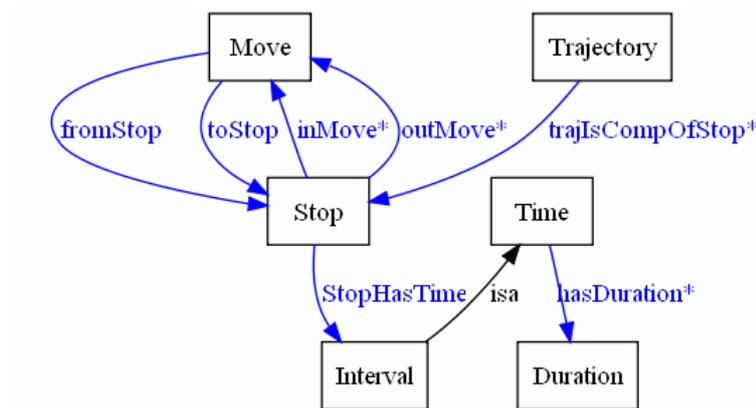
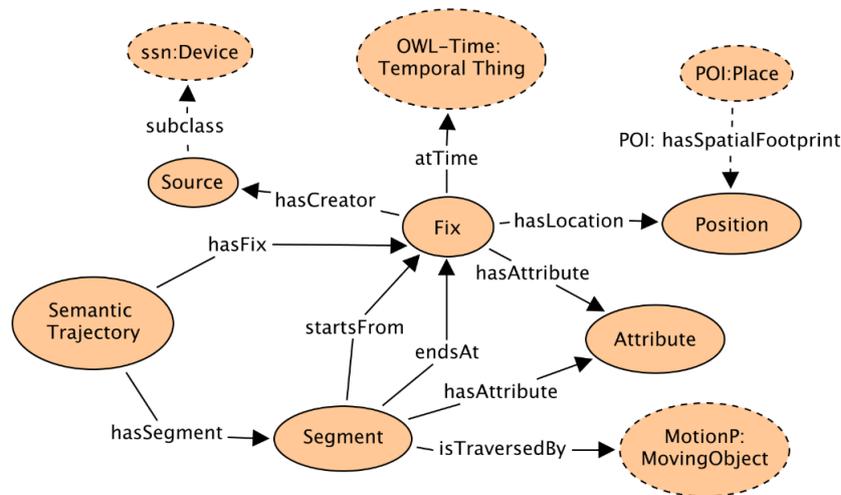


Figure 2.12 – Le *design pattern* proposé par Baglioni et al. [2008]

D'autres patrons d'ontologies, qui reposent sur des structurations différentes de trajectoires spatio-temporelles ont été proposés. Par exemple, Hu et al. [2013] proposent un patron de conception (*geo-ontology design pattern*) qui permet de caractériser une trajectoire sémantique comme une séquence de *segments* spatio-temporels. Ce modèle utilise des ontologies de référence de haut niveau telles que OWL-Time [Hobbs and Pan, 2006b]. Dans ce patron, le concept de *segment* est utilisé pour caractériser une portion de trajectoire qui regroupe un ensemble de positions (*fixes*). Une position fixe est ici donnée comme un point spatio-temporel (des coordonnées géographiques) indiquant la position d'un objet à un instant dans le temps.



**Figure 2.13** – Le design pattern d’ontologie proposé par [Hu et al., 2013]

Dans ce modèle, en comparaison avec celui proposé par [Spaccapietra et al., 2008] un *segment* recoupe à la fois la signification d’un *stop* et d’un *move*. En effet, au niveau logique, une position spatio-temporelle n’est pas associée à un *segment*, mais aux positions fixes (*fixes*).

Krisnadhi et al. [2016] proposent une généralisation de ce *pattern*, en utilisant la notion de *place*, à la place des coordonnées géographiques qui sont associées à un *fix*. Ces deux patrons de conception ont une forte composante géographique et ne sont pas adaptés à la modélisation de trajectoires métaphoriques agéographiques telles qu’on les rencontre dans les trajectoires de vie.

#### 4.4 Conclusion

Dans cette section nous nous sommes intéressés au Web sémantique et aux ontologies de trajectoires sémantiques. Nous avons évoqué le potentiel de réutilisation d’ontologies existantes, particulièrement de celles décrivant des informations spatiales ou temporelles. Le potentiel du LOD Cloud pour le croisement de données a également été abordé.

Sur le plan conceptuel, ces modèles ne permettent pas de modéliser des trajectoires agéographiques et des transitions dans le cas de trajectoires discrètes. L’utilisation d’ontologie a cependant l’avantage de permettre de modéliser une information thématique à différents niveaux de précision en utilisant des relations adaptées. De plus, l’utilisation de patrons de conception est particulièrement utile dans le cas des ontologies, les concepts proposés dans le *design pattern* pouvant alors servir de points d’extension pour l’utilisation

d'ontologies déjà existantes.

Certaines ontologies d'événements pourraient être une source d'inspiration pour caractériser ces transitions. L'ontologie EO <sup>15</sup> permet également de caractériser des facteurs explicatifs d'événements, mais des propriétés plus fines doivent être définies pour une utilisation dans le contexte des trajectoires de vie.

## 5 Conclusion

Dans ce chapitre nous nous sommes intéressés aux origines des modèles de trajectoires sémantiques, puis aux modèles conceptuels eux-mêmes et enfin aux ontologies. Sur le plan conceptuel, les travaux présentés dans cette partie traitent principalement de trajectoires ayant une dimension spatiale forte, ancrées dans l'espace géographique, pour lesquelles peuvent être représentées à la fois des périodes de stabilité et des périodes de mouvement. La présentation de l'origine de ces modèles, et notamment de la *time-geography* a permis de mettre en évidence les manques des bases conceptuelles de ces modèles pour caractériser les trajectoires de vie.

Il est également important de noter que les modèles de trajectoires sémantiques présentés ici proposent une caractérisation générique des trajectoires. Ces approches sont donc importantes dans la perspective d'un modèle de trajectoires de vie de haut niveau et pouvant caractériser différentes thématiques. Le patron de conception, qui a vocation à être appliqué pour répondre à un problème générique, et peut être étendu, est particulièrement adapté pour proposer un modèle de trajectoire de vie.

Dans l'objectif de caractériser différents points de vue thématiques au sein d'un même modèle d'autres approches de modélisation intégrant de multiples points de vue sont abordés dans le chapitre suivant, afin d'examiner leurs intérêts pour la modélisation de trajectoires de vie.

---

15. <http://motools.sourceforge.net/event/event.html>

# Multiples points de vue et trajectoires

---

## Sommaire

<b>1</b>	<b>Introduction</b> . . . . .	<b>43</b>
<b>2</b>	<b>Information multi-point de vue</b> . . . . .	<b>44</b>
2.1	La notion de point de vue . . . . .	44
2.2	La multi-représentation . . . . .	45
2.3	Conclusion . . . . .	45
<b>3</b>	<b>Information multi-granulaire</b> . . . . .	<b>46</b>
3.1	La granularité . . . . .	46
3.2	Approches utilisées pour la modélisation multi-granulaire	47
3.3	Conclusion . . . . .	49
<b>4</b>	<b>Modéliser des trajectoires multi-points de vue et multi-granulaires</b>	<b>49</b>
4.1	Introduction . . . . .	49
4.2	Trajectoires sémantiques multi-points de vue et multi-granulaires . . . . .	50
4.3	Trajectoires de vie de Thériault . . . . .	52
4.4	Conclusion . . . . .	55
<b>5</b>	<b>Conclusion</b> . . . . .	<b>55</b>

---

## 1 Introduction

Les trajectoires de vie sont par nature multi-dimensionnelles : temporelles, mais aussi spatiales et thématiques. De plus, chaque trajectoire thématique de la trajectoire de vie (trajectoire résidentielle, professionnelle...) peut également être considérée comme une dimension de la trajectoire de vie. Dans le cas des trajectoires sémantiques, considérer une seule représentation thématique pour une trajectoire peut également s'avérer insuffisant [Ferrero et al., 2016].

Cette question renvoie en ingénierie de la connaissance à la notion de *point de vue* que présentons dans la section 2. La question du point de vue renvoie également à la granularité de l'information, là aussi temporelle, spatiale et thématique. Comme nous l'avons vu dans la partie introductive, la possibilité de stocker une information multigranulaire est essentielle dans le cas des trajectoires de vie.

Dans cette partie, nous nous intéressons d'abord aux travaux sur la modélisation d'information multi-points de vue, puis aux approches consacrées à une information multi-granulaire, et enfin nous passons en revue les modèles de trajectoires sémantiques qui prennent en compte ces notions.

## 2 Information multi-point de vue

Nous présentons dans un premier temps la notion de point de vue, puis une approche de modélisation de cette notion très utilisée pour la modélisation spatio-temporelle : la *multi-représentation*.

### 2.1 La notion de point de vue

Un point de vue [Ullman, 1988] est une perception parmi d'autres qu'un observateur a d'une entité étudiée. Pour Mariño [1993], la notion de point de vue met "en correspondance un agent et un monde". Dans notre cas, l'*agent* est l'expert qui cherche à étudier des trajectoires de vie, et le *monde* correspond à la trajectoire de vie de l'individu. L'expert retient les points de vue qui l'intéresse pour son cas d'application (par exemple résidentiel, professionnel...). Conformément à la conception de point de vue la plus répandue (correspondant à la notion de *vue* dans les systèmes de gestion de bases de données), nous considérons que l'union des points de vue forme une vision complète et cohérente de l'entité.

Pour [Euzenat, 1999], l'utilisation de cette notion peut permettre de simplifier un phénomène tout en préservant sa complexité en faisant ressortir les traits importants pour une situation d'observation particulière de l'entité. Nous allons maintenant voir comment les différents points de vue d'une entité du monde réel peuvent être modélisés. La multi-représentation est une approche de représentation de la connaissance qui le permet.

## 2.2 La multi-représentation

La multi-représentation est utilisée pour représenter les différents points de vue qu'un observateur a d'une entité étudiée. Chaque représentation est utilisée pour caractériser une perception, il y a donc un *mapping* bijectif entre perceptions et représentations, c'est à dire que à une perception correspond une et une seule représentation et réciproquement. Une représentation est donc la partie d'un modèle qui correspond à un point de vue particulier.

Le langage MADS [Parent et al., 2006a] supporte la multi-représentation [Vangenot et al., 2002]. Ce langage intègre à des modèles entité-relation des pictogrammes qui permettent de représenter l'information spatio-temporelle. Afin d'organiser les différentes représentations les auteurs utilisent des principes simples [Parent et al., 2006b]. Pour chaque représentation d'une même information, ils proposent l'utilisation d'un *stamp* (tampon) spécifique. Ainsi, un *stamp* correspondant à chaque point de vue est défini, et celui-ci est utilisé pour *tamponner* les différents composants du modèle entité-relation qui sont valides pour ce point de vue. Cette approche permet de caractériser de multiples points de vue dès la phase de modélisation. Dans le modèle, un *stamp* traduit donc l'appartenance à un point de vue. L'expert d'un domaine souhaitant modéliser des trajectoires de vie, pourrait donc choisir lui même un *stamp* pertinent, qui permettrait alors de tamponner chaque trajectoire thématique et ses composants. Dans nos travaux, nous réutilisons cette approche en l'adaptant pour permettre la modélisation d'une ontologie de trajectoires de vie intégrant plusieurs points de vue.

D'autres approches existent pour la modélisation de multiples représentations comme Vuel [Bedard and Bernier, 2002]. Cette approche utilise la notion de vue telle qu'elle est présente dans les Systèmes de Gestion de Bases de Données. Des approches basées sur UML existent également, comme par exemple OMT-G [Borges et al., 2001], travail dans lequel les différentes représentations d'une entité sont des sous-classes d'une classe représentant cette entité. Les attributs communs aux différentes représentations sont donc définis dans la superclasse et chaque représentation en hérite. Parmi ces langages, le langage MADS est le seul qui permette la multi-représentation des associations et pas seulement des entités.

## 2.3 Conclusion

Nous avons présenté dans cette partie la notion de point de vue ainsi que la multi-représentation. La notion de point de vue est particulièrement adaptée pour la modélisation d'un objet (par exemple, d'une personne) pour lequel

un observateur (par exemple, un spécialiste de domaine) aurait différentes thématiques d'intérêt (résidentielle, familiale, professionnelle...). L'utilisation de la multi-représentation dans le langage MADS (langage basé sur le modèle entité-relation) a montré son utilité. L'approche de *stamping* est une approche simple permettant d'intégrer de multiples points de vue dans un modèle conceptuel et qui peut être adaptée pour la modélisation d'ontologies.

### 3 Information multi-granulaire

Dans cette partie, de la même façon que précédemment, nous présentons d'abord la notion de *granularité*, afin d'évaluer son intérêt pour la modélisation de trajectoire de vie. Puis, nous décrivons différentes approches pour la modélisation de l'information multi-granulaire. Nous montrons les avantages et limitations de ces approches pour la modélisation de trajectoires de vie.

#### 3.1 La granularité

Nous l'avons vu, le modèle doit supporter différents niveaux de granularité temporelle [Hornsby and Egenhofer, 2002], en raison de la capacité (ou non) des individus à situer précisément les événements de leur vie, mais aussi pour supporter des analyses ultérieures. Ce raisonnement est aussi valable pour l'information de nature thématique, particulièrement lors de la phase d'exploitation des trajectoires. Si, dans le cas de la collecte de l'information, l'importance de la granularité apparaît moins évidente, elle l'est largement dans le cas de l'exploitation. Considérons par exemple l'attribut *emploi* qui décrit une partie de la trajectoire professionnelle d'un individu. Cette information, peut-être considérée à un niveau de précision inférieur, par exemple la catégorie socioprofessionnelle. Dans le cas d'une collecte de données effectuée directement auprès de l'individu, il est peu probable qu'il soit plus facile pour lui de renseigner sa catégorie socioprofessionnelle que son emploi. Cependant, situer cet emploi sur un référentiel multi-granulaire, qui intègre les catégories socioprofessionnelles, permettra une meilleure exploitation des trajectoires de vie collectées. Cela permettra, par exemple, de connaître l'ensemble des trajectoires des individus appartenant à une catégorie socioprofessionnelle donnée, même si ceux-ci n'ont renseigné que leur emploi.

Un raisonnement similaire est également appliqué pour déterminer le niveau de granularité auquel il convient de considérer l'information de nature géographique, ce qui renvoie à la question du MAUP (Modifiable Area Unit Problem) de Openshaw and Taylor [1979]. Les frontières évoluant au cours du

temps, il convient de permettre la traçabilité des changements [Plumejeaud, 2011, Bernard et al., 2018] qui impactent les unités territoriales (quartier, ville, département, *etc.*), supports de l'information consignée (comme le lieu de résidence, par exemple). La problématique de la granularité étant fortement dépendante du domaine d'application, il revient au modèle d'offrir toute la flexibilité nécessaire pour le stockage de ces informations temporelles et spatiales à un niveau suffisant de détail permettant les analyses ultérieures.

Pour Vangenot et al. [2002], les différents niveaux de granularité correspondent aux différents niveaux de perception possibles d'une entité du monde réel. En ce sens les différents niveaux de granularité correspondent donc à différents points de vue. La multi-représentation peut donc également être utilisée pour modéliser différents niveaux de granularité. Ce n'est cependant pas la seule approche à pouvoir être utilisée dans ce but, comme nous allons le voir maintenant.

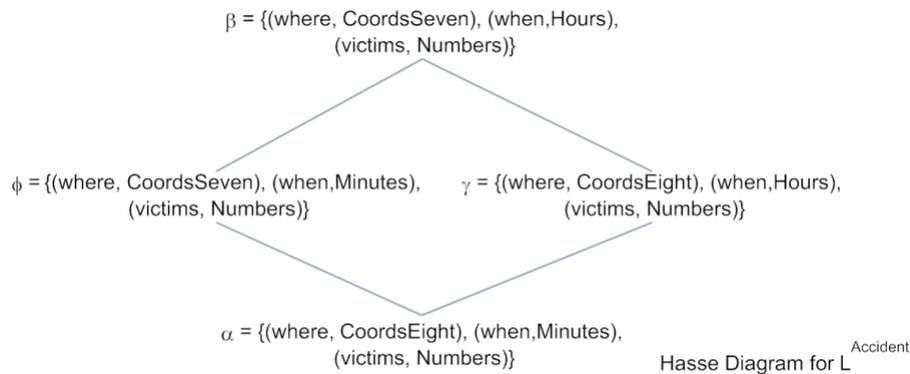
### 3.2 Approches utilisées pour la modélisation multi-granulaire

**La multi-résolution** La multi-résolution [Weibel and Dutton, 1999] est une approche centrée sur la caractérisation de la dimension spatiale à différents niveaux de granularité en cartographie. Selon la granularité, une même entité du monde réel est représentée différemment. Par exemple, les contours d'un bâtiment peuvent être représentés à un niveau fin de granularité tandis qu'à un niveau plus grossier, il n'est plus représenté que par un point. Finalement, son existence peut même être remise en question à un niveau de granularité où la représentation de ce bâtiment n'est plus pertinente.

**L'utilisation de la multi-représentation** La multi-représentation, présentée dans la partie précédente, peut être utilisée pour modéliser la granularité. Les niveaux de granularité sont alors considérés comme différents points de vue. Il s'agit d'associer un *stamp* (ou *tampon* en français) à la représentation d'une granularité et d'estampiller par ce *stamp* les entités qui correspondent à cette granularité. Cette méthode a cependant ces limites. En effet, un trop grand nombre de représentations peut faire perdre en clarté au modèle conceptuel, même si pour Parent et al. [2006a] un travail dans l'organisation de ces différentes représentations pourrait être mené. De plus, il n'existe pas de technique d'agrégation des données modélisées en vue d'analyses ultérieures.

**Les modèles basées sur la théorie de la granularité** Dans leurs travaux sur la granularité temporelle [Bettini et al., 1998] posent les bases d'une théorie de

la granularité. En se basant sur cette théorie, [Silva et al. \[2015\]](#) proposent un modèle permettant la définition de granularité dans n'importe quel domaine, et pas uniquement le temps. Dans ce modèle, une entité du monde réel est décrite par un prédicat comprenant plusieurs arguments. Pour chaque argument un ensemble de niveau de granularité est défini. Chaque granularité est une division d'un domaine (par exemple, heures, minutes, secondes). Chaque division est non décomposable et appelée *granule* (par exemple, une heure, une minute ou une seconde). Par exemple, le prédicat accident présenté sur la figure 3.1 a trois arguments : *where*, *when* et *victims* et son argument *when* a deux niveaux de granularité : heure et minute. Un *atome*, valable pour un prédicat défini et décrivant un phénomène du monde réel (un accident dans l'exemple précédent), peut être énoncé et chacun de ses arguments utilise alors une division non décomposable d'un de ses niveaux de granularité : une *granule*. Par exemple, un atome du prédicat accident peut être exprimé ainsi  $accident((where, coord1), (when, 01-01-2002\ 07h\ am), (victims, one))$ .



**Figure 3.1** – Les différents niveaux de détail d'un prédicat représentés sur un digramme de Hasse [[Silva et al., 2015](#)]

Les auteurs définissent également la notion de niveau granularité auquel est exprimé un prédicat (voir figure 3.1). Nous l'avons vu, chaque argument d'un prédicat est lui-même exprimé à un niveau de granularité. L'ensemble des niveaux de granularité auxquels sont exprimés chacun des arguments du prédicat est alors considéré comme le niveau de granularité du prédicat. Dès lors, des prédicats composés du même ensemble d'arguments (par exemple *when*, *where*, *victims*) peuvent être reliés par une relation *est plus détaillé que* (*finer than*), étant entendu qu'un prédicat est plus détaillé qu'un autre uniquement si chacun de ses arguments est d'un niveau de granularité plus fin ou égal et que au moins un est plus fin. Les différents niveaux de granularité d'un prédicat sont appelés LoDs (*levels of detail* en anglais).

Ce travail permet ensuite la définition de la notion de *généralisation* d'un

*atome*, qui permet d'augmenter le niveau de granularité auquel est exprimé un prédicat.

### **3.3 Conclusion**

Nous avons vu dans cette section que la granularité était importante pour les trajectoires de vie, à la fois pour la modélisation, la collecte de données et leur exploitation. Nous avons ensuite présenté différentes approches de modélisation de l'information multi-granulaire. Les limites de ces approches de modélisation impactent également les phases de collecte et d'exploitation. Le modèle basé sur la granularité proposé par [Silva et al. \[2015\]](#) est intéressant pour la modélisation de trajectoires de vie car il permet de définir des niveaux de granularité pour n'importe quel domaine et pas seulement l'espace et le temps. A ce titre, ces travaux constituent une base conceptuelle intéressante pour caractériser la granularité dans les trajectoires de vie.

## **4 Modéliser des trajectoires multi-points de vue et multi-granulaires**

### **4.1 Introduction**

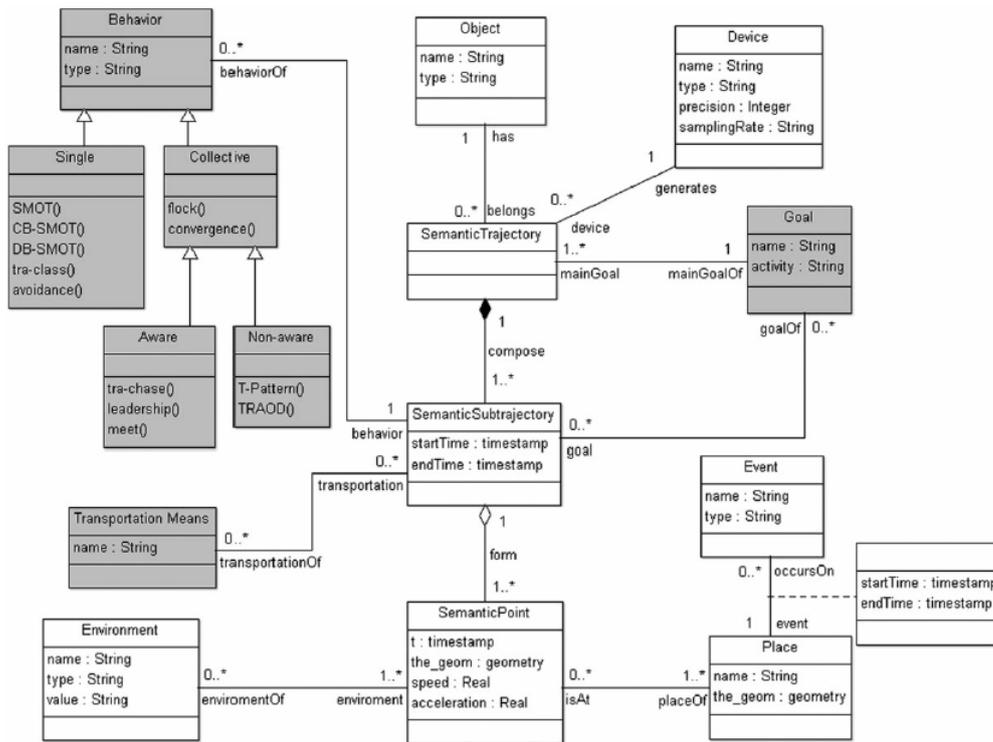
*Points de vue, aspects* [[Ferrero et al., 2016](#)], *sous-trajectoires* [[Bogorny et al., 2014](#)] sont autant de termes différents qui prennent en compte la multiplicité des thématiques d'intérêt dans la modélisation de trajectoire. Dans de récents travaux, [Ferrero et al. \[2016\]](#) pointent le challenge important que constitue l'étude des trajectoires avec une sémantique multidimensionnelle. Il ne s'agit plus seulement de s'intéresser aux dimensions spatiale, temporelle et sémantique, mais de considérer une sémantique multi-points de vue, dont chacun peut être vu comme une dimension de la trajectoire. L'enjeu est une caractérisation plus fine de l'évolution spatiale ou temporelle. En effet, pour comprendre la trajectoire d'un individu, il faut s'intéresser à son environnement géographique mais aussi à un contexte personnel, social... Des travaux commencent donc à prendre en compte une sémantique multi-thématique.

Dans cette section nous présentons les approches existantes pour la modélisation de ces trajectoires multi-points de vue. Nous étudions d'abord les modèles de trajectoires sémantiques caractérisant de multiples aspects, avant de nous intéresser au modèle de trajectoire de vie élaboré par [Thériault et al. \[2002\]](#).

## 4.2 Trajectoires sémantiques multi-points de vue et multi-granulaires

**Trajectoires intégrant de multiples aspects** De nombreuses approches pour la modélisation des trajectoires sémantiques considèrent trois dimensions : le temps, l'espace et la sémantique (voir section 3 du chapitre II).

Cependant, de récentes approches s'attachent à considérer également une sémantique multidimensionnelle [Ferrero et al., 2016].



**Figure 3.2** – Le modèle CONSTANT [Bogorny et al., 2014] prend en compte de multiples aspects pour une trajectoire

Bogorny et al. [2014] proposent un modèle supportant la représentation des différents points de vue d'une trajectoire, qui sont considérés comme un ensemble *d'aspects*.

Dans ce modèle, caractérisé grâce au langage UML et présenté sur la figure 3.2, la trajectoire sémantique (classe *SemanticTrajectory*) appartient (relation *belongs*) à un objet (classe *Object*) qui caractérisent l'objet mobile au moyen de deux attributs de type chaîne de caractères, *name* et *type*. Chaque aspect de la trajectoire est caractérisé par la classe *SemanticSubtrajectory* qui a comme attributs deux *timestamp*, *startTime* et *endTime*. La classe *SemanticTrajectory* est une *composition* de une ou plusieurs *SemanticSubTrajectory*.

Chaque sous-trajectoire a un but (classe *Goal*) caractérisé par deux attributs de type chaîne de caractères, *name* et *activity*. La trajectoire est reliée à son but principal par la relation *mainGoal*. L'un des buts des sous-trajectoires est donc le but de la trajectoire principale. Chaque sous-trajectoire est elle-même une composition de *SemanticPoint*, classe qui représente des points GPS (les données brutes de la trajectoire). Chacun de ces points peut-être associé à un ou plusieurs lieux (classe *Place*) sur lesquels peuvent survenir des événements (classe *Event*). Le reste des classes est consacré à des méthodes permettant la détection de *pattern* dans les trajectoires. Ce modèle apporte un éclairage intéressant concernant la modélisation de trajectoires multipoints de vue en utilisant la notion de *composition*, ici avec le langage UML, mais reste centré sur la modélisation de trajectoires d'objets mobiles et n'est pas adapté à la modélisation de trajectoires agéographiques car toute sous-trajectoire est une composition de géométrie, et les concepts utilisés sont ceux du mouvement dans l'espace géographique (vitesse, accélération...). Il n'aborde pas non plus la question des trajectoires multigranulaires.

**Trajectoires multigranulaires** Fileto et al. [2015] introduit un *framework* conceptuel pour l'enrichissement de trajectoires géographiques basé sur le Web sémantique. Ce *framework* repose sur l'ontologie Baquara représentée sur la figure 3.3.

Dans ce modèle, le mouvement est caractérisé par des segments (*MovementSegment*). Il s'agit de trajectoires ou bien de séquences ordonnées temporellement d'épisodes (*Episode*) (de type *Stop* ou *Move*).

L'ontologie inclut également la caractérisation d'événements externes (classe *Event*) à la trajectoire dans son module consacré à l'analyse du mouvement (*MovementAnalysisFacet*). Ces événements reliés à un lieu (*Place*) et une information temporelle (*Time*) peuvent ensuite être mis en relation avec les trajectoires dans un but d'analyse.

La particularité principale de ce modèle est d'introduire la notion de *granularité* entre les différents épisodes composant une trajectoire par l'introduction d'une relation hiérarchique entre les différents segments.

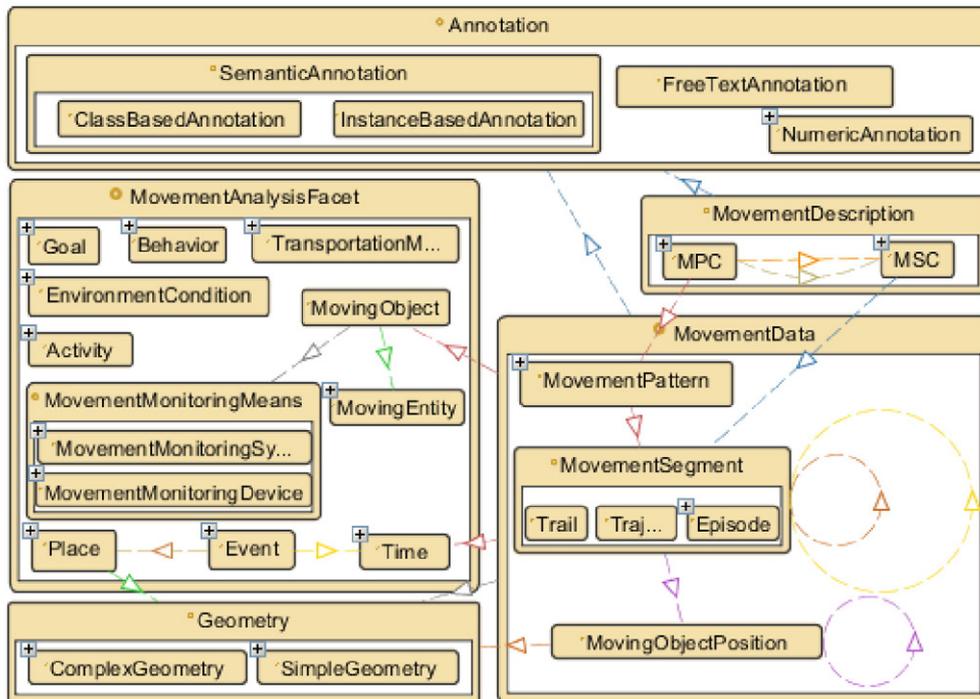


Figure 3.3 – Représentation de l'ontologie Baquara [Fileto et al., 2015]

### 4.3 Trajectoires de vie de Thériault

**Un modèle basé sur la dimension temporelle** La dimension temporelle est essentielle dans l'étude des trajectoires de vie. En effet, même si la dimension spatiale ne doit pas être négligée pour les perspectives qu'elle ouvre en termes d'analyse des données, la seule dimension commune entre toutes les trajectoires de vie est la dimension temporelle. Dans des travaux précurseurs Thériault et al. [2002] proposent des modèles de données pour décrire des trajectoires de vie complexes, dans lesquels sont représentés également les événements qui caractérisent un changement dans la (trajectoire de) vie d'une personne. Ces recherches font suite à une série de travaux [Claramunt and Thériault, 1995, Claramunt et al., 1997b, 1998b] consacrés à la modélisation d'événements et à la modélisation d'informations temporelles qui eux mêmes tirent leurs fondements du *framework* conceptuel proposé par Peuquet [Peuquet, 1994, Peuquet and Duan, 1995] pour modéliser le temps, l'espace et la thématique.

Deux approches sont utilisées pour caractériser le temps comme le rappellent Plumejeaud et al. [2011] : l'approche absolue, celle de Newton, et l'approche relative, introduite par Leibniz. Ainsi, l'approche absolue considère le temps comme un flux mesurable, une seule dimension pouvant être représenté par

une courbe alors que l'approche relative se concentre sur des entités du monde réel et leurs relations [Claramunt et al., 1998a], ce qui conduit à considérer le temps comme une succession d'événements. Pour Claramunt et al. [1998a], il est possible de combiner les deux approches en considérant l'évolution du temps comme une succession d'événements, sans pour autant renoncer à le mesurer, grâce aux unités dont nous disposons (année, jour, heure. . .). Évaluer une durée présuppose un *référentiel* et la définition d' *unités de mesure*.

**Les fondements conceptuels du modèle** La modélisation du temps est souvent associée à celle de l'espace géographique. Dans ses travaux, Peuquet [1994], propose un framework conceptuel permettant d'associer le temps (Quand?), l'espace géographique (Où?) et la thématique (Quoi?). Ce système de référence est connu sous le nom de *triade de Peuquet*. Il permet la modélisation d'observations spatio-temporelles en y associant de la sémantique (la thématique). Claramunt et al. [1998a], pointant les limites de cette triade, qui ne permet pas la représentation d'événements et de processus, ni l'expression de causalités, présentent un *framework* caractérisant trois niveaux de connaissances scientifiques : description, expérimentation et explication. Le niveau de description est celui de l'observation des *faits* et reprend la triade de Peuquet (Quand, Où, Quoi). Le second niveau, celui de l'expérimentation, permet la caractérisation d'événements et la compréhension des processus à l'œuvre et donc les relations entre ces événements. Enfin, le troisième niveau est celui de l'expérimentation : celui de la compréhension des causes. Il s'agit de comprendre l'évolution des faits précédemment décrits, ce qui pourrait permettre idéalement de les prévoir. Ces travaux conceptuels déboucheront un peu plus tard sur les premiers modèles de trajectoires de vie.

**Le modèle et ses entités** Les travaux conduits par Marius Thériault font référence dans le domaine de la modélisation de trajectoires de vie.

Le modèle spatio-temporel pour l'analyse des trajectoires de vie [Thériault et al., 1999] repose sur trois trajectoires différentes – résidentielle, familiale et professionnelle. Chacune de ces trajectoires est modélisée conceptuellement par des *épisodes* – des *statuts* stables pendant un intervalle de temps –, et des événements qui viennent altérer un ou plusieurs de ces statuts. Un statut correspond à la valeur d'un attribut. Par exemple, sur la figure 3.4, la valeur de l'attribut *Occupation*, correspond au statut *student* (étudiant).

Le modèle est présenté selon une approche relationnelle, qui pourrait également être adaptée à une approche orientée objet. Ce modèle, alimenté par des données de recensement officiel et implanté dans une base de données,

permet d'effectuer des requêtes spatiales, temporelles et thématiques, dont les résultats peuvent ensuite être exploités pour des analyses statistiques.

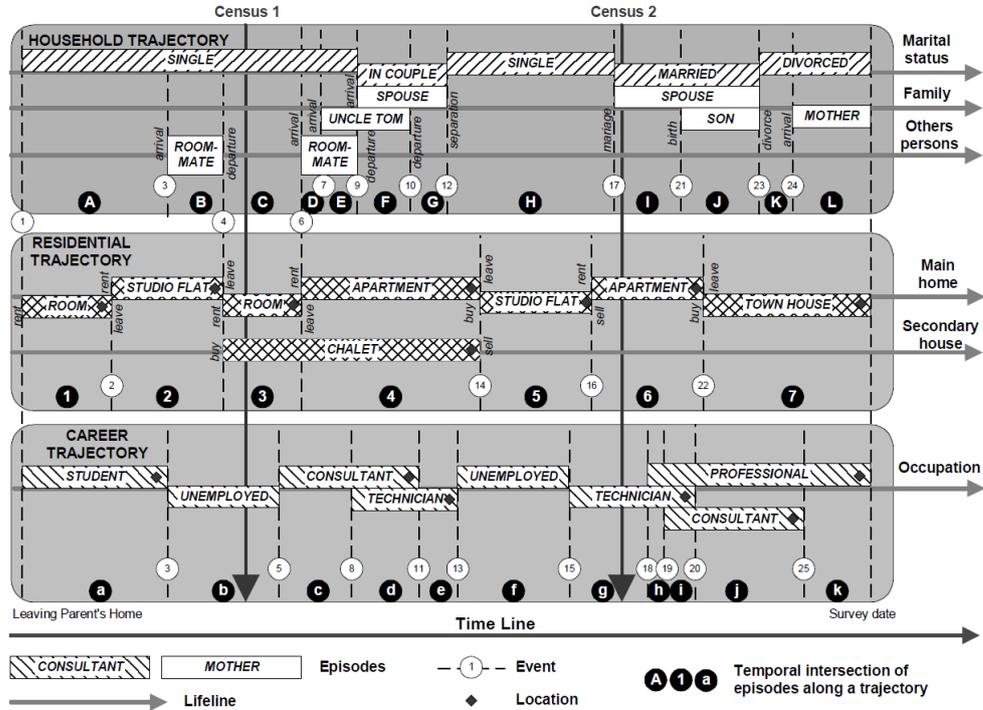


Figure 3.4 – La trajectoire de vie de Thériault et al. [2002]

Plus tard, le modèle sera modifié de façon à permettre de déterminer la probabilité qu'un événement survienne dans certaines conditions dans une trajectoire de vie en utilisant la régression de Cox, modèle statistique permettant la manipulation d'événements et adapté aux données tronquées [Thériault et al., 2002].

Les conditions du choix résidentiel sont également abordées par Vandersmissen et al. [2009] avec une emphase particulière sur le rôle joué par les changements d'emplois qui interviennent dans la vie professionnelle des individus.

Les modèles qui sont proposés dans ces travaux sont particulièrement intéressants dans l'optique d'une modélisation de trajectoires agéographiques. Les concepts proposés, notamment les *épisodes* et les *événements*, permettent d'envisager de caractériser des trajectoires indépendamment de la dimension spatiale. Cependant, un modèle plus générique doit être défini pour caractériser des trajectoires issues de cas d'applications différents. On peut également noter que ces modèles sont centrés sur l'aspect temporel des choix résidentiels : on s'intéresse davantage aux raisons pour lesquelles les individus vont déménager à un moment donné (en fonction des circonstances de leur vie) qu'aux raisons pour lesquelles ils choisissent le lieu où ils vont emménager. De

plus l'évolution des technologies numériques, et l'émergence de nouveaux paradigmes tel que le Web sémantique permettent d'envisager une modélisation caractérisant davantage d'informations sans augmenter la complexité pour le modélisateur, en réutilisant des ontologies, temporelles et spatiales, mais aussi des ontologies de domaines liées au cas d'application. Le Web sémantique ouvre donc des perspectives en termes de modélisation des thématiques retenues pour les trajectoires. Grâce aux ontologies utilisées, des croisements de données à des fins d'analyse pourront également être envisagés.

#### 4.4 Conclusion

Dans ce chapitre, nous avons passé en revue des travaux portant sur la modélisation des multiples points de vue d'une trajectoire. Ceux-ci ne permettent pas de modéliser des trajectoires métaphoriques géographiques. L'approche Baquara [Fileto et al., 2015] intègre la notion de granularité et permet sa modélisation grâce à l'expression d'un lien entre épisode caractérisant une relation hiérarchique.

Le modèle de trajectoire de vie de Thériault et al. [2002] est le seul modèle permettant de caractériser à la fois des trajectoires agéographiques et des trajectoires géographiques. Il permet également de caractériser les événements de vie qui constituent une transition entre des périodes de stabilité, même s'il n'intègre pas de solution conceptuelle pour représenter des éléments d'explications permettant de mieux comprendre les trajectoires. Sur le plan conceptuel, il constitue donc une base solide de modélisation de trajectoires de vie. Cependant, les concepts définis ici pour le cas d'application de l'étude des trajectoires résidentielles doivent être redéfinis à un niveau plus générique. Cette redéfinition pourrait également permettre une intégration dans le Web des données. De plus, il n'aborde pas la question de la modélisation à différents niveaux de granularité.

## 5 Conclusion

De nombreux travaux sur la modélisation d'informations multidimensionnelles existent. Certaines approches permettant de modéliser la notion de point de vue, comme la multi-représentation, pourraient être réutilisées dans le cadre de la conception d'un modèle générique de trajectoires de vie. De même, certaines approches de modélisation et de représentation de l'information multi-granulaire pourraient permettre de caractériser le niveau de granularité des trajectoires de vie afin de permettre une collecte et une

exploitation multi-granulaire de trajectoires.

Dans le chapitre suivant, nous résumons les limites des modèles conceptuels existant, ainsi que le chemin restant à parcourir vers une approche de modélisation, de collecte et d'analyse de trajectoires de vie utilisable par un spécialiste de domaine.

# Synthèse et positionnement

---

## Sommaire

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>57</b>
<b>2</b>	<b>Modéliser des trajectoires de vie</b> . . . . .	<b>57</b>
2.1	Limites des approches de modélisation et positionnement	58
2.2	Analyse comparative des modèles conceptuels . . . . .	60
2.3	Conclusion . . . . .	63
<b>3</b>	<b>Approches de modélisation, d’acquisition et d’exploitation des données</b> . . . . .	<b>63</b>
3.1	Introduction . . . . .	63
3.2	Limites des approches présentés dans l’état de l’art . .	64
3.3	Vue générale de notre approche . . . . .	65
3.4	Conclusion . . . . .	66
<b>4</b>	<b>Conclusion</b> . . . . .	<b>66</b>

---

## 1 Introduction

Dans ce chapitre, nous présentons une vision globale des limites de l’état de l’art et des défis à relever pour la modélisation des trajectoires de vie.

Nous nous concentrons d’abord sur la modélisation, puis sur les approches globales proposées par les auteurs cités dans l’état de l’art, c’est-à-dire la démarche proposée qui intègre la modélisation, mais aussi l’acquisition de données et leur exploitation.

## 2 Modéliser des trajectoires de vie

Nous distinguons la capacité de caractérisation, qui désigne ce que un modèle permet de caractériser, et l’approche de modélisation elle-même, qui désigne la phase lors de laquelle le modèle est conçu et éventuellement adapté pour

un cas d'application donné, ainsi que les moyens mis en œuvre pendant cette phase.

## 2.1 Limites des approches de modélisation et positionnement

Nous comparons ici les approches de modélisation présentées dans l'état de l'art. Le tableau 2.1 résume les approches selon les critères suivants : les concepts de structuration utilisés, le méta-modèle, le fait qu'elles utilisent un *design pattern* et enfin la facilité pour passer au modèle et à son instanciation pour un cas d'application particulier.

Le concept de structuration permet d'évaluer la généralité du modèle, en fonction de ce que lui-même permet de caractériser. Concernant le méta-modèle, nous nous intéressons particulièrement aux ontologies, dont nous avons vu les avantages qu'offre leur intégration dans le Web sémantique. De même, nous nous intéressons particulièrement aux approches utilisant un *design pattern*. Quant à la facilité d'utilisation du modèle proposé, dans notre cas elle doit être suffisante pour permettre à un spécialiste de domaine de prendre en charge la modélisation. Nous revenons sur chacun de ces points à la suite du tableau.

	Concepts de structuration	Méta modèle	Design pattern	Facilité d'utilisation
Spaccapietra et al. [2008]	<i>stops and moves</i>	MADS <sup>1</sup>	oui	non
Baglioni et al. [2008]	<i>stops and moves</i>	Ontologie	oui	non
Hu et al. [2013]	<i>segment</i>	Ontologie	oui	non
Bogorny et al. [2014]	<i>sous-trajectoires</i>	UML	non	non
Andrienko et al. [2011]	<i>event</i>	Conceptuel	non	non
Thériault et al. [2002]	<i>episode</i>	Entité relation	non	non

**Table 4.1** – Comparatif des techniques de modélisation

**Concept de structuration** Spaccapietra et al. [2008] et par la suite Baglioni

et al. [2008] utilisent le concept de *stop* et de *move*. Toutes ces approches ont montré leur utilité pour la modélisation de trajectoires sémantiques spatio-temporelles, mais ne permettent pas de modéliser des trajectoires agéographiques.

Spaccapietra et al. [2008] invitent à redéfinir ces notions dans l'espace abstrait. Un concept générique, permettant à la fois la modélisation d'épisodes de trajectoires géographiques et de trajectoires agéographiques, est cependant requis. La notion de segment proposé par Hu et al. [2013] a également une connotation géographique très forte.

La notion de *sous-trajectoire* proposée par [Bogorny et al., 2014] permet la modélisation de points de vue multiples, mais la sémantique se limite au *but* de la trajectoire, qui est caractérisé par un seul attribut (*activity*).

Les notions d'*épisodes* et d'*événements* proposée par Thériault offrent l'avantage de permettre de modéliser des trajectoires métaphoriques, géographiques ou agéographiques. Dans le cas d'une trajectoire géographique, la composante spatiale est un attribut de l'épisode d'une trajectoire qui, en son absence, serait agéographique. À un certain niveau d'abstraction, le concept d'épisode (sans attribut spatial) permet donc bien de caractériser de manière générique trajectoire métaphorique géographique et trajectoire métaphorique agéographique.

Cependant, l'espace *abstrait* de la trajectoire métaphorique, tout comme l'espace géographique, doivent pouvoir être caractérisés de manière plus expressive que selon un modèle attributaire (comme le permet ce modèle entité-relation), par exemple sous la forme d'une ontologie, ce qui permet d'intégrer des liens, notamment hiérarchique en vue de caractériser des niveaux de granularité différents, entre les concepts caractérisés.

Les notions d'*épisodes* et d'*événements* utilisés par Thériault ne sont cependant présentés qu'en lien avec un seul cas d'application et demandent donc à être redéfinies de manière plus générique.

**Méta-modèle** Les méta-modèles utilisés dans les propositions présentées ici sont : UML ([Bogorny et al., 2014]), le modèle entité relation (Spaccapietra et al. [2008]) (Thériault, 1999, 2002) ou bien un langage de description d'ontologie (Hu et al. [2013], Baglioni et al. [2008]). Les travaux proposant des ontologies se distinguent i) car ils permettent de caractériser une sémantique complexe grâce aux langages du Web sémantique et à la réutilisation d'ontologies existantes ii) car ils permettent une interopérabilité avec des données externes lors de la phase d'exploitation des données.

*Design pattern* À l'exception du modèle de Thériault [Thériault et al., 2002, 1999], la plupart des modèles proposés sont des *design patterns*. Ces patrons de conception sont destinés à être appliqués pour créer un modèle valide pour un cas d'application. Dans le cas des ontologies, les classes proposées dans les *patterns* servent de point d'ancrage pour l'utilisation d'ontologies de domaine qui permettent la description d'une sémantique complexe pour une thématique particulière. Les points de vue thématiques selon lesquels sont observés une trajectoire de vie sont différents selon le domaine d'application considéré, et ce procédé est donc particulièrement adapté pour la modélisation des trajectoires de vie. Le choix d'une approche par *design pattern* permet une grande généralité et est structurant pour l'approche globale, qui intègre l'acquisition et l'exploitation des données. Nous abordons ce sujet dans la section 3 de ce chapitre, consacrée aux démarches globales dans l'état de l'art. Auparavant, nous nous intéressons aux modèles conceptuels.

## 2.2 Analyse comparative des modèles conceptuels

Le tableau 2.2 compare les capacités de modélisation des différents modèles présentés dans l'état de l'art selon les critères suivants : prise en compte de multiples points de vue thématiques, modélisation de trajectoire géographique, de transitions entre épisodes, capacité à supporter de multiples niveaux de granularité et facteurs explicatifs d'événements de la trajectoire. Nous discutons des capacités des modèles pour chacun de ces critères puis nous mettons en évidence les défis restant à relever dans lesquels s'inscrit notre contribution.

	Points de vue	Trajectoires agéographiques	Transitions	Facteurs explicatifs	Granularité
[Spaccapietra et al., 2008]	non	non	non	non	non
[Bogorny et al., 2014]	oui	non	non	non	non
[Fileto et al., 2015]	non	en partie	non	non	en partie
[Thériault et al., 2002]	oui	oui	oui	non	non

Table 4.2 – Comparatif des capacités de modélisation

**Les multiples thématiques** La notion de multiples thématiques renvoie à celle des multiples *aspects* d'une trajectoire. De récents travaux ont abordé la modélisation de trajectoires multi-aspects [Bogorny et al., 2014], mais cela reste un challenge important [Ferrero et al., 2016]. Cette notion renvoie à celle de la généricité du modèle et de l'approche à adopter. En effet, on ne sait pas à l'avance quelles seront les thématiques utiles pour les trajectoires de vie d'un cas d'application donné. Un modèle conceptuel adapté à différents cas d'application doit donc permettre de caractériser génériquement différentes thématiques. Nous traitons la question de la généricité de l'approche dans la partie 3.3. Concernant le modèle conceptuel en lui-même, la notion de sous-trajectoire, proposée par Bogorny et al. [2014] ne permet pas de caractériser des aspects agéographiques et limite la sémantique à la notion de *but* de la trajectoire, en décrivant à l'aide d'un attribut l'activité à laquelle la (sous-) trajectoire est dédiée.

Dans notre proposition, nous utilisons la notion de point de vue [Ullman, 1988] pour traiter la question des multiples thématiques d'une trajectoire de vie. Les points de vue thématiques sont ainsi caractérisés par plusieurs attributs. L'utilisation du mécanisme de *stamping*, tiré de la multireprésentation

(Parent, 2006) permet de définir une ontologie intégrant de multiples point de vue, dont chacun correspond à une thématique.

**Trajectoires métaphoriques et géographiques** Les modèles de trajectoires sémantiques ne permettent pas de caractériser des trajectoires métaphoriques. Spaccapietra et al. [2008] proposent de redéfinir dans un espace abstrait les notions de *stops* et de *moves* sur lesquelles repose leur modèle. Le modèle proposé par Thériault et al. [1999, 2002] permet, lui, de caractériser à la fois des trajectoires métaphoriques et des trajectoires géographiques. Afin d'être adaptable à différents cas d'application, il peut cependant bénéficier des pratiques de modélisation issues de l'état de l'art des trajectoires sémantiques. Dans notre proposition, nous nous inspirons de l'approche de Thériault et al. [2002], et nous généralisons les notions d'*épisodes* et d'*événements* présentées dans ces travaux pour proposer un *design pattern* valable pour les trajectoires métaphoriques, géographiques ou agéographiques.

**Trajectoires discrètes et transitions** Les modèles de trajectoires sémantiques reposant sur la time-geography ne sont pas adaptés pour caractériser les transitions entre deux épisodes d'une trajectoire ce que est nécessaire dans un espace discret. Le modèle de Thériault permet de caractériser des trajectoires discrètes grâce aux notions d'*épisodes*, et surtout d'*événements*, qui caractérisent les transitions entre épisodes, mais encore une fois les concepts qu'il utilise ne sont pas présentés à un niveau suffisamment générique. De plus la description d'un événement, qui ne comprend que son *type* demande à être étendue. Nous redéfinissons de manière générique le concept d'événement proposé par Thériault (2002) afin de disposer d'un concept caractérisant les événements de vie que constituent ces transitions entre les épisodes d'une trajectoire pour différents cas d'application.

**Explications des événements de vie** De la même façon, les modèles de trajectoires sémantiques ne permettent pas de caractériser des causalités pour ces événements de vie. Le modèle de trajectoire de vie de Thériault est utilisé au sein d'une approche s'intéressant aux conditions de survenue des événements, mais ne permet pas de modéliser directement des explications pour ces événements. Notre objectif est ici de proposer un modèle permettant de caractériser et expliquer les événements de vie, à la fois en fonction des multiples thématiques de la trajectoire et de l'environnement dans lequel l'individu évolue. Nous présentons ce modèle dans la section 3 du chapitre V.

### **3. Approches de modélisation, d'acquisition et d'exploitation des données**

---

**Niveaux de granularité des trajectoires** La plupart des modèles présentés dans l'état de l'art ne permettent pas de modéliser les éléments d'une trajectoire à différents niveaux de granularité. Pour qu'un modèle de trajectoire supporte différents niveaux de granularité, les concepts sur lesquels ils reposent doivent pouvoir caractériser ces différents niveaux ou bien les attributs eux même doivent pouvoir avoir des valeurs à différents niveaux de granularité. Le Web sémantique propose des ontologies qui caractérisent l'information spatiale et temporelle à différents niveaux de granularité. Grâce à certaines relations, une ontologie peut également décrire différents niveaux de granularité pour une information thématique. C'est le cas par exemple de l'ontologie des Professions et Catégories Socioprofessionnelles<sup>2</sup> de l'INSEE. Notre objectif est ici de proposer une approche utilisant le web sémantique permettant de modéliser des trajectoires contenant une information multigranulaire, à la fois spatiale, temporelle et thématique.

#### **2.3 Conclusion**

Nous avons résumé dans cette partie les limites des capacités de caractérisation des modèles conceptuels de trajectoires pour les trajectoires de vie, et ceci en fonction des particularités essentielles de celles-ci. Pour chacune de ces particularités, nous avons introduit où se situe notre contribution. Dans la section suivante, nous abordons plus généralement les approches pour la modélisation, l'acquisition et l'exploitation de trajectoires de vie.

## **3 Approches de modélisation, d'acquisition et d'exploitation des données**

### **3.1 Introduction**

Nous présentons ici ce que nous appelons les *approches globales d'études des trajectoires*. Une approche globale d'étude de trajectoires comprend la modélisation, l'acquisition des données et leur exploitation. Nous nous concentrons d'abord sur les approches mises en œuvre dans l'état de l'art, en en pointant particulièrement les limites, puis nous nous intéressons à l'approche que nous mettons en œuvre dans ces travaux, en mettant en avant ce qui fait sa généralité.

---

2. <http://rdf.insee.fr/codes/index.html>

### 3.2 Limites des approches présentés dans l'état de l'art

**Les trajectoires sémantiques** Les travaux sur les trajectoires spatio-temporelles proposent des solutions et des architectures adaptées pour les trajectoires géographiques (Yan, 2011). Les différentes méthodologies proposées utilisent des données GPS brutes et proposent des méthodes de structuration en épisodes de ces trajectoires spatio-temporelles. Les trajectoires structurées sont ensuite enrichies, principalement d'informations géographiques, avant d'être analysées. Une telle méthodologie n'est pas adaptée dans le cas des trajectoires de vie, quand les informations brutes concernant les trajectoires ne se limitent pas à des localisations. De nouvelles méthodologies doivent donc être proposées, comprenant les différentes étapes importantes pour l'étude des trajectoires de vie : la modélisation, l'acquisition de données et leur exploitation. Pour atteindre cet objectif, nous allons examiner comment les approches très génériques utilisées pour les trajectoires sémantiques peuvent être adaptées pour les trajectoires de vie, en nous concentrant sur les approches proposant un *design pattern*.

Dans les différents travaux présentés, les auteurs montrent l'applicabilité de leur *design pattern* en l'appliquant pour un ou deux cas d'application. Pour chaque cas d'application, en plus de leur patron de conception, les auteurs fournissent un modèle dédié à un cas d'application, ainsi que des requêtes d'exploitation démontrant les potentialités du modèles. Ces requêtes d'exploitation ne sont donc pas présentées à un niveau générique. [Jin and Claramunt \[2018\]](#) pointent d'ailleurs le manque d'opérateurs génériques dans les propositions faites par les chercheurs de ce domaine. La figure 4.2 présente les différents composants qui sont généralement proposés par les auteurs dans une approche d'étude des trajectoires utilisant un *design pattern* : le patron de conception lui-même, le modèle dédié à un cas d'application et les requêtes d'exploitation, elle aussi dédiées à un cas d'application.



**Figure 4.1** – Les différents composants nécessaires à la gestion de données dans une approche d'étude des trajectoires par *design pattern* dans les travaux existants

Dans notre cas, l'approche doit cependant être complétée en raison de la nature de l'acquisition de données qui prend la forme d'une collecte auprès d'individus. Il ne s'agit donc pas de structurer des données brutes comme dans le

### 3. Approches de modélisation, d'acquisition et d'exploitation des données

cas des trajectoires sémantiques. Des requêtes dédiées à la collecte de données doivent donc également être proposées.

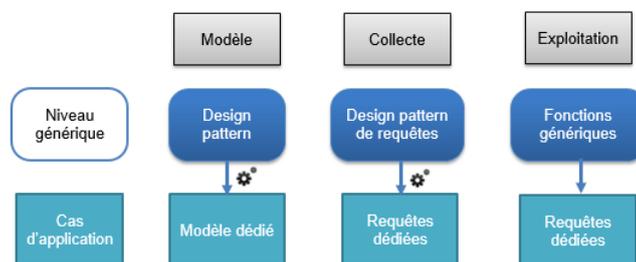


**Figure 4.2** – Les différents composants nécessaires à la gestion de données dans une approche d'étude des trajectoires par *design pattern* intégrant la collecte de données

Les approches présentées ici n'intègrent pas les éléments nécessaires à une totale généricité de l'approche, au niveau du modèle, mais aussi au niveau de l'exploitation et de l'acquisition des données. Nous présentons dans la partie suivante notre approche pour intégrer de la généricité à ces différents niveaux.

### 3.3 Vue générale de notre approche

Nous présentons ici brièvement notre approche, ainsi que l'outillage que nous offrons pour la mettre en place. À un niveau générique, nous proposons des patrons de conception de trajectoires de vie, ainsi que des patrons de conception de requêtes de collecte et d'exploitation de données. Des algorithmes dédiés permettent de passer du niveau générique à un cas d'application en fonction de certains paramètres. Nous proposons une méthodologie et un outillage permettant de mettre en œuvre notre approche lors des trois étapes de modélisation, acquisition et exploitation des données. Nous présentons notre contribution pour chacune de ces étapes.



**Figure 4.3** – Vue générale de notre approche et de l'outillage mis en place

**Modélisation** Concernant la modélisation de trajectoires de vie, nous proposons un *design pattern* de trajectoires, ainsi que des algorithmes permettant de créer un modèle dédié par application et paramétrages de ces *patterns*.

Dans un cas d'application, nous montrons ensuite comme sont utilisés ces algorithmes. Les contributions liées au modèle de trajectoires de vie que nous proposons ont été présentées dans la section 2 de ce chapitre, dédiée aux modèles conceptuels.

**Acquisition des données** Notre proposition comprend également des *design patterns* de requêtes pour l'acquisition des données, qui sont valables pour n'importe quel modèle issu du *design pattern* de trajectoire proposé. Ces requêtes permettent l'acquisition de données multigranulaires grâce à l'utilisation d'ontologies caractérisant différents niveaux de granularité temporels, spatiaux et thématiques, et ce selon de multiples points de vue. Des algorithmes sont également proposés pour générer les requêtes dédiées automatiques à partir des *patterns* de requêtes. Notre proposition met l'accent sur la possibilité d'une collecte de données directement auprès des individus, en utilisant le paradigme de l'information géographique volontaire (Volunteered Geographic Information ou VGI).

**Exploitation des données** Enfin, nous proposons également des fonctions génériques l'exploitation des données. Ces fonctions sont utilisables pour tous les modèles de trajectoires de vie dédiés à un cas d'application issus du patron de conception que nous proposons. Elles sont implémentées dans le langage SPARQL.

### 3.4 Conclusion

Nous avons donc pointé le fait que les approches d'étude de trajectoires sémantiques ne permettent pas d'étudier les trajectoires de vie. Nous avons ensuite situé l'approche originale que nous présentons, qui s'appuie sur des techniques de modélisation utilisées pour les trajectoires sémantiques. Nous avons finalement présenté succinctement ce que permet notre contribution pour la modélisation, l'acquisition et l'exploitation des données.

## 4 Conclusion

Nous proposons donc une approche globale permettant la modélisation, l'acquisition et l'exploitation de trajectoires de vie. Cette approche repose sur un modèle conceptuel de trajectoires de vie ainsi que sur une méthodologie sur laquelle s'appuyer pour les étudier. La mise en place de cette méthodologie est possible grâce à un outillage permettant de modéliser ces trajectoires

multi-thématiques, d'acquérir des données et de les exploiter selon différents niveaux de granularité.

Notre modèle conceptuel est basée sur celui de Thériault, dont nous avons mis en avant l'intérêt pour la modélisation générique de trajectoires géographiques et métaphoriques, mais aussi les limites. Notre modèle intègre également des approches qui ont fait leurs preuves pour la modélisation générique de trajectoires sémantiques. La création de ce nouveau modèle, sous la forme d'une ontologie, permet l'utilisation des ressources du web sémantique. Dans le prochain chapitre, nous présentons ce modèle conceptuel, patron de conception de trajectoire de vie.



# Contributions



# Patron et modèle socles l'approche

---

## Sommaire

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>71</b>
<b>2</b>	<b>Patron de conception de trajectoire de vie</b> . . . . .	<b>72</b>
2.1	Notions d'épisodes et d'événements . . . . .	72
2.2	Trajectoire de vie métaphorique . . . . .	75
2.3	Trajectoire de vie géographique . . . . .	76
2.4	Conclusion . . . . .	77
<b>3</b>	<b>Modèle de facteurs explicatifs</b> . . . . .	<b>77</b>
3.1	La notion de facteurs explicatifs . . . . .	77
3.2	Typologie des facteurs explicatifs . . . . .	81
3.3	Conclusion . . . . .	82
<b>4</b>	<b>Conclusion</b> . . . . .	<b>82</b>

---

## 1 Introduction

Nous présentons ici 1) sur un patron de conception d'ontologie dont l'application permet la modélisation d'une trajectoire thématique 2) sur un modèle de facteurs explicatifs, qui permet d'expliquer les événements qui surviennent dans une trajectoire de vie. Le patron de conception d'ontologie doit être appliquée successivement pour chacune des thématiques d'une trajectoire de vie. Il permet d'obtenir autant de modèles de trajectoires thématiques qui combinés constituent un modèle de trajectoires de vie. Chacune des trajectoires thématiques modélisées grâce au patron peut-être géographique (sous-section 2.3) ou agéographique (sous section 2.2). L'information contenue dans ces trajectoires peut être caractérisée à différents niveaux de granularité.

Une fois l'ontologie de trajectoire de vie obtenue, le modèle de facteurs explicatifs doit être utilisé pour intégrer des facteurs explicatifs pour les événements caractérisés par le modèle de trajectoire de vie.

Nous présentant dans le chapitre VI la démarche méthodologique permettant l'application du patron et l'intégration des facteurs explicatifs.

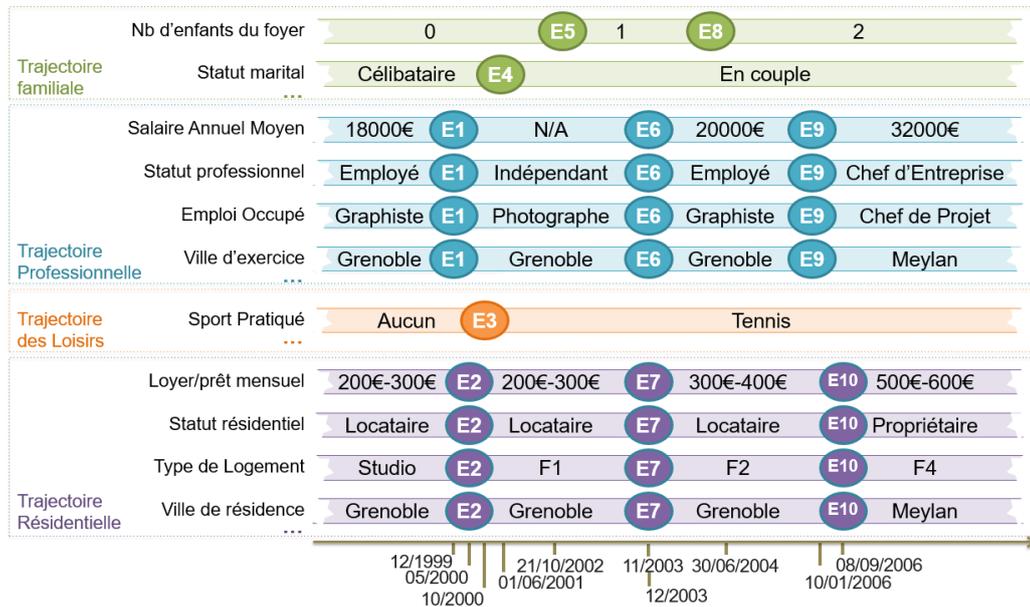
## 2 Patron de conception de trajectoire de vie

Dans cette section, nous présentons notre patron de conception d'ontologies qui permet par applications successives la création d'un modèle de trajectoire de vie. Nous définissons d'abord les principaux concepts sur lesquels reposent notre patron. Dans un second temps, nous présentons l'extrait du patron permettant de modéliser des trajectoires thématiques agéographiques. Enfin, nous présentons l'ensemble du patron, en nous concentrant sur les classes permettant de modéliser des trajectoires thématiques géographiques.

### 2.1 Notions d'épisodes et d'événements

Pour traiter la question des différentes thématiques d'une trajectoire de vie, nous nous appuyons sur la notion de point de vue (voir [Drews, 1993]). Nous montrons ici comment l'utilisation de la notion de point de vue, présentée dans la section 2, permet de caractériser les différents *aspects* d'une trajectoire au sens de [Ferrero et al., 2016], dont chacun correspond à une thématique de la trajectoire de vie. Nous considérons ainsi que chaque trajectoire thématique d'une trajectoire de vie est un point de vue selon lequel est observée une trajectoire. Nous définissons ici les concepts clés de notre proposition, qui sont utilisés pour former le *design pattern*.

Dans notre modèle, chaque trajectoire thématique est composée d'épisodes qui décrivent l'état d'une personne considérée selon un point de vue particulier. Entre les épisodes, une relation d'ordre non stricte peut être établie grâce à l'algèbre d'Allen (Allen, 1983). Conceptuellement, notre approche associe les conceptions du temps de Leibniz et de Newton, le premier considérant le temps comme une succession d'événements, et le second comme un flux mesurable en valeur absolue. En effet, nous considérons l'évolution du temps comme une succession d'évènements, mais sans pour autant renoncer à le mesurer, grâce aux unités dont nous disposons (année, jour, heure...). La figure 5.1 présente un exemple de trajectoire de vie contenant quatre trajectoires thématiques : familiale, professionnelle, loisirs et résidentielle. Nous utilisons cette figure comme exemple pour illustrer les concepts que nous définissons ci-dessous.



**Figure 5.1** – extrait d'une trajectoire de vie centrée sur l'aspect résidentiel. Représentation inspirée de Thériault [Thériault et al., 1999]

**Épisode.** Un épisode correspond à l'état stable d'une personne observée d'un certain point de vue pendant un intervalle de temps. L'état d'un point de vue est caractérisé par les valeurs des attributs qui le composent. Il est considéré comme *stable* quand celles-ci restent inchangées au cours du temps. La durée de l'épisode correspond à un intervalle de temps  $A$  compris entre deux instants  $ts$  et  $te$  (tel que  $ts < te$ ). Chacun de ces deux instants peut correspondre à un événement dans la trajectoire.

Prenons l'exemple de la trajectoire professionnelle sur figure 5.1. Cette trajectoire thématique est décrite par quatre attributs appartenant au point de vue professionnel : *Salaire Annuel Moyen*, *Statut Professionnel*, *Emploi Occupé*, et *Ville d'exercice*. Cette trajectoire comprend quatre épisodes, lors desquels les valeurs respectives de ces attributs ne changent pas.

**Événement.** Les événements correspondent à quelque chose qui survient et qui peut être observé (Abler, 1971).

Nous modélisons ici les événements internes à la vie d'un individu et les événements externes à sa vie, mais qui peuvent avoir une influence sur sa trajectoire de vie.

**Événement interne.** Un événement interne marque le début et la fin d'un épisode. Il est rattaché à un point de vue et se traduit par le passage d'un état stable à un autre sur ce point de vue. L'événement interne provoque donc un changement dans au moins une valeur d'attribut. Il a un type et est associé à un chronon  $t$  pertinent pour le phénomène observé (ce qui permet de le situer temporellement).

Sur l'exemple, sur la figure 5.1, l'événement E5 est un événement *naissance* rattaché au point de vue familial et entraîne la modification de l'attribut donnant le nombre d'enfants du ménage.

**Événement externe.** Un événement externe a un type et est associé à un chronon  $t$  pertinent pour le phénomène observé (ce qui permet de le situer temporellement).

Un événement externe peut être rapporté par un individu, mais n'implique pas *directement* de modification de son état (au sens de modification de valeurs d'attributs). S'il ne fait pas intrinsèquement partie de la trajectoire de vie d'un individu, un événement externe peut néanmoins influencer notablement celle-ci (par exemple, pour le cas des trajectoire résidentielle, le fait que les conditions de prêts bancaires soient favorablement réévaluées peut être à l'origine d'un achat immobilier).

Le changement de valeur d'un des attributs décrivant l'individu entraîne donc un changement d'épisode et la caractérisation d'un événement. Ce choix de modélisation impose de bien choisir l'information à représenter (quels sont les attributs pertinents), mais aussi de déterminer le degré de précision pertinent. Ainsi, concernant, par exemple, le loyer d'un logement, ce n'est pas le montant exact qui est intéressant mais plutôt une fourchette de valeurs. Cela présente l'avantage d'en assurer une certaine stabilité. Il est en effet inutile à l'échelle de la vie d'enregistrer un nouvel épisode car le loyer a augmenté faiblement. Le patron que nous proposons permet de modéliser des trajectoires agéographiques et des trajectoires géographiques. Nous présentons ici successivement ces deux parties. Les différents concepts utilisés dans l'ontologie sont formalisés en utilisant le modèle Ressource Description Framework (RDF)<sup>1</sup>. La formalisation de l'ontologie et de ses concepts est disponible dans l'annexe A).

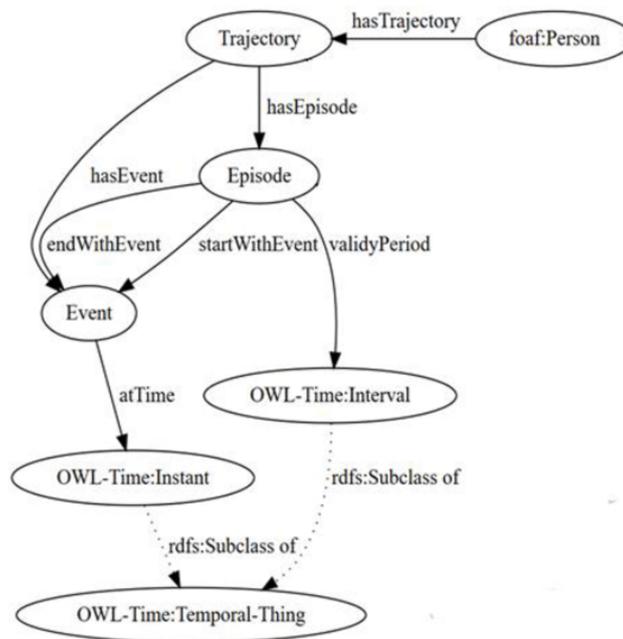
---

1. <https://www.w3.org/2001/sw/wiki/RDF>

## 2.2 Trajectoire de vie métaphorique

La première partie du patron de conception d'ontologie que nous proposons est présentée sur la figure 5.2. Elle peut être appliquée pour modéliser une trajectoire thématique métaphorique. Les concepts principaux qui composent ce patron sont les épisodes (*episode*) et les événements (*event*). Les classes et les relations qui composent le *design pattern* sont décrites ci-dessous.

**Trajectory.** Cette classe correspond à la notion de *trajectoire thématique métaphorique*. Elle est associée à une classe *foaf:Person* par une propriété *hasTrajectory*. Elle est reliée aux épisodes (classe *Episode*) et aux événements (classe *Event*) grâce aux propriétés *hasEpisode* and *hasEvent*.



**Figure 5.2** – Le *design pattern* de trajectoires de vie métaphorique

**Episode.** Cette classe correspond à la notion d'épisode précédemment introduite. La propriété *validityPeriod* caractérise la durée de l'épisode qui correspond à un intervalle de temps *OWL-Time:Interval*. Un épisode commence avec un événement et se termine avec un événement (propriétés *startWith* et *endWith*).

**Event.** Cette classe correspond à la notion d'événement précédemment introduite. La propriété *atTime* associe un événement avec un *chronon* (Abler, 1971) pertinent pour le phénomène observé (qui permet de dater

l'événement).

La dimension temporelle des épisodes et des événements est caractérisée en utilisant l'ontologie OWL-Time<sup>2</sup>.

### 2.3 Trajectoire de vie géographique

Comme nous l'avons défini précédemment, nous considérons qu'une trajectoire géographique est un cas particulier de trajectoire métaphorique. Cela explique que le patron dédié aux trajectoires géographiques soit une spécialisation de celui dédié aux trajectoires métaphoriques. Le patron a donc une seconde partie qui, une fois appliquée, permet de caractériser les trajectoires thématiques géographiques. La classe caractérisant cette notion, ainsi que celle d'épisode géographique, sont des spécialisations des classes trajectoires et épisodes précédemment présentées. Nous présentons ici ces différentes classes.

Si la trajectoire à modéliser a une composante géographique, une classe plus spécifique caractérisant les trajectoires doit donc être utilisée, la classe *Spatial-Trajectory*. De même, pour chaque épisode de la trajectoire, la classe *SpatialEpisode* doit être utilisée. Celle-ci est liée à la classe de l'ontologie Geosparql<sup>3</sup>, qui décrit la composante spatiale de l'épisode. Finalement, l'épisode est relié à l'ontologie Geofla de l'Institut national de l'information géographique et forestière (IGN), qui décrit les divisions administratives françaises. L'utilisation de cette ontologie est réservée aux cas d'application localisés en France ; pour les autres cas, une autre ontologie caractérisant les divisions administratives sur le même modèle peut-être utilisée.

---

2. <https://www.w3.org/TR/owl-time/>

3. <http://www.opengis.net/ont/geosparql>

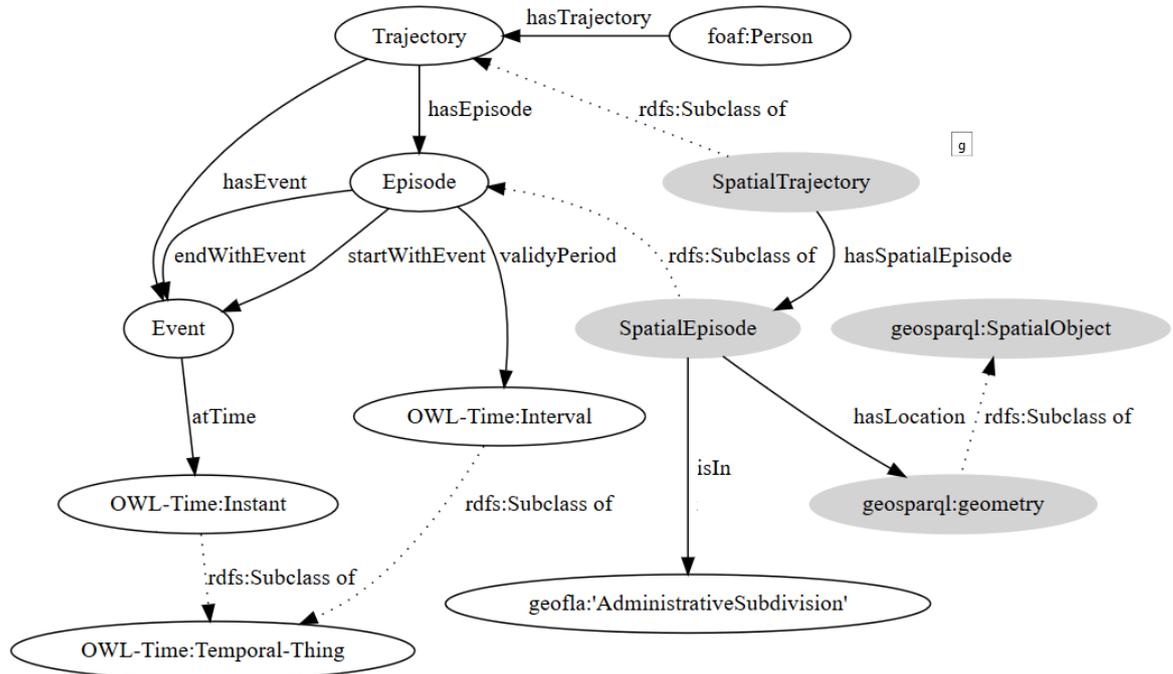


Figure 5.3 – Le design pattern de trajectoires de vie

## 2.4 Conclusion

Ce *design pattern* permet de modéliser de manière générique une trajectoire métaphorique et une trajectoire géographique, la trajectoire géographique étant considérée comme une spécialisation du concept plus général de trajectoire métaphorique. Nous décrivons dans le chapitre VI comment ce *design pattern* peut être utilisé pour obtenir un modèle de trajectoire de vie pour un cas d'application particulier, en utilisant notamment la technique du *stamping* proposé par [Vangenot et al., 2002], pour supporter les multiples points de vue d'une trajectoire. Auparavant, nous présentons un modèle permettant de caractériser des facteurs explicatifs pour les événements d'une trajectoire.

## 3 Modèle de facteurs explicatifs

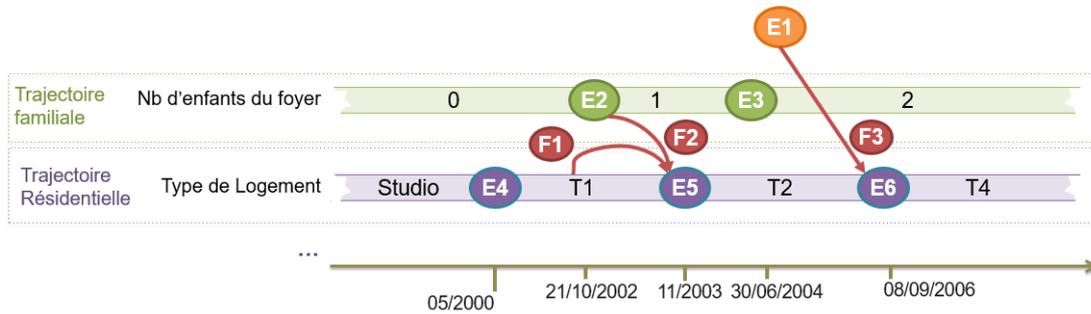
### 3.1 La notion de facteurs explicatifs

La trajectoire de vie d'une personne est donc marquée par des événements qui viennent modifier son état d'un point de vue donné. Un des enjeux de l'étude des trajectoires de vie est d'expliquer ces changements d'état. Les expliquer, c'est donner davantage de sens à la trajectoire de vie que celui qui est

caractérisé par la sémantique fournie par les seuls attributs. C'est également s'approcher davantage de la notion de *carrière* de Becker Howard [1985], que nous citons en introduction, en modélisant "les facteurs dont dépend la mobilité d'une position [sociale, professionnelle...] à l'autre". Pour le cas d'application de la trajectoire résidentielle, c'est permettre de *donner du sens* [Authier et al., 2012], à la trajectoire résidentielle. La modélisation des *facteurs explicatifs* pour ces changements est une étape nécessaire pour pouvoir les exploiter ou encore les inférer à partir d'une collection de trajectoires.

Dans notre modèle, nous avons choisi d'associer les facteurs explicatifs au événements. En effet, ici, expliquer les événements, c'est expliquer le passage d'un épisode à un autre. Modéliser les facteurs explicatifs des événements revient donc à modéliser des éléments d'explication, de la fin d'un épisode et du début d'un autre.

Les facteurs explicatifs peuvent être *internes* ou *externes*, c'est-à-dire qu'ils peuvent soit être relatifs aux circonstances de vie de l'individu, soit dépendre de circonstances auxquelles ils ne sont pas liés directement. Les facteurs internes pourront être liés à des événements internes (la naissance d'un enfant, un changement de situation professionnelle...). Mais les facteurs explicatifs de la trajectoire de vie peuvent également être trouvés dans des conditions de l'environnement (économique, social, etc.) dans lequel évolue l'individu. Nous ne modélisons ces facteurs externes que dans la mesure où les individus se sentent impactés par ceux-ci, ce qui signifie que l'individu est capable de rapporter un événement externe à sa trajectoire. Par exemple, en expliquant les motivations qui l'ont poussé à déménager (événement de vie) un individu pourra citer des effractions dans son quartier ayant déclenché chez lui un sentiment d'insécurité. Les facteurs externes seront donc liés à des événements externes. La figure 5.4 présente des exemples de facteurs explicatifs internes et externes. Cette figure illustre de manière très simplifiée deux trajectoires thématiques (familiale et résidentielle) qui caractérisent des épisodes et des événements de vie de l'individu Thomas. Elle présente également un événement externe (E1) qui caractérise des effractions ayant eu lieu dans son quartier. Dans cet exemple, les facteurs explicatifs F1 et F2 sont des facteurs explicatifs internes de l'événement E5 alors que le facteur explicatif F3 est un facteur externe de l'événement E6.



**Figure 5.4** – Exemples de facteurs explicatifs

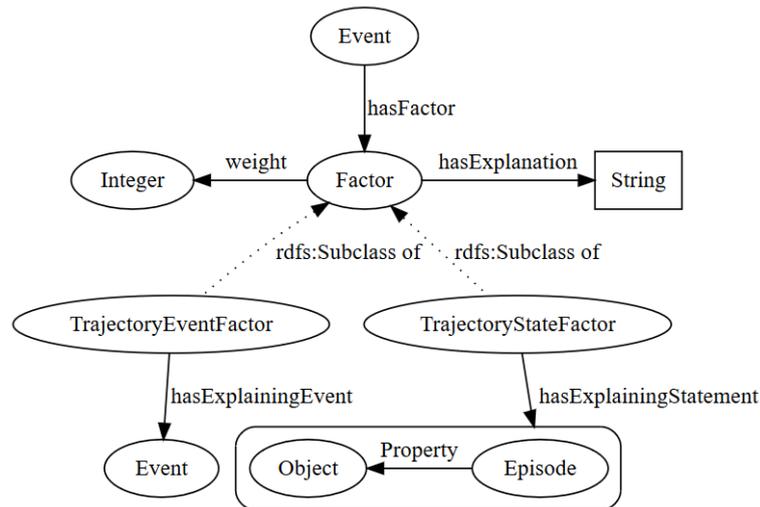
Un facteur explicatif permet d’associer un événement de vie qu’il explique à un événement ou épisode, qui en est tout ou partie de l’explication. Ainsi, un événement *expliqué* est associé par le facteur explicatif à un événement ou un épisode *explicatif*. Par exemple, sur la figure 5.4, l’événement E5 est *expliqué* par deux facteurs explicatifs. Le facteur F1 associe l’événement E5 à un épisode, et, plus particulièrement, à un *statut* de cet épisode (qui caractérise le *type de logement* comme un T1). Le facteur F2 associe, quant à lui, à l’événement E5 un autre événement, E2, qui caractérise une *naissance* dans la trajectoire familiale. Chaque événement peut ainsi être expliqué par plusieurs facteurs.

En résumé, le *design pattern* présenté ici permet donc de modéliser des facteurs explicatifs d’événements de trajectoires. Dans notre approche, nous considérons qu’un événement peut être expliqué soit :

- Par d’autres événements de vie. Par exemple, Thomas déménage pour un appartement plus grand (événement) car une naissance a eu lieu, c’est le cas du facteur explicatif F2 sur la figure 5.4.
- Par les caractéristiques d’un épisode. Par exemple, Thomas déménage pour un appartement de type T2 car son précédent appartement était de type T1. Cet état de fait est porté par un épisode : c’est l’exemple du facteur explicatif F1 sur la figure 5.4.
- Par des événements externes. Par exemple, Thomas déménage suite à un événement externe ayant engendré chez lui un sentiment d’insécurité. C’est l’exemple du facteur explicatif F3 sur la figure 5.4.

Des combinaisons entre de telles explications sont bien entendues possibles. Par exemple, i) Thomas a eu une promotion (événement expliqué), ii) il a eu un enfant (événement de vie) et iii) son appartement précédent était trop petit (caractéristique de l’épisode précédent : type de logement=T1), il a donc déménagé dans un appartement plus grand.

Notre modèle de facteurs explicatifs est présenté sur la figure 5.5. Les classes et les propriétés de ce modèle sont décrites ci-dessous.



**Figure 5.5** – Modèle de facteurs explicatifs de trajectoires

Notre modèle de facteurs explicatifs peut s'appliquer à toute trajectoire de vie une fois que le *design pattern* précédemment présenté est appliqué. Dans la figure 5.5, la classe Event en haut de l'image correspond à l'événement à expliquer.

**Factor** est la classe qui caractérise un facteur explicatif d'un événement. La classe Event caractérisant l'événement à expliquer est liée à un facteur explicatif *Factor* par une propriété *hasFactor*. La relation *hasExplanation* (propriété de donnée ou *data property*) associe un *Factor* à une chaîne de caractères (type *String*) qui fournit si nécessaire une information additionnelle à propos d'un facteur explicatif particulier. La relation *weight* (propriété de donnée ou *data property*) associe un facteur explicatif avec un entier (type *Integer*) qui permet d'exprimer le poids de ce facteur dans la survenue de l'événement expliqué. Quand plusieurs facteurs explicatifs existent pour le même événement, cette propriété permet une compréhension plus fine de l'influence de chacun de ces facteurs et permet ainsi de discriminer les événements perçus comme secondaires et les événements perçus comme majeurs par l'individu.

### 3.2 Typologie des facteurs explicatifs

La classe *Factor* est étendue par deux autres classes : *TrajectoryEventFactor* et *TrajectoryStateFactor*. Ces deux classes correspondent chacune aux deux manières d'expliquer un événement qui ont été décrites précédemment : par d'autres événements ou bien par les caractéristiques d'épisode.

**TrajectoryEventFactor** caractérise un facteur explicatif qui est associé à un événement explicatif. Cet événement explicatif appartient à la trajectoire de cet individu ou d'un autre individu. Par exemple, le déménagement de Thomas (l'événement expliqué) peut-être expliqué sa propre *promotion* (un événement explicatif de sa trajectoire professionnelle). Celui-ci peut également être expliqué par la *promotion* de sa femme (un événement explicatif de la trajectoire professionnelle de sa femme). Un facteur explicatif *TrajectoryEventFactor* est donc associé à un événement explicatif par une relation *hasEvent*. Cette classe a deux sous classes *InternalEventFactor* et *NetworkEventFactor*. Ces deux concepts permettent de supporter les deux exemples donnés ci-dessus.

**InternalEventFactor** est associé à un événement explicatif appartenant à la trajectoire du même individu. Dans l'exemple précédent à propos de l'individu Thomas, le facteur explicatif qui fait référence à la naissance d'un enfant comme événement explicatif est une instance de la classe *InternalEventFactor*.

**NetworkEventFactor** est associé à un événement explicatif appartenant à la trajectoire d'un autre individu. Dans l'exemple précédent, le facteur explicatif faisant référence à la promotion de l'épouse de Thomas est une instance de la classe *NetworkEventFactor*. Les deux individus associés (directement ou indirectement) forment par conséquent tout ou partie d'un réseau. De telles relations peuvent être exprimées en utilisant l'ontologie Friend of a Friend<sup>4</sup> (foaf).

Le prochain type de facteurs explicatifs que nous décrivons utilise la réification RDF, qui permet de faire une déclaration à propos d'une autre déclaration (un triplet sujet-prédicat-objet se voit assigner une URI et est traité comme une ressource que l'on peut donc utiliser dans une autre déclaration). Cela permet à une propriété d'avoir pour co-domaine, un

---

4. <http://xmlns.com/foaf/spec/>

*statement*, c'est-à-dire un triplet décrivant lui même quelque chose. Ici, il s'agit d'associer un facteur explicatif à un *statement* décrivant un état, c'est-à-dire un statut valable dans un épisode.

**TrajectoryStateFactor** caractérise un facteur explicatif qui a un *statement* (triplet sujet-prédicat-objet) explicatif qui décrit un état dans la trajectoire de l'individu. Par exemple, un déménagement est expliqué par certaines caractéristiques de l'épisode résidentiel précédent comme, par exemple, un nombre de pièces trop petit.

*TrajectoryStateFactor* est donc associé à un *statement expliquant* par une relation `hasStatement`.

### 3.3 Conclusion

Nous avons présenté un modèle de facteurs explicatifs. La notion de facteurs explicatifs se décline en une typologie qui permet de caractériser des facteurs explicatifs de natures différentes, selon qu'ils se réfèrent notamment à un événement ou à un épisode, les deux pouvant contenir des explications pour un changement d'état dans une trajectoire. Nous présentons dans le chapitre VI comment ce modèle peut être instancié pour caractériser les facteurs explicatifs d'événements d'une trajectoire de vie donnée, créée, quant à elle, en utilisant le patron de conception (voir section 2 de ce chapitre).

## 4 Conclusion

Nous avons présenté un patron de conception générique permettant de modéliser des trajectoires métaphoriques et des trajectoires géographiques qui ensemble constituent le modèle d'une trajectoire de vie adaptée aux besoins d'une étude. Ce modèle de trajectoire de vie est enrichie par le recours à un second modèle qui permet de caractériser des facteurs explicatifs d'événements. Nous avons décrit les principaux concepts sur lesquels repose notre approche pour la modélisation de trajectoires de vie. Le patron de conception permet de décrire une trajectoire thématique, dont les épisodes, définis grâce à la notion de *point de vue*, peuvent comprendre plusieurs statuts valables lors de ces épisodes. Nous présentons dans le chapitre VI comment des applications successives de ce patron permet de modéliser les multiples thématiques d'une

trajectoire de vie. La notion *d'événement* permet de caractériser les transitions entre épisodes. Enfin, la notion de *facteur explicatif* permet de donner du sens à ses transitions et de mieux expliquer une trajectoire de vie.

Les modèles que nous avons définis reposent sur une approche à base d'ontologies destinées à être publiées selon les principes du Web sémantique. Les ontologies définissent formellement des concepts et leurs relations (elles sont en cela un modèle) et ont vocation à être instanciées (elles véhiculent en cela des données conformes au modèle).

Nos ontologies sont placées dans le Web des Données Liées (Linked Open Data – LOD) notamment pour garantir l'interopérabilité de nos travaux. Cela passe, par exemple, par l'exploitation de descriptions existantes éprouvées et disponibles (par exemple, pour les informations spatiales et temporelles les ontologies OWL-Time et GeoSPARQL). Nous utilisons l'expressivité des langages du Web sémantique à des fins de modélisation, ce qui est valorisé par un requêtage permettant une analyse fine des données (voir section 4 du chapitre VII).

Le chapitre VII sera l'occasion de montrer comment appliquer notre approche à un cas d'étude, celui de l'étude des trajectoires résidentielles. Dans le chapitre suivant, nous décrivons la méthodologie permettant l'application du patron de conception et l'utilisation du modèle de facteurs explicatifs, ainsi que l'architecture logicielle sur laquelle elle repose.



# Méthodologie et architecture pour l'étude des trajectoires de vie

---

## Sommaire

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>85</b>
<b>2</b>	<b>Vue générale de l'approche et de l'architecture</b> . . . . .	<b>86</b>
<b>3</b>	<b>Modélisation et paramétrage de l'architecture</b> . . . . .	<b>87</b>
3.1	Introduction . . . . .	87
3.2	Principales étapes de la démarche de modélisation . . . . .	87
3.3	Fonctionnement du module <i>Ontology Manager</i> pour la modélisation . . . . .	90
3.3.1	La conception du modèle . . . . .	90
3.3.2	Configuration du module de collecte . . . . .	94
3.4	Conclusion . . . . .	97
<b>4</b>	<b>Acquisition de données</b> . . . . .	<b>97</b>
4.1	Conclusion . . . . .	100
<b>5</b>	<b>Exploitation des données</b> . . . . .	<b>100</b>
<b>6</b>	<b>Conclusion</b> . . . . .	<b>112</b>

---

## 1 Introduction

Nous proposons une approche pour l'étude des trajectoires de vie reposant sur le patron de conception et le modèle présenté dans la partie précédente (chapitre V). L'étude des trajectoires comprend leur modélisation, l'acquisition des données et l'exploitation de ces données. Nous fournissons une méthodologie supportant chacune de ces étapes. Notre approche s'appuie notamment sur une démarche de modélisation (voir figure 6.2) dont la mise en œuvre concrète est facilitée par la proposition d'une architecture logicielle (voir figure 6.1 qui vise à outiller l'ensemble de l'approche).

Nous présentons dans un premier temps une vue générale de la méthodologie et de l'architecture logicielle associée, avant de nous concentrer respectivement sur les 3 étapes outillées par notre approche : la modélisation, l'acquisition des données, puis leur exploitation.

## 2 Vue générale de l'approche et de l'architecture

Nous commençons par donner une vue générale de la méthodologie et de l'architecture logicielle que nous proposons (voir Figure 6.1). Celle-ci tire profit du Web sémantique tout au long du processus présenté ci-dessous.

Notre architecture est composée de quatre modules offrant des fonctionnalités qui peuvent être invoquées via des API (Application Programming Interface).

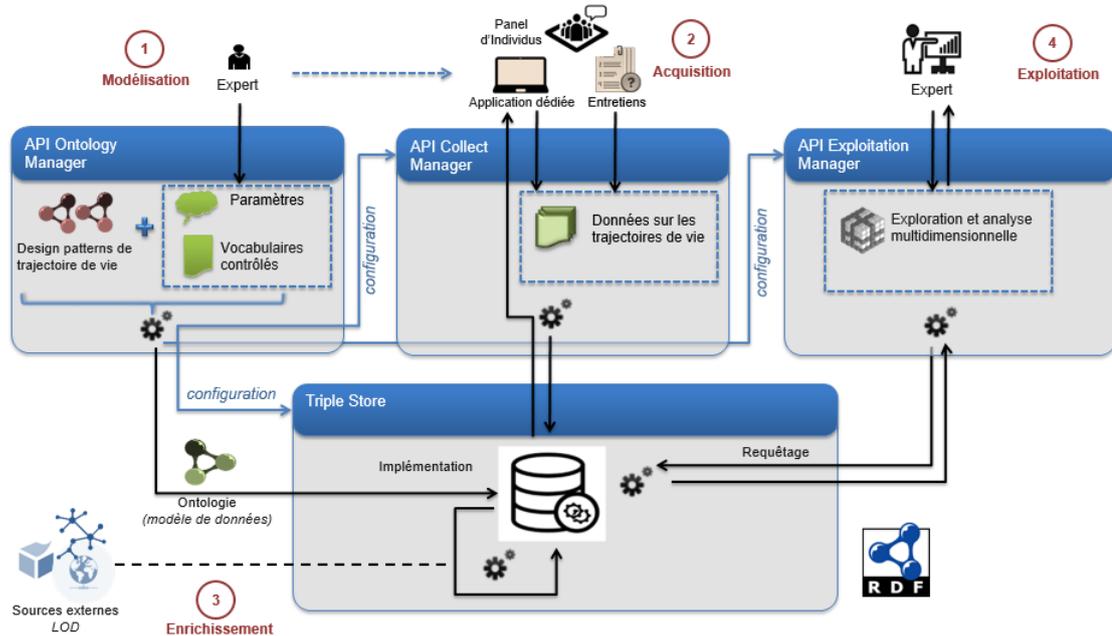
Ces fonctionnalités sont le support d'une démarche qui comprend quatre étapes illustrées par les pastilles numérotées de 1 à 4 dans la Figure 2 : 1) modélisation de la trajectoire, par un expert du domaine d'application cible, en utilisant le module *Ontology Manager* ; 2) acquisition des données qui viendront alimenter le modèle par le biais du module *Collect Manager* ; 3) enrichissement des informations collectées grâce au *Triple Store Manager* et 4) exploitation des données par des experts à des fins d'analyse à travers le module *Exploitation Manager*. Nous présentons ci-après les étapes de modélisation, de collecte, et d'exploitation des données, qui ont fait l'objet des travaux réalisés.

Le module *Ontology Manager* a un rôle central puisque, en plus de la création de l'ontologie, il assure la configuration des autres modules en fonction du cas d'application.

Cette architecture est implémentée sous la forme d'une librairie JAVA qui contient pour l'instant trois APIs, dédiées à la modélisation, l'acquisition et l'exploitation. Ces APIs utilisent elles-mêmes l'API JENA <sup>1</sup>, qui permet de manipuler des ontologies et leurs instances et de construire des applications pour le Web sémantique.

---

1. <https://jena.apache.org/>



**Figure 6.1** – Architecture pour la collecte et l'analyse de trajectoires de vie [Villanova-Oliver et al., 2018]

Dans les sections suivantes, nous présentons en détail les différentes étapes de la démarche et le fonctionnement des différents modules de l'architecture, en commençant par la modélisation.

### 3 Modélisation et paramétrage de l'architecture

#### 3.1 Introduction

Dans un premier temps, nous présentons la démarche originale permettant la création d'une ontologie de trajectoire multi-point de vue, ainsi que la configuration des autres modules de l'architecture. Dans un second temps, nous présentons les algorithmes et les méthodes utilisés par le module *Ontology Manager* lors de cette étape.

#### 3.2 Principales étapes de la démarche de modélisation

Lors de la phase de création du modèle, un expert souhaitant modéliser une trajectoire de vie pour un domaine d'application particulier utilise le module appelé *Ontology Manager*. Le module permet de faciliter la démarche de cet expert qui aura auparavant décidé de ce que doit contenir le modèle de trajectoire de vie pour le cas d'application qui l'intéresse. L'utilisation du module

est donc précédée d'une d'analyse des besoins qui aura permis de déterminer ce qu'il est nécessaire de modéliser conceptuellement. Il est ensuite possible de paramétrer les méthodes de l'*Ontology Manager* comme nous le décrivons dans cette partie. Celui-ci offre des procédures [Noel et al., 2017] qui assurent la production de l'ontologie à partir du patron de conception et en utilisant des algorithmes (algorithmes 1 et 2)). Il assure également la configuration des autres modules, qui servent de support à l'acquisition et l'exploitation des données.

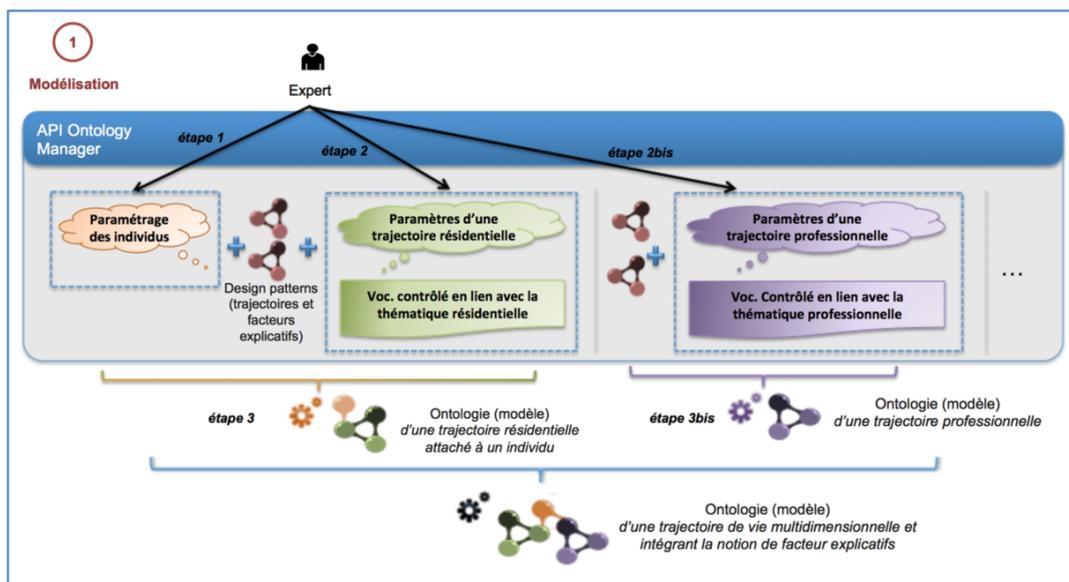


Figure 6.2 – Représentation des étapes de la démarche de modélisation [Villanova-Oliver et al., 2018]

Le patron de conception de trajectoire (présenté section 2), ainsi que le modèle de facteurs explicatifs (présenté section 3), que nous avons définis sont utilisés par le module *Ontology Manager*. La démarche d'utilisation du module, qui permet d'exploiter ces modèles est illustrée sur la figure 6.2. Le patron de trajectoires métaphoriques et trajectoires géographiques est utilisé pour définir une partie d'ontologie représentant une trajectoire thématique. Le modèle caractérisant les facteurs explicatifs permet ensuite de doter tout événement associé à une trajectoire thématique de la capacité à être utilisé comme facteur explicatif dans l'ontologie.

L'étape 1 consiste à paramétrer des attributs permettant de caractériser le "propriétaire" d'une trajectoire de vie. Il s'agit de fournir les éléments qui serviront à décrire les individus pour lesquels on crée l'ontologie : à minima

un code, invariant au cours du temps et non traçable si le respect de l'anonymat est souhaitable, sera utilisé. D'autres attributs, permettant par exemple des analyses socio-démographiques, peuvent être fournis mais il doit s'agir d'informations invariantes (une année de naissance, par exemple).

L'étape 2 est consacrée au paramétrage de la première trajectoire thématique. L'expert donne un nom (utilisé pour générer les classes de l'ontologie de trajectoire pour la thématique créée, par exemple résidentielle dans la Figure 6.2), son type (agéographique ou géographique, ce qui permet de sélectionner la bonne version du patron de trajectoires thématiques), la liste des différents attributs descriptifs qui participeront de la définition des épisodes de cette thématique. Pour chaque attribut, l'expert fournit un nom et son type. Pour ce dernier, le modélisateur dispose d'une liste de types de données simples qui est un sous-ensemble des types de données définis par XMLSchema<sup>2</sup>. Le modélisateur pourra également recourir à une ontologie ou à un vocabulaire contrôlé. Ce sera notamment le cas pour les attributs géographiques (pour nommer une ville de résidence par exemple, on utilisera des ontologies de référence), et ainsi bénéficier implicitement et automatiquement de la connaissance de la structuration administrative hiérarchique d'organisation du territoire lors des analyses. C'est également pour caractériser certaines informations thématiques pour lesquelles il existe un vocabulaire ou une ontologie de domaine pertinente. Par exemple, pour décrire la situation professionnelle d'un individu on peut utiliser le vocabulaire SKOS décrivant les Professions et Catégories Socioprofessionnelles<sup>3</sup> avec plusieurs niveaux de précision. Nous expliquons dans la sous-section suivante (sous-section 3.3 de ce chapitre) comment ces vocabulaires sont intégrés à l'ontologie. Pour d'autres informations, il peut ne pas exister de vocabulaire adapté. Dans ce cas ; il est possible de créer un vocabulaire *ad hoc* afin qu'il soit utilisé dans l'ontologie. Pour chaque thématique, le modélisateur doit également fournir une typologie d'événements pertinents (par exemple, pour la thématique résidentielle, on aura certainement affaire à des déménagements, des achats de biens, des travaux, etc.). Ces typologies sont également fournies sous la forme d'ontologies ou de vocabulaires contrôlés.

Les étapes 2bis et 3bis symbolisent que l'expert peut réappliquer le processus de paramétrage et de génération de l'ontologie autant de fois que nécessaire (*i.e.* autant qu'il y a de trajectoires thématiques à créer pour les besoins de l'étude). Dans la Figure 6.2, on crée une deuxième trajectoire thématique (ici professionnelle) qui est automatiquement liée à la partie de l'ontologie

2. <http://www.w3.org/2001/XMLSchema>

3. <https://www.insee.fr/fr/metadonnees/definition/c1493>

précédemment créée.

Au final, l'ontologie produite par le module est un modèle de trajectoires de vie i) observables selon différents points de vue (*i.e.* les différentes trajectoires thématiques définies), et ii) incluant une dimension explicative à travers la notion de facteurs explicatifs.

À l'issue du processus de modélisation décrit précédemment, l'ontologie est stockée dans un *Triple Store*. Le module *Ontology Manager* prend également en charge la configuration du code des API des autres modules pour les rendre compatibles avec l'ontologie de trajectoire de vie créée (voir Figure 6.1). Ainsi, les différents modules sont proposés dans une version adaptée permettant respectivement l'acquisition des données et leur exploitation.

### 3.3 Fonctionnement du module Ontology Manager pour la modélisation

Cette section introduit le fonctionnement du module *Ontology Manager* pour les deux rôles qui lui sont assignés lors de la phase modélisation et de paramétrage : la conception du modèle (de l'ontologie) et la configuration du module de collecte. Pour chacun de ces rôles, nous décrivons les différentes méthodes utilisées et les deux algorithmes qui assurent l'application du patron de conception à partir des paramètres fournis, ainsi que l'intégration des ontologies de domaine et des vocabulaires contrôlés dans l'ontologie de trajectoire de vie finale.

#### 3.3.1 La conception du modèle

La création du modèle (*i.e.* l'ontologie de trajectoires de vie correspondant aux besoins applicatifs) repose sur deux méthodes que nous décrivons ci-dessous : `CreateTrajectoryModel()` et `AddTrajectory()`. Ces méthodes utilisent le patron de conception permettant de modéliser une trajectoire thématique géographique ou agéographique.

**Modéliser une trajectoire thématique** Nous présentons ici l'algorithme qui permet d'appliquer le *design pattern* proposé pour créer la première trajectoire thématique d'un modèle de trajectoire de vie.

L'algorithme, implémenté dans la méthode `CreateTrajectoryModel()`, crée un modèle de trajectoire pour un point de vue donné selon sa nature (agéographique ou géographique). Cet algorithme intervient donc après l'étape 2 présentée sur la figure 6.2.

Le booléen `isGeographical` vaut `true` si la trajectoire est géographique, faux sinon. La chaîne de caractères `viewStamp` permet de décrire le point de vue et est utilisée pour "tamponner" la trajectoire, reprenant ainsi l'approche proposée par (Parent et al, 2008). Par exemple, nous utilisons la chaîne de caractères *residential* pour estampiller les classes et les relations de la trajectoire géographique résidentielle.

La `HashTable EpiAttributes` est une `Hashtable <<String, Model>>` dont chaque tuple correspond à un des attributs descriptifs de la trajectoire thématique, qui ont été fournis par l'expert à l'étape 2. La chaîne de caractères `AttName` caractérise le nom de chacun de ces attributs. La variable `attValues`, qui est de type `Model` (un type de variable de l'API Jena) permet de caractériser pour chaque attribut, soit son type de données (chaîne de caractères, entier), soit le vocabulaire contrôlé associé qui permet de caractériser les différentes valeurs de cet attribut. Par exemple, pour le point de vue résidentiel, des attributs pertinents sont `HousingType` (dont la valeur est issue d'un vocabulaire contrôlé) et `NumberOfRooms` (un entier).

```

Lexique
1 Model trajectoryPattern ;
2 Boolean isGeographical ;
3 String viewStamp ;
4 HashTable<<String AttName, Model AttValues>> EpiAttributes ;

Algorithme
5 Initialize(foaf :Person);
6 CreateRelevantClassesAndRelations(isGeographical, viewStamp) ;
7 Connect(foaf :Person, Trajectory, viewStamp) ;
8 CharacterizeEpisodes(EpiAttributes);

```

**Algorithm 1:** Création d'une ontologie de trajectoire thématique

La méthode `Initialize(foaf:Person)` (ligne 5) crée le préfixe *foaf* dans l'ontologie et initialise la classe *foaf:Person*.

La méthode `CreateRelevantClassesAndRelations(isGeographical, viewStamp)` (ligne 6) crée les classes et les relations en utilisant la bonne version du *pattern* selon la valeur du booléen `isGeographical`. Si la valeur du booléen est `false`, les classes créées sont *Trajectory*, *Episode* et *Event* et les relations créées sont *hasTrajectory*, *hasEpisode*, *hasEvent*, *endWith*, *startWith*. Si le booléen `isGeographical` vaut `vrai`, les classes *GeographicalTrajectory* et *GeographicalEpisode* sont créées

à la place de la version agéographique. Dans les deux cas, chaque classe et relation du patron est renommée à sa création en préfixant son nom avec la chaîne de caractères `viewStamp`. Par exemple, si la chaîne de caractères est "Residential", cela entraîne la création de la classe *ResidentialSpatialEpisode* ou encore de la relation *hasResidentialEpisode*.

La méthode `Connect(foaf:Person, Trajectory, viewStamp)` (ligne 7) crée une relation entre la classe *foaf:Person* et la classe *Trajectory* venant d'être préfixée, par exemple *ResidentialSpatialTrajectory*.

La méthode `CharacterizeEpisodes(EpiAttributes)` (ligne 8) crée pour chaque paire de la Hashtable *EpiAttributes* le prédicat pertinent, dont le domaine est la classe *Episode* (ou *GeographicalEpisode*) précédemment préfixée. Le prédicat créé par cette méthode est soit une propriété de données (*DataProperty*), soit une propriété d'objet (*ObjectProperty*), selon que la valeur de l'attribut est décrite par un type de donnée ou par un vocabulaire contrôlé. Dans le premier cas, le co-domaine de la propriété de donnée est un type de donnée (celui choisi pour l'attribut en question lors du paramétrage). Dans le second cas, le co-domaine de la propriété d'objet est une classe du vocabulaire dont les instances sont les concepts décrits dans ce vocabulaire. Cette méthode de manipulation conjointe de SKOS et de OWL est conforme aux recommandations du W3C<sup>4</sup>. Prenons, par exemple, le cas des deux attributs du point de vue résidentiel évoqué plus haut. Concernant le premier attribut, la propriété de données `NumberOfRoom` a pour co-domaine un entier (*Integer*). Concernant le second attribut, la propriété d'objet `HousingType` a pour co-domaine *HousingTypeConcept*. *HousingTypeConcept* est une sous classe de *skos:Concept* qui caractérise, elle, les différents concepts dans un vocabulaire SKOS, et a pour instances l'ensemble des concepts qui décrivent un type de logement dans le vocabulaire (studio, appartement...).

**Modéliser de multiples trajectoires thématiques** Pour construire un modèle plus complexe qui intègre les multiples points de vue d'une trajectoire de vie, nous proposons l'algorithme 2 qui sera utilisé autant de fois qu'il faut ajouter une thématique au modèle. Il fonctionne de la même manière que l'algorithme 1 sauf qu'il ne crée pas une nouvelle classe *foaf:Person*, mais associe la nouvelle trajectoire avec la classe *foaf:Person* existante. Cet algorithme utilise donc une variable supplémentaire, une chaîne de caractères `PersonUri` qui correspond à l'identifiant *Uniform Resource Identifier* (URI) de la classe *foaf:Person* précédemment créée. Notons ici qu'il s'agit bien de l'URI

---

4. <https://www.w3.org/2006/07/SWD/SKOS/skos-and-owl/master.html>

permettant de référencer la classe décrivant les individus dans l'ontologie, et non d'une instance de celle-ci.

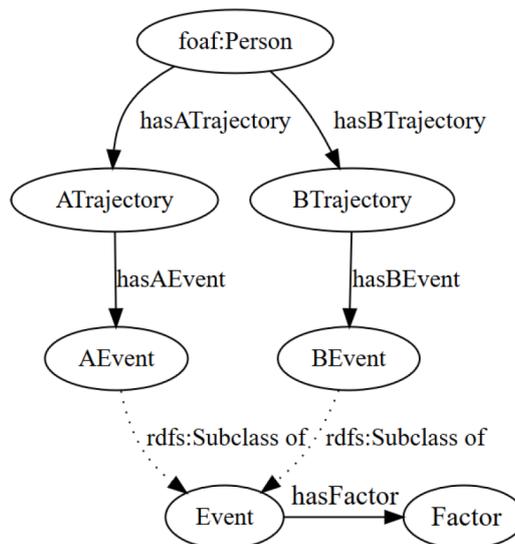
```

Lexique
1 Model lifeTrajectory ;
2 Boolean isGeographical ;
3 String viewStamp ;
4 HashTable<<String AttName, String AttValues>> EpiAttributes ;
5 String PersonURI ;

Algorithme
6 CreateRelevantClassesAndRelations(isGeographical, viewStamp) ;
7 Connect(PersonURI, Trajectory, viewStamp) ;
8 CharacterizeEpisode(EpiAttributes) ;

```

**Algorithm 2:** Ajout d'une trajectoire thématique à l'ontologie de trajectoire de vie



**Figure 6.3** – Représentation de l'association de deux trajectoires thématiques A et B

La figure 6.3 illustre comment deux trajectoires thématiques respectivement tamponnées A et B sont finalement liées dans le même modèle correspondant à un modèle (une ontologie) de trajectoire de vie. La propriété *hasFactor* qui associe un événement (*Event*) à un facteur explicatif (*Factor*)

peut ensuite être utilisée pour chaque classe Event préfixée, comme nous le décrivons dans la section 3 du chapitre V.

À ce stade, après avoir utilisé le module *Ontology Manager* et les méthodes proposées autant de fois que nécessaire, l'expert dispose d'un modèle de trajectoire complet, qui prend la forme d'une ontologie. Afin d'assurer le peuplement de celle-ci, nous définissons les requêtes d'insertion adéquates. Cela est fait grâce à la configuration du module de collecte.

### 3.3.2 Configuration du module de collecte

La configuration du module de collecte pour un cas d'application dédié, c'est-à-dire une ontologie créée à l'étape précédente, est assurée par le module *Ontology Manager*. Elle repose sur des patrons de conception de requêtes. La figure 6.4 présente les différents types de requêtes présents dans le module de collecte de notre environnement. Dans le module de collecte, (*Collect Manager*), deux types de requête SPARQL sont présents : les requêtes *génériques* et les requêtes *dédiées*.

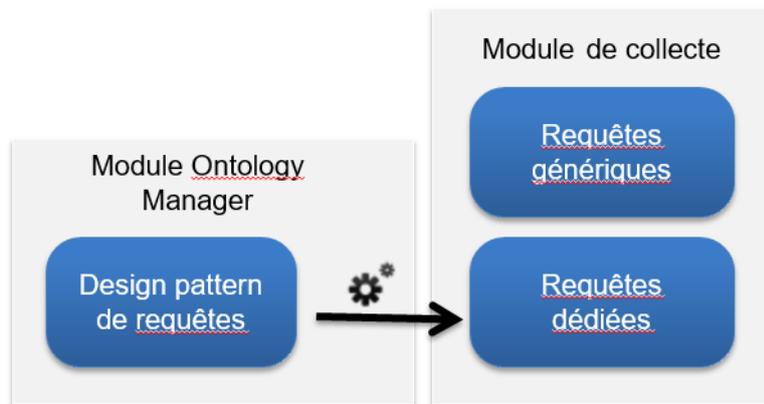


Figure 6.4 – Requêtes génériques et requêtes dédiées

Les requêtes génériques sont des requêtes valables pour tous les modèles pouvant être issus du patron de conception d'ontologie (et donc générées par les algorithmes 1 et 2). Les requêtes génériques ne nécessitent donc pas de configuration. Il s'agit pour la collecte, uniquement des requêtes permettant d'insérer les facteurs explicatifs dans le *Triple Store*. En effet, le modèle de facteur explicatif ne nécessite pas d'être étendu pour un cas d'application particulier et peut-être utilisé directement en association avec le modèle de trajectoire de vie dédié. Nous montrons ces requêtes et la façon dont elles sont utilisées par notre API dans la partie consacrée à l'acquisition des données (voir section 4 de ce chapitre).

Les requêtes dédiées sont les requêtes valables pour un cas d'application déterminé. Ce sont ces requêtes qui doivent être configurées par l'*Ontology Manager* pour pouvoir supporter le modèle de trajectoire de vie créé. Nous avons donc conçu des patrons de requêtes, présents dans le module *Ontology Manager*, qui sont étendus par des algorithmes pour créer les requêtes dédiées, à partir des mêmes paramètres que ceux fournis lors de la création de l'ontologie. Nous utilisons ici la même technique de "tamponnage" (*stamping*) que précédemment [Vangenot et al., 2002]. Pour illustrer ce fonctionnement, nous nous intéressons ci-dessous à la requête permettant l'insertion d'un épisode d'une trajectoire thématique. Nous avons choisi pour notre ontologie le préfixe *lto* (Life Trajectory Ontology).

```

INSERT DATA
{
  L'épisode est associé à sa trajectoire
      <URI-Trajectory> lto:hasEpisode <URI-Episode> .
  L'épisode a un type
      <URI-Episode> a lto:Episode ;
  L'épisode a un événement de début
      lto:startWithEvent <URI-Event1> ;
  L'épisode a un événement de fin
      lto:endWithEvent <URI-Event2> ;
  ainsi que une période de validité
      lto:validityPeriod <Period> .
}

```

**Requête 1** : patron de conception de requête d'insertion d'un épisode

Le patron de conception de requête (Requête 1) pour l'insertion d'un épisode d'une trajectoire thématique comprend uniquement les propriétés déjà définies dans le *design pattern* d'ontologie et non (encore) tamponnées. Il s'agit des propriétés permettant d'associer un épisode à sa trajectoire, de lui donner son type, et de le relier à son événement de début, à son événement de fin ainsi que à sa période de validité. Les URIs (*Unique Resource Identifier*) identifient de manière unique les ressources dans le *Triple Store*. Dans notre approche, nous utilisons un motif permettant de générer des URIs uniques pour les différents types de ressources en respectant les recommandations du W3C<sup>5</sup>. L'URI de l'épisode est donc généré au moment de son insertion. Les URIs qui

5. <https://www.w3.org/TR/cooluris>

identifient les autres instances associées à l'épisode doivent être connues par le développeur. On peut noter que plusieurs instances peuvent être insérées en même temps dans le *Triple Store*, par exemple un épisode et les événements auxquels il se rapporte.

Le requête 2 est générée à partir du *design pattern* précédent grâce un algorithme implémenté dans le module *Ontology Manager* et aux informations passées en paramètres par l'expert. Elle permet l'insertion d'un épisode d'une trajectoire thématique. La trajectoire thématique concernée est une trajectoire agéographique et est caractérisée par la chaîne de caractères *viewstamp*. Dans l'exemple suivant, le texte "viewstamp" est utilisé pour dénoter toute chaîne de caractères choisie pour une thématique. Il pourrait ainsi être remplacé par la valeur "Residential". Dans cet exemple, les épisodes de la trajectoire thématique *viewstamp* ont deux propriétés, qui correspondent à des paramètres fournis par l'expert lors de la phase de modélisation : la propriété d'objet P1, dont le co-domaine est caractérisé par la classe A d'un vocabulaire contrôlé, ainsi que la propriété de données P2 et dont le co-domaine est un entier.

```

INSERT DATA
{
    <URI-Trajectory> a lto:viewstampTrajectory ;
    lto:hasViewstampEpisode <URI-Episode> .
    <URI-Episode> a lto:viewstampEpisode .
    lto:startWithViewstampEvent <URI-Event1> ;
    lto:endWithViewstampEvent <URI-Event2> ;
    lto:validityPeriod <Period> ;

    lto:P1 lto:A .
    lto:P2 xsd:Integer .
}
    
```

**Requête 2** : requête dédiée d'insertion d'un épisode

Les autres patrons de conception de requêtes permettent de générer, selon une approche similaire, les requêtes dédiées pour, respectivement, l'insertion d'une trajectoire, celle d'un épisode géographique, et celle d'un événement.

Les requêtes dédiées sont ensuite utilisées pour l'insertion des données par le développeur utilisant l'API.

### 3.4 Conclusion

Notre approche de modélisation permet la création d'une ontologie adaptée à un cas d'étude comprenant de multiples trajectoires thématiques qui sont autant de points de vue utiles pour ce cas. Nous avons présenté cette approche ainsi que l'outillage permettant la génération de cette ontologie à partir des patrons de conception présentés dans le chapitre V. Une fois l'ontologie créée à l'aide des paramètres fournis par l'expert, le module *OntologyManager* configure le module de collecte de l'architecture, en exploitant ces mêmes paramètres. À la suite de cette étape de modélisation, l'expert pourra procéder à l'acquisition de données pertinentes pour son cas d'application.

## 4 Acquisition de données

Lors de la phase d'acquisition des données, nos APIs proposent différentes méthodes qui sont utiles i) pour faciliter la construction d'une interface de collecte ii) pour l'insertion des données collectées dans le *Triple Store*.

**Faciliter la construction d'une interface de collecte** Pour notre approche nous avons choisi de nous concentrer sur la collecte de données *auprès des individus* comme moyen d'acquisition des données. Ce choix requiert selon nous la construction d'une interface permettant de collecter des données dans un format conforme à l'ontologie. Dans le but de faciliter la construction d'une interface de collecte de données pour un cas d'application, l'API *Ontology Manager* fournit plusieurs méthodes. Ces différentes méthodes sont utilisées dans le chapitre dédié au cas d'application de l'étude des trajectoires résidentielles (section 3 chapitre VII) pour construire une interface de collecte de trajectoires de vie. Nous les présentons successivement ci-après.

```
getViewstamps() : List<<String>> viewstamps
```

Cette méthode renvoie la liste *viewstamps* des chaînes de caractères utilisées pour tamponner chacune des trajectoires thématiques de l'ontologie. Cela permet donc au développeur utilisant l'API *Ontology Manager* de connaître le nombre de trajectoires thématiques de l'ontologie de trajectoire de vie, ainsi que le nom retenu pour chacune d'entre elles. Ainsi, on peut savoir qu'une ontologie de trajectoire de vie conforme à la figure 5.1 (page 71) est constituée de 4 thématiques (résidentielle, professionnelle, familiale et loisirs).

```
getAttributes(String viewstamp) :  
List<<Resource>> attributes
```

Cette méthode renvoie une liste *attributes* de ressources dont chacune correspond à un attribut dans l'ontologie de trajectoire thématique dont le nom correspond à la chaîne de caractères *viewstamp* passée en paramètre. Cela permet, par exemple, toujours pour une ontologie conforme à la figure 5.1, d'accéder aux ressources décrivant les attributs de la trajectoire résidentielle : *Loyer/prêt*, *Statut résidentiel*, *Type de logement* et *Ville de résidence*. Chacune de ces ressources peut ensuite être utilisée pour paramétrer les méthodes suivantes.

```
getAttributeLabel(Resource attribute) : String at-  
tributeLabel
```

Cette méthode renvoie une chaîne de caractères *attributeLabel* qui correspond au label de l'attribut identifié par la ressource *attribute* passée en paramètre. Par exemple, la chaîne de caractère *Statut résidentiel*.

Les méthodes suivantes renvoient des variables de type *Model*. Ce type de données de l'API Jena, permet de caractériser une ontologie ou un vocabulaire contrôlé.

```
getAttributeValues(Resource attribute) : Model at-  
tributeValues
```

Cette méthode renvoie une variable de type *Model* qui correspond à l'ontologie, au vocabulaire contrôlé ou au type de données décrivant les différentes valeurs possibles pour l'attribut identifié par la ressource passée en paramètre. Par exemple, le vocabulaire contrôlé décrivant l'ensemble des *statuts résidentiels*.

```
getEventTypeValues(String viewstamp) : Model  
eventTypeValues
```

Cette méthode renvoie une variable de type *Model* qui correspond au vocabulaire contrôlé qui décrit les différents types d'événements qui peuvent être représentés pour la trajectoire thématique dont le stamp *viewstamp* est passé en paramètre.

Nous montrons dans la section 3 du chapitre VII comment ces méthodes peuvent être utilisées pour construire une interface de collecte pour le cas d'application des trajectoires résidentielles. Une fois ces méthodes utilisées pour construire l'interface de collecte, les méthodes présentées dans le paragraphe suivant peuvent être utilisées par un développeur pour insérer les

données collectées dans une base de données utilisant les technologies du Web sémantique.

**Faciliter l'insertion des données dans le Triple Store** L'API *Collect Manager* propose des méthodes permettant l'insertion des données dans un *Triple Store*. Comme nous l'avons présenté dans la partie précédente, ces requêtes sont soit génériques, soit dédiées.

La requête 2 est une requête générique d'insertion d'un facteur explicatif de type *Factor*. Les variables de type *xsd:Integer* et *xsd:String* sont fournies par l'individu enquêté avant l'insertion du facteur explicatif. Celui-ci fournit également l'événement expliquant de ce facteur explicatif (ici représenté par son URI,  $\langle \text{URI-Event2} \rangle$ ).

```

INSERT DATA
{
  <URI-Factor> rdfs:type lto:Factor .
  <URI-Event1> lto:hasFactor <URI-Factor> .
  <URI-Factor> lto:hasExplanation xsd:String .
  <URI-Factor> lto:weight xsd:Integer .
  <URI-Factor> lto:hasExplainingEvent <URI-Event2> .
}

```

**Requête 2** : requête d'insertion d'un facteur explicatif de type *Factor*

Des requêtes semblables existent pour tous les types de facteurs explicatifs, respectivement *lto:TrajectoryEventFactor*, *lto:InternalEventFactor* et *lto:NetworkEventFactor*. Dans le cas des facteurs de type *lto:TrajectoryStateFactor*, ceux qui font référence non pas à un événement mais à *statement* (une situation dans la trajectoire), le mécanisme d'insertion est un peu différent.

Le *statement* concerné est d'abord réifié, c'est-à-dire qu'il est transformé de manière à pouvoir être désigné par un seul URI (désigné dans la requête 3 par  $\langle \text{URI-statement} \rangle$ ). Ainsi, un triplet représentant un *statement* de la forme  $\langle \text{episode} \rangle \langle \text{property} \rangle \langle \text{value} \rangle$  dans le *triple store*, sera modifié de la manière suivante :

```

<URI-Statement> rdfs:type rdf:Statement .
<URI-Statement> rdf:subject <episode> .
<URI-Statement> rdf:predicate <property> .
<URI-Statement> rdf:object <value> .

```

Ce *statement* pourra ainsi être désigné grâce à l'URI `<URI-Statement>`. Une fois le *statement* réifié, la requête suivante peut être utilisée pour insérer un facteur explicatif.

```
INSERT DATA
{
  <URI-Factor> rdfs:type lto:TrajectoryStateFactor .
  <URI-Event> lto:hasFactor <URI-Factor> .
  <URI-Factor> lto:hasExplanation xsd:String .
  <URI-Factor> lto:weight xsd:Integer .
  <URI-Factor> lto:hasExplainingStatement <URI-Statement> .
}
```

**Requête 3** : requête d'insertion d'un facteur explicatif de type  
TrajectoryStateFactor

L'acquisition des données est une étape importante pour l'étude des trajectoires de vie. Nous avons donc proposé ici de faciliter l'acquisition des données des trajectoires de vie à travers la mise à disposition d'un ensemble de requêtes SPARQL paramétrables pouvant être utilisées pour peupler le modèle.

## 4.1 Conclusion

Concernant l'acquisition des données notre approche est centrée sur la collecte auprès d'individus. Elle permet de faciliter : *i*) l'insertion des données dans le *Triple Store* en fournissant des requêtes dédiées appropriées permettant d'insérer des instances d'individus et de trajectoires de vie ou de parties de trajectoires de vie (épisodes, événements, facteurs explicatifs) ; *ii*) la construction d'une interface de collecte de données conforme au modèle en fournissant à un développeur toutes les informations nécessaires pour élaborer cette interface. Nous proposons également des solutions d'exploitation des données collectées, que nous présentons dans la section suivante.

## 5 Exploitation des données

L'objectif d'exploitation des données dépend fortement du cas d'application. Des fonctionnalités d'exploration et d'analyse des données de trajectoires de vie sont cependant préprogrammées et valables quel que soit le type de

trajectoires de vie considéré. En effet, comme nous le pointions dans l'état de l'art (chapitre IV), les travaux existants proposent peu de solutions génériques pour l'exploitation des données. De nouveaux opérateurs génériques permettant la manipulation d'un ensemble de trajectoires dans une base de données doivent être développés. Les fonctionnalités que nous proposons prennent la forme de fonctions SPARQL pouvant être utilisées dans des requêtes permettant d'obtenir des résultats conformes à des critères spatiaux, temporels et/ou thématiques en exploitant l'ontologie de trajectoire de vie.

Nous donnons dans la section 4 du chapitre VII des exemples de requêtes pour le cas d'application des trajectoires résidentielles. Ces exemples mettent en œuvre les opérateurs présentés dans ce qui suit.

**Étendre le langage SPARQL** L'exploitation des données conformes aux modèles est effectuée en utilisant le langage SPARQL<sup>6</sup>. Dans les requêtes que nous proposons pour notre cas d'application, nous exploitons le modèle selon toutes ses dimensions : temporelles, géographiques et thématiques (voir section 4 du chapitre VII). Concernant la dimension temporelle, les requêtes permettent, par exemple, de connaître la situation d'un (ou plusieurs) individu(s) à un instant  $t$  ou pendant un intervalle de temps  $[t1, t2]$ . Par situation, nous entendons la restitution des épisodes observés au temps  $t$  ou dans l'intervalle  $[t1, t2]$ . Sur le plan géographique, il est possible de savoir quels lieux font partie de la trajectoire des individus. Le modèle permet également de connaître les facteurs explicatifs rapportés par les individus pour éclairer leurs trajectoires. Il est par exemple possible de s'intéresser à un événement précis de la trajectoire et d'en rechercher les causes en parcourant le modèle. Des requêtes alliant dimensions temporelles, géographiques, thématiques, incluant des critères relatifs à des facteurs explicatifs sont également possibles.

À un niveau générique, pour répondre aux limites évoquées dans l'état de l'art (voir chapitre IV), nous proposons des fonctions SPARQL valables pour n'importe quel modèle issu de l'application du patron de conception. Nous présentons ici deux catégories de fonctions, qui permettent *i*) de trier les épisodes sur des critères temporels *ii*) d'exploiter les facteurs explicatifs dans un ensemble de trajectoires tirant ainsi parti d'une des particularités de notre modèle. Pour un cas d'application (voir chapitre VII section 4) il est ainsi possible de faire émerger de la connaissance en utilisant ces fonctions et en exploitant les épisodes et événements de plusieurs trajectoires thématiques d'une trajectoire de vie.

---

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

Le langage SPARQL (voir section 4 du chapitre II) est basé sur la correspondance de motifs entre deux graphes : le graphe de la requête (le motif recherché) et le graphe de données (dans lequel on recherche le motif). Une requête de sélection des données (SELECT<sup>7</sup>) comprend donc un ensemble de triplets de la forme *?sujet ?predicat ?objet* sur la base duquel s'opère le *matching*. Le résultat d'une requête correspond à l'ensemble des triplets contenus dans le *Triple Store* qui correspondent au motif recherché.

Les fonctions que nous proposons ici, appelées *Property Function*<sup>8</sup> dans la librairie JENA<sup>9</sup>, doivent être utilisées comme des *prédicats* dans les requêtes. Une fonction est donc utilisée dans un *triplet* et a pour argument son *sujet* et son *objet*. L'URI de la fonction (dans notre cas de la forme `ltofFn:nomDeLaFonction`) renvoie alors à une fonction dans l'interpréteur de requêtes. Lorsque le *matching* est opéré par celui-ci, c'est donc la fonction qui va juger de la validité du triplet. Si le triplet est valide, il peut alors faire partie des solutions de la requête, sinon, il n'est pas considéré comme une solution. Si le triplet est valide, les arguments sont donc renvoyés par la fonction ; modifiés ou non. Dans notre cas, nous ne modifions pas les arguments, mais nous choisissons de les renvoyer uniquement si ils correspondent à un motif dans le graphe de données (différent pour chaque fonction). Ces fonctions permettent ainsi de simplifier les requêtes SPARQL en faisant correspondre à un simple triplet de la forme *?sujet ltofFn:nomDeLaFonction ?objet*, un motif plus complexe dans le graphe de données présent dans le *Triple Store*. Toutes les fonctions que nous proposons ont pour préfixe *ltofFn*.

Nous les avons implémentées en utilisant la librairie JENA qui propose d'ajouter des fonctions<sup>10</sup> à son interpréteur de requêtes SPARQL, ARQ<sup>11</sup>. Les classes JAVA implémentant les fonctions que nous proposons sont disponibles dans l'annexe C. Celles-ci peuvent également être utilisées en téléchargeant le code source de ces fonctions<sup>12</sup>. Ce code est fourni avec un jeu de données (un ensemble d'instances de trajectoire inspirées de trajectoires de vies réelles), qui permet de tester les fonctions proposées.

**Fonctions temporelles sur les épisodes de trajectoires** Les fonctions suivantes ont pour objectif de situer temporellement deux à deux les épisodes de trajectoires thématiques. Chacune de ces fonctions a pour arguments deux

---

7. <https://www.w3.org/TR/rdf-sparql-query/#select>

8. [https://jena.apache.org/documentation/query/writing\\_propfuncs.html](https://jena.apache.org/documentation/query/writing_propfuncs.html)

9. <https://jena.apache.org/index.html>

10. [https://jena.apache.org/documentation/query/writing\\_propfuncs.html](https://jena.apache.org/documentation/query/writing_propfuncs.html)

11. <https://jena.apache.org/documentation/query/>

12. <https://gricad-gitlab.univ-grenoble-alpes.fr/steamer/thesis/lifeline>

épisodes (l'un est le *sujet* du triplet, l'autre son *objet* lors de l'utilisation de la fonction). Il s'agit de comparer les deux intervalles temporels auxquels correspondent ces épisodes. Les intervalles temporels sont comparés suivant qu'ils correspondent ou non à une relation temporelle. Les trois relations que nous utilisons ici font partie de celles définies par [Allen, 1984]

Nous définissons ici trois fonctions temporelles : `ltofn:before`, `ltofn:after`, ainsi que `ltofn:overlaps`.

**lto:Episode ?episode1 ltofn:before lto:Episode ?episode2 .**

Si l'épisode?episode1 se déroule *avant* l'épisode?episode2, la fonction renvoie ses deux arguments, sinon, elle ne renvoie pas de solution.

Le triplet est donc valide si le motif suivant existe dans le graphe de données :

```
?episode1 lto:validityPeriod ?epi1ValidityPeriod .
?epi1ValidityPeriod time:hasEnd ?epi1DateEnd .
?epi1DateEnd time:inXSDDate ?epi1DateEndXSD .
?episode2 lto:validityPeriod ?epi2ValidityPeriod .
?epi2ValidityPeriod time:hasBeginning ?epi2DateBegin .
?epi2DateBegin time:inXSDDate ?epi2DateBeginXSD .
FILTER (?epi1DateEndXSD <= ?epi2DateBeginXSD)
```

**lto:Episode ?episode1 ltofn:after lto:Episode ?episode2 .**

Si l'épisode?episode1 se déroule *après* l'épisode?episode2, la fonction renvoie ses deux arguments, sinon, elle ne renvoie pas de solution.

Le triplet est donc valide si le motif suivant existe dans le graphe de données :

```
?episode2 lto:isIn ?commune1 .
?commune1 rdfs:label "GRENOBLE"@fr .
?episode1 lto:validityPeriod ?epi1ValidityPeriod .
?epi1ValidityPeriod time:hasEnd ?epi1DateEnd .
?epi1DateEnd time:inXSDDate ?epi1DateEndXSD .
?episode2 lto:validityPeriod ?epi2ValidityPeriod .
?epi2ValidityPeriod time:hasBeginning ?epi2DateBegin .
?epi2DateBegin time:inXSDDate ?epi2DateBeginXSD .
FILTER (?epi2DateEndXSD <= ?epi1DateBeginXSD)
```

**lto:Episode ?episode1 ltofn:overlaps lto:Episode ?episode2 .**

Si l'épisode *?episode1* chevauche l'épisode *?episode2*, la fonction renvoie ses deux arguments, sinon, elle ne renvoie pas de solution.

Le triplet est donc valide si le motif suivant existe dans le graphe de données :

```
?episode1 lto:validityPeriod ?epi1ValidityPeriod .  
?epi1ValidityPeriod time:hasBeginning ?epi1DateBegin .  
?epi1DateBegin time:inXSDDate ?epi1DateBeginXSD .  
?epi1ValidityPeriod time:hasEnd ?epi1DateEnd .  
?epi1DateEnd time:inXSDDate ?epi1DateEndXSD .
```

```
?episode2 lto:validityPeriod ?epi2ValidityPeriod .  
?epi2ValidityPeriod time:hasBeginning ?epi2DateBegin .  
?epi2DateBegin time:inXSDDate ?epi2DateBeginXSD .  
?epi2ValidityPeriod time:hasEnd ?epi2DateEnd .  
?epi2DateEnd time:inXSDDate ?epi2DateEndXSD .
```

```
FILTER (?epi1DateBeginXSD < ?epi2DateEndXSD  
      && ?epi1DateEndXSD < ?epi2DateBeginXSD)
```

Ces fonctions permettent de comparer des épisodes de trajectoires selon des critères temporels. En cela, elles répondent à un manque que nous avons identifié dans l'état l'art en offrant des opérateurs génériques pour manipuler un ensemble de trajectoires. Nous les utilisons pour notre cas d'application dans la section 4 du chapitre VII.

**Fonctions exploitant la notion de facteurs explicatifs** Afin de faciliter la compréhension des fonctions que nous proposons, la figure 6.5 présente un ensemble de trajectoires thématiques et de facteurs explicatifs qui sert d'exemple pour illustrer leur fonctionnement.

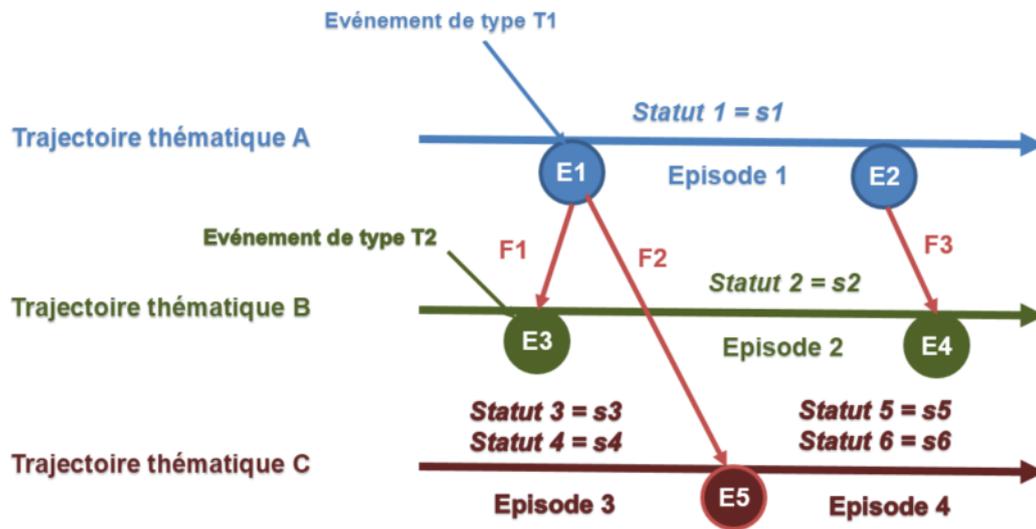


Figure 6.5 – Un exemple de trajectoire de vie

Cette figure présente un exemple de trajectoire de vie composée de trois trajectoires thématiques, A, B et C. La trajectoire thématique A contient deux événements, E1 et E2. L'événement E1 est un événement de type T1. Entre les deux événements, un épisode existe lors duquel le statut  $s1$  est observable. Un statut, conformément à notre modèle, désigne la valeur d'un attribut d'un épisode d'une trajectoire. Par exemple, dans une trajectoire résidentielle, le statut correspondant à l'attribut *Statut résidentiel* peut être *Propriétaire*. La trajectoire thématique B comprend un événement E3, après lequel le statut  $s2$  est observable, et un événement E4. La trajectoire thématique C comprend un événement E5. Avant cet événement, un épisode existe pendant lequel les statuts  $s3$  et  $s4$  sont observables. Après cet événement, un autre épisode est représenté lors duquel les statuts  $s5$  et  $s6$  sont observables.

La figure présente également trois facteurs explicatifs F1, F2 et F3. Ainsi, l'événement E1 a pour facteur explicatif F1, pour lequel E3 est l'événement explicatif. L'événement E1 a également pour facteur explicatif F2, pour lequel E5 est l'événement explicatif. L'événement E2 a pour facteur explicatif F3, pour lequel l'événement explicatif est E4. Ici les événements E1 et E2 sont donc des événements expliqués, tandis que E3, E4 et E5 sont des événements explicatifs.

On peut noter ici que l'événement E5 qui explique l'événement E1 s'est produit après lui. Ceci signifie que notre modèle est capable de représenter que l'individu a anticipé l'événement E1. Un exemple simple est celui d'un événement *naissance*, qui aurait été anticipé par l'individu et aurait contribué à provoquer un événement *déménagement* avant que la naissance ait lieu.

Notre modèle est capable de caractériser ces cas d'anticipation passée, quand ils sont rapportés *a posteriori* par l'individu.

Les fonctions suivantes permettent d'exploiter les facteurs explicatifs contenus dans un ensemble de trajectoires ayant été préalablement collectées. Pour chacune de ces fonctions, nous donnons sa signature, sa définition, et le motif qu'elle vérifie dans le graphe de données.

**Filtrer un facteur explicatif et un statut** Les deux fonctions suivantes permettent de filtrer un facteur explicatif et un statut associé.

La première, `ltofn:isFactorEnterIn`, teste si un facteur explicatif marque le début d'un épisode pour lequel ce statut est valable. Cela permet, par exemple, d'obtenir les facteurs explicatifs des événements qui marquent l'accès à la propriété (le début du statut "Propriétaire") dans les trajectoires considérées.

**`lto:factor factor ltofn:isFactorEnterIn OWL:Thing status`**

Si `factor` est le facteur explicatif d'un événement marquant le début d'un épisode pour lequel un individu a un statut `status`, la fonction retourne ses deux arguments, sinon, elle ne retourne pas de solution.

Le triplet `?factor ltofn:isFactorEnterIn ?status` est donc valide si le motif suivant existe dans le graphe de données :

```
?episode ?statusProperty ?status .  
?episode lto:startWith ?event .  
?event lto:hasFactor ?factor .
```

Pour les données présentées sur la figure 6.5, `F1 ltofn:isFactorEnterIn s1` et `F2 ltofn:isFactorEnterIn s1` sont des triplets valides.

La fonction peut donc être utilisée dans une requête SPARQL de la manière suivante :

```
SELECT ?factor WHERE {  
  ?factor a lto:explanatoryFactor .  
  ?factor ltofn:isFactorEnterIn s1 .  
}
```

Pour les données présentées sur la figure la requête renvoie les facteurs F1 et F2.

Une seconde fonction permet d'obtenir les facteurs explicatifs qui marquent *la fin* des épisodes pour lesquels un statut est valide.

**lto:factor factor ltofn:isFactorOutFrom OWL:Thing status**

Si `factor` est le facteur explicatif d'un événement marquant la fin d'un épisode pour lequel un individu a un statut `status`, la fonction retourne ses deux arguments, sinon, elle ne retourne pas de solution.

Le triplet `?factor ltofn:isFactorOutFrom ?status` est donc valide si le motif suivant existe dans le graphe de données :

```
?episode ?statusProperty ?status .
?episode lto:endWith ?event .
?event lto:hasFactor ?factor .
```

Toujours dans les cas de l'exemple de la figure 6.5, F3 `ltofn:isFactorOutFrom s1` est un triplet valide.

La fonction peut être utilisée dans une requête SPARQL de la manière suivante :

```
SELECT ?factor WHERE {
  ?factor a lto:ExplanatoryFactor .
  ?factor ltofn:isFactorOutFrom s1 .
}
```

Ainsi, dans le cas de la trajectoire de vie de la figure 6.5, cette requête ne renvoie que le facteur F3.

**Filtrer un facteur explicatif et un type d'événement** Les fonctions suivantes permettent de filtrer un facteur explicatif et un type d'événement.

Ils permettent respectivement de tester un facteur explicatif avec un événement explicatif ou un événement expliqué. Toujours sur l'exemple de la figure 6.5, l'événement E1 est expliqué en partie par le facteur explicatif F1, pour lequel E3 est un événement explicatif.

**lto:factor factor ltofn:isFactorOfExplainedEventType**  
**skos:Concept eventType**

Si le facteur explicatif `factor` est expliqué par un événement de type `eventType`, la fonction renvoie ses deux arguments. Sinon, elle ne renvoie pas de solution.

Le triplet `?factor ltofn:isFactorOfExplainedEventType ?eventType` est donc valide si le motif suivant existe dans le graphe de données :

```
?event lto:hasFactor ?factor .  
?event lto:hasType ?eventType .
```

Ainsi, toujours en suivant les données présentées sur la figure 6.5, le triplet `F1 ltofn:isFactorOfExplainedEventType T1` est valide.

La fonction peut être utilisée dans une requête SPARQL de la manière suivante :

```
SELECT ?factor WHERE {  
  ?factor a lto:ExplanatoryFactor .  
  ?factor ltofn:isFactorOfExplainedEventType T1 .  
}
```

Dans ce cas, la requête renvoie tous les facteurs explicatifs expliqués par un événement de type `T1`. Le facteur `F1` est donc le seul facteur explicatif qui correspond à cette requête.

**lto:factor factor ltofn:isFactorOfExplainingEventType**  
**skos:Concept eventType**

Si le facteur explicatif `factor` explique un événement de type `eventType`, la fonction renvoie ses deux arguments. Sinon, elle ne renvoie pas de solution.

Le triplet `?factor ltofn:isFactorOfExplainingEventType ?eventType` est donc valide si le motif suivant existe dans le graphe de données :

```
?factor lto:hasExplainingEvent ?event .  
?event lto:hasType ?eventType .
```

Ainsi, toujours en suivant les données présentées sur la figure 6.5, le triplet `F1 IsFactorOfExplainingEventType T2` est valide.

La fonction peut être utilisée dans une requête SPARQL de la manière suivante :

```
SELECT ?factor WHERE {
  ?factor a lto:explanatoryFactor .
  ?factor ltofn:isFactorOfExplainingEventType T2 .
}
```

Dans ce cas, la requête renvoie tous les facteurs explicatifs qui expliquent un événement de type T2. Dans la trajectoire de vie exemple de la figure, il s'agit du facteur F1.

#### **Filtrer un facteur explicatif et un type de trajectoire thématique**

Nous avons vu qu'un facteur explicatif est lié à deux événements. Chacun de ces événements appartient à une trajectoire thématique. On peut donc dire qu'un facteur explicatif est lié à deux trajectoires thématiques : la trajectoire expliquée (dont il explique un événement) et la trajectoire explicative (celle à laquelle appartient l'événement explicatif). Par exemple, sur la figure 6.5, le facteur explicatif F1 explique un événement de la trajectoire thématique A et a pour événement explicatif un événement de la trajectoire thématique B. On peut noter que ces deux trajectoires thématiques sont différentes dans les exemples de la figure 6.5, mais ce n'est pas forcément le cas (un événement peut expliquer un autre événement appartenant à la même thématique).

Les deux fonctions que nous proposons ici permettent ainsi de filtrer les facteurs explicatifs à partir d'une trajectoire thématique *expliquée*, ou *explicative* passée en paramètre, respectivement pour chacun de ces deux cas.

Nous verrons dans notre cas d'application (voir section 4 du chapitre VII) que, combinées, ces fonctions peuvent être utilisées pour formuler une requête telle que : "quels sont les facteurs explicatifs d'événements de la trajectoire résidentielle qui sont expliqués par des événements de la trajectoire familiale?".

#### **`lto:factor factor ltofn:isFactorOfExplainedTrajType` **OWL:Class trajectoryType)****

Si le facteur explicatif `factor` est expliqué par une trajectoire thématique de type `trajectoryType`, la fonction renvoie ses arguments. Sinon, elle ne renvoie pas de solution. Le paramètre `OWL:Class`

`trajectoryType` désigne une classe *tamponnée* dont une instance est une trajectoire thématique, par exemple *lto :FamilialTrajectory*.

Le triplet `?factor ltofn:isFactorOfExplainedTrajType ?trajectoryType` est donc valide si le motif suivant existe dans le graphe de données :

```
?event lto:hasFactor ?factor .  
?trajectory lto:hasEvent ?event .  
?trajectory rdfs:type ?trajectoryType .
```

**lto:Factor factor ltofn:isFactorOfExplainingTrajType**  
**OWL:Class trajectoryType**

Si `factor` explique une trajectoire thématique de type `trajectoryType`, la fonction renvoie ses arguments. Sinon, elle ne renvoie pas de solution.

Le triplet `?factor ltofn:isFactorOfExplainingTrajType ?trajectoryType` est donc valide si le motif suivant existe dans le graphe de données :

```
?factor lto:hasExplainingEvent ?event .  
?trajectory lto:hasEvent ?event .  
?trajectory rdfs:type ?trajectoryType .
```

**Filtrer un épisode et un facteur explicatif** Les fonctions présentées ci-dessous ont pour arguments un facteur explicatif et un épisode.

Dans le cas des deux premières fonctions, il s'agit de comparer un facteur explicatif avec un épisode de la même trajectoire thématique que l'événement expliqué par le facteur. Cette fonction est utilisée pour obtenir une meilleure compréhension du facteur explicatif en donnant l'accès aux épisodes auxquels il se rapporte. Par exemple, dans le cas du facteur explicatif d'un déménagement (événement *résidentiel*), il est possible d'utiliser ces fonctions pour connaître la situation *résidentielle* de l'individu *avant*, ou *après*, ce déménagement.

**lto:Episode episode ltofn:episodeBeforeFactor****lto:Factor factor**

Si l'épisode `episode` correspond à l'épisode terminé par l'événement expliqué par le facteur explicatif `factor`, la fonction renvoie ses arguments. Sinon, elle ne renvoie pas de solution.

Le triplet `?episode ltofn:episodeBeforeFactor ?factor` est valide si le motif suivant existe dans le graphe de données :

```
?episode lto:endWith ?event .
?event lto:hasFactor ?factor .
```

Par exemple, en se basant sur les données présentées sur la figure 6.5, Episode 1 `ltofn:episodeBeforeFactor F3` est valide.

**lto:Episode episode ltofn:episodeAfterFactor****lto:Factor factor**

Si l'épisode `episode` correspond à l'épisode commencé par l'événement expliqué par le facteur explicatif `factor`, la fonction renvoie ses arguments. Sinon, elle ne renvoie pas de solution.

Le triplet `?episode ltofn:episodeAfterFactor ?factor` est valide si le motif suivant existe dans le graphe de données :

```
?episode lto:startWith ?event .
?event lto:hasFactor ?factor .
```

Par exemple, en se basant sur les données présentées sur la figure 6.5, Episode 1 `ltofn:episodeAfterFactor F2` est valide.

Dans le cas de la troisième fonction manipulant épisodes et facteurs explicatifs, il s'agit de vérifier qu'un épisode d'une autre trajectoire thématique de la même trajectoire de vie est valide au moment de l'événement expliqué par le facteur.

**lto:Episode episode ltofn:episodeAtFactorDate****lto:Factor factor**

Si l'épisode `episode` est valide à la date de l'événement expliqué par le facteur explicatif `factor` et qu'il appartient à la même trajectoire de vie que celui ci, la fonction renvoie ses arguments, sinon elle ne renvoie pas de solution.

Le triplet `?episode ltofn:episodeAtFactorDate ?factor` est valide si le motif suivant existe dans le graphe de données :

```
?trajectory lto:hasEpisode ?episode .
?trajectory lto:hasEvent ?event .
?event lto:hasFactor ?factor .
?event lto:atTime ?eventDate .
?eventDate OWL-Time:inXSDDate ?eventDateXSD .
?episode lto:validityPeriod ?epiValidityPeriod .
?epiValidityPeriod OWL-Time:hasBeginning ?epiDateBegin .
?epiValidityPeriod OWL-Time:hasEnd ?epiDateEnd .
?epiDateBegin OWL-Time:inXSDDate ?epiDateBeginXSD .
?epiDateEnd OWL-Time:inXSDDate ?epiDateEndXSD .

FILTER { ?eventDateXSD >= ?epiDateBeginXSD
         AND ?eventDateXSD <= ?epiDateEndXSD }
```

Par exemple, en se basant sur les données présentées sur la figure 6.5, les triplets `Episode 2 ltofn:episodeAtFactorDate F2` et `Episode 3 ltofn:episodeAtFactorDate F2` sont valides.

Nous avons présenté ici des opérateurs SPARQL qui permettent d'exploiter des trajectoires de vie stockées dans un *Triple Store*. Ceux-ci permettent en particulier de tirer partie de notre modèle de facteurs explicatifs et de les filtrer selon leur type, selon les événements auxquels ils sont liés ou bien encore selon un statut valide pour un épisode. Ces opérateurs sont génériques et pourront être utilisés pour n'importe quel modèle dédié créé en utilisant notre méthodologie. Nous montrons dans la partie suivante comment ceux-ci peuvent être utilisés pour le cas d'application des trajectoires résidentielles.

## 6 Conclusion

Dans ce chapitre nous avons décrit notre méthodologie de modélisation, collecte et exploitation de trajectoires de vie. Notre approche, générique, repose sur une architecture, dont les composants sont paramétrables pour un cas d'application. Lors de la phase de modélisation, elle permet de créer une ontologie multipoints de vue en s'appuyant sur des algorithmes. Elle facilite

également la collecte des données, en fournissant des méthodes pour faciliter la création d'une interface de collecte, ainsi que pour insérer les données dans un Triple Store. Finalement, nous avons également définis des opérateurs génériques utilisable pour exploiter les données collectées. Dans le prochain chapitre, nous appliquons notre méthodologie à l'étude des trajectoires résidentielles.



# Application de la méthodologie à l'étude des trajectoires résidentielles

---

## Sommaire

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>115</b>
<b>2</b>	<b>Application du patron de conception pour la production d'un modèle cible</b> . . . . .	<b>115</b>
2.1	Choix des points de vue pertinents . . . . .	116
2.2	Création de l'ontologie de trajectoire . . . . .	117
<b>3</b>	<b>Collecte de trajectoires de vie</b> . . . . .	<b>121</b>
<b>4</b>	<b>Analyse des trajectoires de vie collectées</b> . . . . .	<b>124</b>
<b>5</b>	<b>Conclusion</b> . . . . .	<b>138</b>

---

## 1 Introduction

L'étude des trajectoires résidentielles est un enjeu important pour les métropoles (voir chapitre 1). Une première validation de notre approche passe par son application dans ce domaine. Assistés de spécialistes du domaine, nous avons successivement mis en œuvre les étapes de modélisation, d'acquisition des données et d'exploitation, que nous décrivons ici.

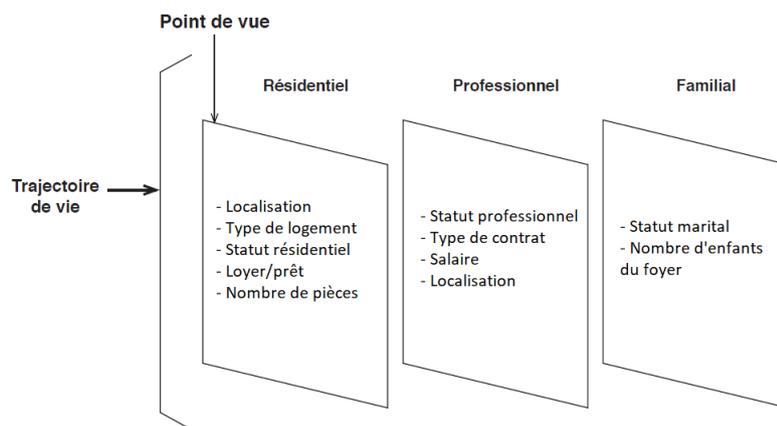
## 2 Application du patron de conception pour la production d'un modèle cible

Dans cette partie, nous utilisons la méthodologie présentée précédemment afin de fournir un modèle de trajectoire de vie adaptée à l'étude des

trajectoires résidentielles. Nous décrivons ici le choix des points de vue pertinents pour ce cas d'application, ainsi que l'ontologie de trajectoire de vie créée lors de cette étape.

## 2.1 Choix des points de vue pertinents

**Choix des points de vue pertinents** Suivant notre méthodologie, la première étape de déterminer l'ensemble des caractéristiques de l'individu évoluant au cours du temps dont l'observation est requise. Chacune doit alors être représentée par un attribut. Pour notre modélisation de trajectoire de vie appliquée à l'étude des trajectoires résidentielles, nous avons sélectionné onze attributs (voir figure 7.1). Ces attributs sont regroupés dans différents points de vue, conformément à la définition du point de vue présenté dans la section 2 de la partie III de l'état de l'art.



**Figure 7.1** – Différents points de vue pertinents pour l'étude du choix résidentiel

La Figure 7.1 présente trois points de vue selon lesquels peut être observée la trajectoire d'une personne. Dans notre cas d'étude, par exemple, les trois points de vue "résidentiel", "familial" et "professionnel" sont essentiels car ce sont des aspects de la vie qu'il est incontournable de prendre en compte pour trouver les facteurs explicatifs de la trajectoire résidentielle.

Le point de vue résidentiel (étiqueté par le stamp *Résidentiel*) contient une dimension géographique, décrite par l'attribut *localisation*. Les autres attributs de ce point de vue sont *Type de logement*, *Nombre de pièces*, *Statut résidentiel* et *Loyer/prêt* qui décrivent respectivement le type et le nombre de pièces de l'habitation de l'individu, ainsi que le statut résidentiel de l'individu et un intervalle de valeurs dans lequel est compris le montant consacré à se loger.

Le point de vue professionnel (étiqueté *Professionnel*) contient également une dimension géographique décrite par un attribut *localisation* qui caractérise la localisation de l'emploi occupé. Les autres attributs de ce point de vue sont *Statut professionnel*, *Type de contrat* et *Salaire* qui décrivent respectivement le type d'emploi, le type de contrat (CDI, CDD, Intérim...) et un intervalle de valeurs dans lequel est compris le salaire de l'individu.

Le point de vue familial (étiqueté *Familial*) ne comprend pas de dimension géographique : la trajectoire familiale est une trajectoire agéographique. Les attributs de ce point de vue sont *Satut marital* et *Nombre d'enfants du foyer* qui caractérisent respectivement le statut marital de l'individu et le nombre d'enfants résidant dans son foyer.

D'autres points de vue peuvent cependant apparaître en fonction des centres d'intérêt de l'individu étudié, tel que les loisirs, qui peuvent en effet, également influencer sur les choix résidentiels.

L'étape suivante de notre méthodologie consiste à choisir les types de données ou les ontologies et vocabulaires contrôlés utilisés pour décrire chacun des attributs choisis. Lors de cette étape, le spécialiste de domaine qui a précédemment identifié ses besoins, doit les confronter aux types de données et aux ontologies et vocabulaires contrôlés existants. Dans le cas où il n'existe aucun vocabulaire contrôlé pertinent pour caractériser un attribut, il peut décider d'en créer un. Nous présentons dans la partie suivante, consacrée à la description de l'ontologie obtenue, les différents types de données, ontologies et vocabulaires contrôlés utilisés ou créés pour cette ontologie.

## **2.2 Création de l'ontologie de trajectoire**

En suivant la méthodologie que nous présentons dans le chapitre VI, le modèle de trajectoire est ensuite obtenu. Chacun des points de vue forme alors une trajectoire thématique et ces trajectoires thématiques sont parties intégrantes de l'ontologie de trajectoire de vie. Comme décrit précédemment, pour chaque attribut, un type de données simple, une ontologie ou un vocabulaire contrôlé a été choisi. Dans le cas du choix d'un vocabulaire contrôlé existant, ce vocabulaire est intégré à l'ontologie de trajectoire de vie en suivant notre approche (partie VI).

Nous décrivons ici chacune des trajectoires thématiques pertinentes pour l'étude de la trajectoire résidentielle. La définition de chaque point de vue thématique de cette ontologie au format OWL (sérialisation Turtle) est disponible dans l'annexe D. L'ensemble de l'ontologie est également disponible en ligne<sup>1</sup>.

1. <http://metropolifeline.imag.fr/ontologies/lto.html>

Les classes non préfixées sur les figures ci-dessous sont celles appartenant à cette ontologie (et pas à une ontologie externe). Le préfixe choisi pour cette ontologie dans sa définition OWL est lto (Life Trajectory Ontology).

### La trajectoire résidentielle

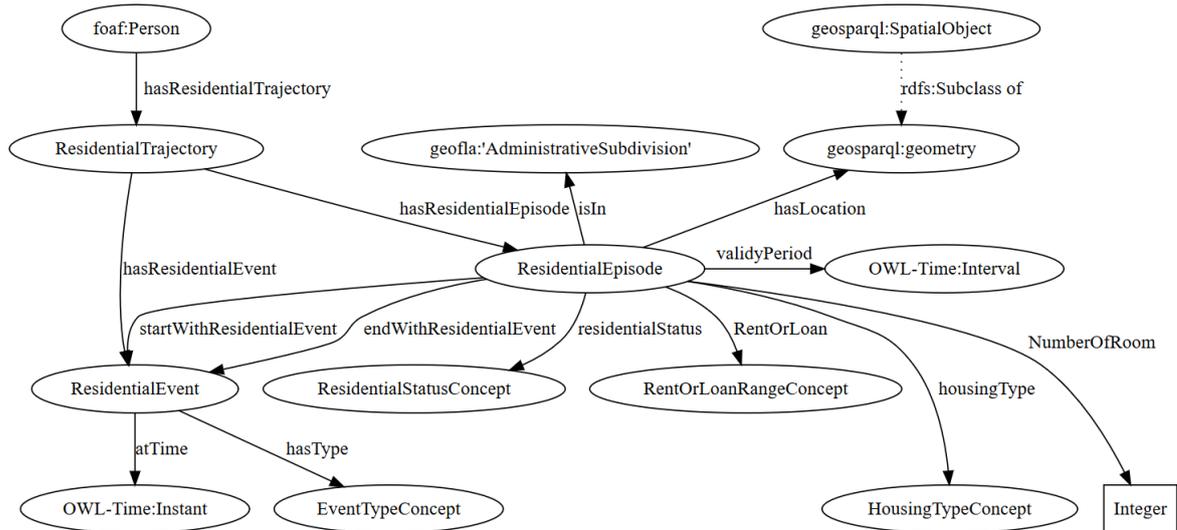


Figure 7.2 – Ontologie de la trajectoire résidentielle

La figure 7.2 représente l'ontologie de trajectoire de vie du point de vue résidentiel. Il s'agit d'une trajectoire géographique. Une trajectoire résidentielle est donc caractérisée par la classe *ResidentialSpatialTrajectory* et un épisode résidentiel par la classe *ResidentialSpatialEpisode*. Ces classes héritent respectivement des classes *Trajectory* et *Episode* (non représentées ici). Un épisode est lié à la classe *ResidentialEvent*, sous-classe de *Event*, par les relations *startWithResidentialEvent* et *endWithResidentialEvent*. Les classes *OWL-Time:Instant* et *OWL-Time:Interval*, deux sous-classes de la classe *OWL-Time:Temporal-Thing* appartenant à l'ontologie OWL-Time caractérisent la dimension temporelle de la trajectoire, respectivement pour les événements et les épisodes. La dimension spatiale des épisodes est modélisée par la classe *geosparql:Geometry* (relation *hasLocation*), ainsi que par la classe *geofla:'AdministrativeSubdivision'* (relation *isIn*). L'ontologie Geofla<sup>2</sup>, une ontologie proposée par l'IGN pour la description des unités administratives françaises est donc utilisée.

Comme nous l'avons vu dans la description de ce point de vue, les différentes caractéristiques thématiques qui nous intéressent ici sont au nombre de

2. <http://data.ign.fr/def/geofla/20140822.htm>

quatre : le type de logement, le nombre de pièces du logement, le montant du loyer ou du prêt et le statut résidentiel de l'individu pendant cet épisode. Le nombre de pièces (propriété *numberOfRooms*) est caractérisé par un type de données simple : un entier (*Integer*). Le type de logement (propriété *housingType*) est caractérisé par la classe *HousingTypeConcept*. Le statut résidentiel (propriété *residentialStatus*) est caractérisé par la classe *ResidentialStatusConcept*. Ces deux classes font chacune partie d'un vocabulaire contrôlé qui a été conçu pour la création de cette ontologie. Ces vocabulaires, qui sont parties intégrantes de l'ontologie, de trajectoire de vie sont disponibles aux sections 1 et 2 de l'annexe E. Nous avons également choisi de décrire des intervalles de coût du logement à l'aide d'un vocabulaire contrôlé présenté dans la section 3 de l'annexe E. Ce choix permet de décrire des intervalles pertinents pour l'analyse ultérieure des données qui seront utilisées par le contributeur lorsqu'il renseignera le montant de son loyer ou de son prêt dans la phase de collecte. L'utilisation d'intervalles permet également de ne pas enregistrer de nouvel épisode pour une variation de coût du logement qui ne fait pas changer celui-ci d'intervalle.

### La trajectoire professionnelle

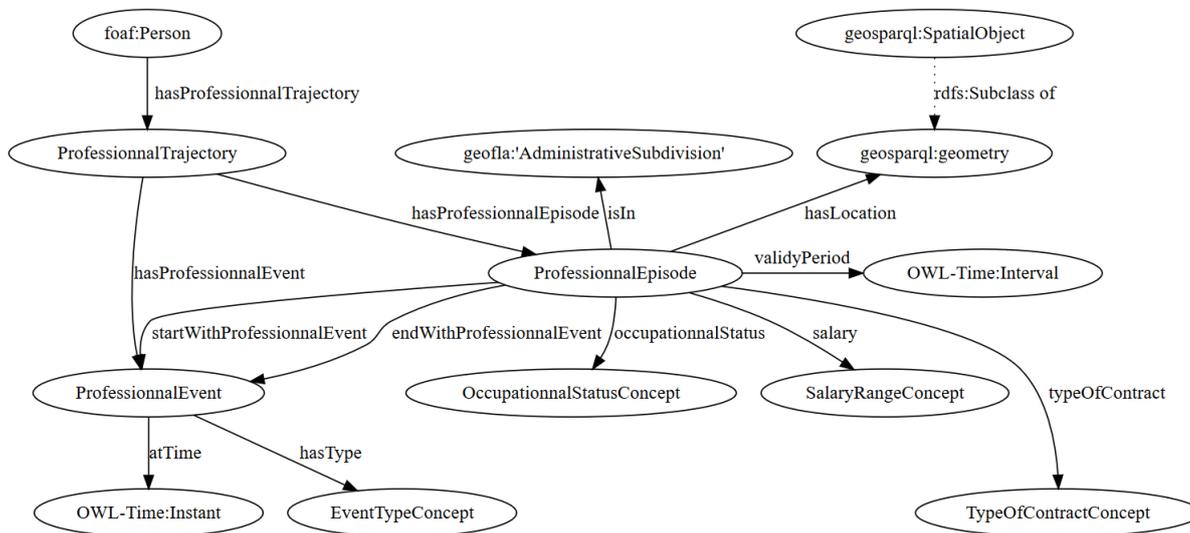


Figure 7.3 – Ontologie de la trajectoire professionnelle

La figure 7.3 représente l'ontologie de trajectoire de vie du point de vue professionnel. Comme la trajectoire résidentielle, cette trajectoire thématique est une trajectoire géographique, elle utilise donc le même *design pattern*. Le mot utilisé pour marquer les classes pour ce point de vue est "Professional".

La trajectoire professionnelle comprend également trois caractéristiques thématiques. Le type de contrat (propriété `typeOfContract`) est caractérisé par la classe *TypeOfContractConcept*. Cette classe fait référence à un vocabulaire contrôlé créé lors de la création de cette ontologie présenté dans la section 4 de l'annexe E. Pour décrire le statut professionnel, nous utilisons un vocabulaire publié par l'INSEE<sup>3</sup>. Ce vocabulaire permet de caractériser la situation professionnelle à quatre niveaux de détails : catégorie socioprofessionnelle agrégée, catégorie socioprofessionnelle détaillée et profession. L'utilisation de ce vocabulaire contrôlé permet de collecter des données à différents niveaux de granularité (voir section 3 de ce chapitre) ainsi que d'effectuer des requêtes à de multiples niveaux de granularité thématique lors de la phase d'analyse (voir section 4 de ce chapitre). Sur la figure 7.3, ce statut professionnel (propriété *occupationalStatus*) est caractérisé par la classe *OccupationalStatusConcept*. Cette classe représente les statuts professionnels dans le vocabulaire choisi. Enfin, de la même façon que le loyer dans la trajectoire résidentielle, le salaire (propriété *salary*) est caractérisé par des intervalles de salaire à l'aide d'un vocabulaire contrôlé. La classe *SalaryRangeConcept* est une super-classe permettant de caractériser tous les intervalles dans ce vocabulaire contrôlé présenté dans la section 5 l'annexe de E.

### La trajectoire familiale

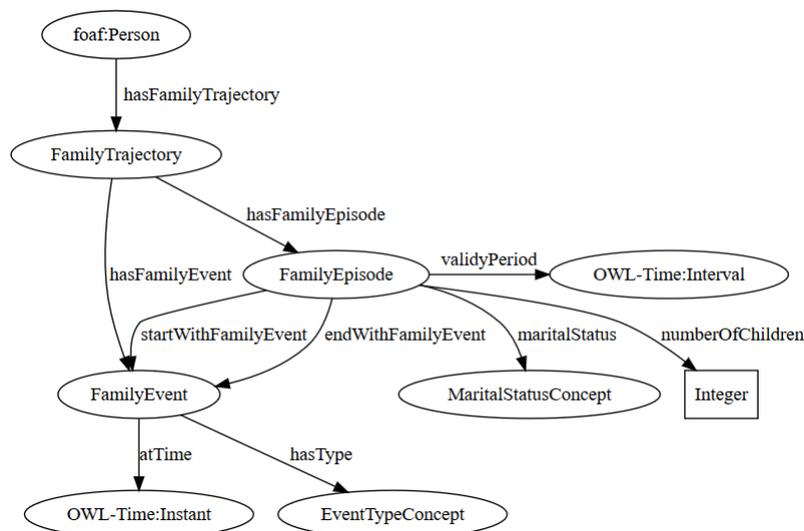


Figure 7.4 – Ontologie de la trajectoire familiale

3. <http://rdf.insee.fr/codes/index.html>

La figure 7.4 représente l'ontologie de trajectoire de vie du point de vue familial. Contrairement aux deux trajectoires thématiques précédentes, ce n'est pas une trajectoire géographique, le *design pattern* utilisé est donc celui dédié aux trajectoires métaphoriques. Le mot utilisé pour marquer les classes pour ce point de vue est "family".

La trajectoire familiale comprend deux caractéristiques thématiques. Le nombre d'enfants du foyer (propriété *numberOfChildren*) est caractérisé par un entier (Integer). Le statut marital (propriété *maritalStatus*) est caractérisé par la classe *maritalStatusConcept* appartenant à un vocabulaire contrôlé, lui aussi créé pour cette ontologie et présenté dans la section 6 de l'annexe E.

### 3 Collecte de trajectoires de vie

Il est possible d'utiliser les APIs que nous proposons pour la phase de collecte de données. Pour ce cas d'application, nous avons choisi de collecter des données auprès d'un échantillon de la population de la métropole de Grenoble en utilisant une application dédiée. Nous avons réalisé les développements correspondants. Les différentes interfaces de cette application sont disponibles en annexe G. Nous montrons ici comment l'API du module Ontology Manager permet une création facilitée de l'interface et comment l'API du module Collect Manager permet un enregistrement facilité des données. Les différentes méthodes auxquelles nous faisons référence sont celles décrites dans la section 4 du chapitre VI.

**Création de l'interface et utilisation de l'API Ontology Manager** L'application que nous avons développée comprend donc une interface de collecte qui se présente sous la forme d'un formulaire pour chaque trajectoire thématique. La figure 7.5 représente l'interface de collecte d'un épisode de type résidentiel.

Cette interface est une proposition de formulaire développé dans le cadre de nos tests. Il s'agit ici de récolter des informations relatives à la trajectoire résidentielle d'individus.

Episode résidentiel

Date d'emménagement

Nombre de pièces

Type de logement

Situation résidentielle

Loyer

Appartement

Locataire

600 - 900 €

Rechercher un lieu

Adresse

Code postal 38000

Ville Grenoble

Pays France



Raisons du déménagement

Quelles raisons ont influencé votre choix ?

1 Événement professionnel

Figure 7.5 – Interface de collecte d'un épisode résidentiel

La partie supérieure de la figure montre les différents attributs qu'il convient de compléter pour décrire un épisode résidentiel. Ces attributs (*Nombre de pièces*, *Type de logement*, *Situation résidentielle* et *Loyer*) correspondent aux attributs de l'épisode dans l'ontologie de trajectoire résidentielle (figure 7.2). Pour accéder à chaque attribut, le développeur utilise la méthode `getAttributes(String viewstamp)` qui renvoie toutes les ressources correspondantes aux attributs pour un point de vue donné. Les différents *labels*

indiqués dans le formulaire sont établis à partir des différents *labels* des classes correspondantes dans l'ontologie. Pour y accéder le développeur de l'interface utilise, pour chaque ressource renvoyée par la méthode précédente, la méthode `getAttributeLabel(Ressource attribute)` qui renvoie une chaîne de caractères correspondant au *label* de l'attribut dont la ressource est passée en paramètre.

Les différentes valeurs acceptées dans l'interface pour chacun de ces attributs correspondent également aux différentes valeurs qu'il est possible de leur affecter en restant conforme à l'ontologie. Par exemple, le champ *nombre de pièces* devra être complété par un entier. Dans le cas où un attribut est défini dans l'ontologie grâce à un vocabulaire contrôlé, les différentes valeurs de ce vocabulaire sont proposées comme les différents choix pour compléter le champ correspondant, c'est ce que l'on peut par exemple voir pour l'attribut *Situation résidentielle* sur la figure 7.5. Le développeur peut accéder aux types de données ou au vocabulaire correspondant grâce à la méthode `getAttributeValue(Ressource attribute)` qui fournit l'information pour l'attribut identifié par la ressource passée en paramètre.

Cette approche facilite donc le travail du développeur et, si aucune information externe n'est requise, permet d'envisager d'automatiser en partie la construction des formulaires de collecte. Nous abordons la question de l'automatisation de la construction du formulaire de collecte dans la partie VIII, dans laquelle nous abordons les perspectives de ces travaux.

La localisation géographique du lieu de résidence peut être réalisée au moyen d'une carte dotée de fonctionnalités de recherche textuelle pour faciliter la saisie : par exemple, lorsque le nom d'un quartier est donné (comme ici "Ile Verte"), la carte se centre sur le polygone correspondant et permet de zoomer si besoin pour donner des informations plus précises (place, rue, etc.). Enfin, la partie basse de la figure 7.5 montre comment l'utilisateur peut saisir des facteurs explicatifs liés au déménagement (événement) qui marque le début de l'épisode décrit : il peut choisir un type d'événement dans la liste (ici seul est visible un événement de la thématique familiale, mais il pourrait s'agir d'un événement professionnel) et renseigner les champs associés qui deviendront alors accessibles.

La même démarche est suivie pour construire les formulaires correspondants aux autres points de vue. Ces formulaires sont présentés dans l'annexe G.

**Instances de trajectoires de vie** Les instances de trajectoires de vie ainsi collectées sont alors transmises au *Triple Store* grâce aux requêtes présentées

dans la section 4 de la partie VI. La figure 7.6 présente un exemple d'instance de trajectoire résidentielle (compatible avec l'ontologie présentée sur la figure 7.2) annotée et stockée dans le *Triple Store*.

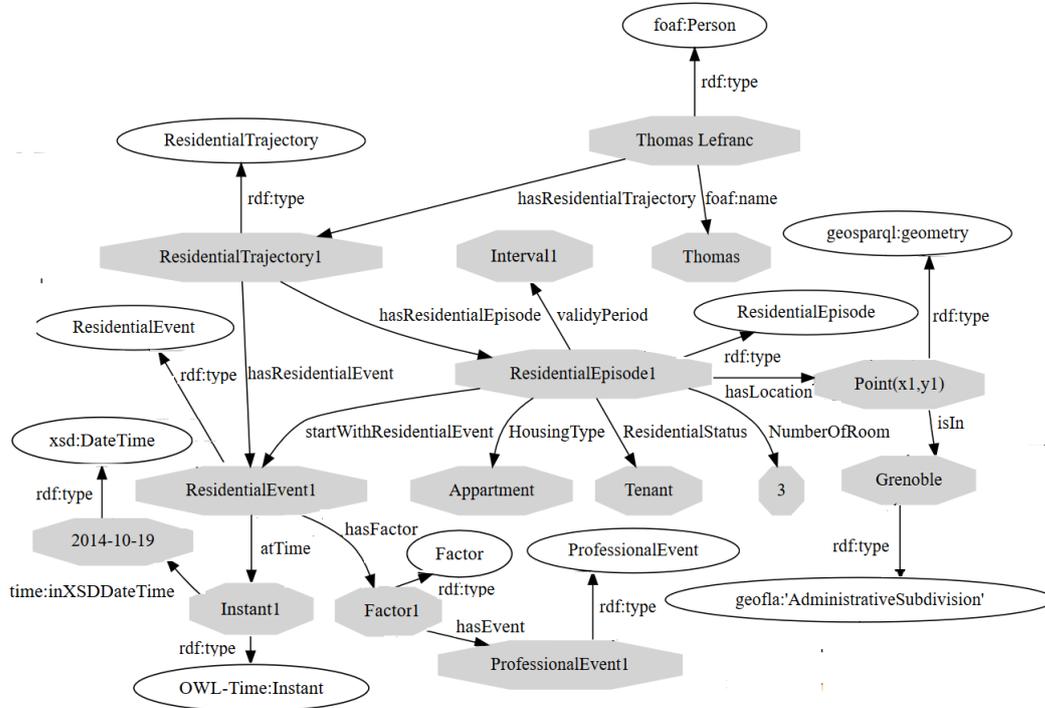


Figure 7.6 – Un exemple d'instance de trajectoire résidentielle

Cette figure représente un extrait de la trajectoire résidentielle de l'individu Thomas (*ResidentialTrajectory1*) qui inclut un épisode résidentiel (*ResidentialEpisode1*) qui débute avec le déménagement de Thomas (*ResidentialEvent1*) qui a eu lieu le 19-10-2014. Durant cet épisode résidentiel, Thomas était le locataire d'un appartement trois pièces. Nous savons que la localisation du logement (*hasLocation*) est située dans l'unité administrative de Grenoble, en France. La figure nous apprend également qu'un des facteurs explicatifs (*Factor1*) de l'événement résidentiel *ResidentialEvent1* est un événement professionnel (*ProfessionalEvent1*).

## 4 Analyse des trajectoires de vie collectées

Notre modèle, qui caractérise les facteurs explicatifs des événements de vie, permet de s'intéresser de manière fine aux raisons des déménagements. Avec des spécialistes du domaine, nous avons sélectionné des questions pertinentes sur les raisons des déménagements dans la métropole de Grenoble.

Pour répondre à ces questions, nous utilisons le langage SPARQL ainsi que les opérateurs définis dans la section 5 du chapitre VI. Ces requêtes permettent d’obtenir des résultats conformes à l’ontologie **Life Trajectory Ontology** (*lto*) précédemment produite et publiée en ligne<sup>4</sup>. Pour des raisons de lisibilité des requêtes, nous n’indiquons la signification des préfixes que dans la première requête où ils sont utilisés. Les résultats sont obtenus au format RDF et correspondent aux URIs des ressources ou aux variables présentes dans le *Triple Store*. Ces résultats, obtenus à partir de critères spatiaux, temporels et/ou thématiques peuvent être exploités pour construire des représentations graphiques utilisables par les sociologues.

Comme nous le rappelions en introduction, la compréhension des trajectoires résidentielles est un enjeu important pour les métropoles et les acteurs de l’aménagement urbain. Ainsi, les couples avec enfants privilégient un habitat éloigné des centres métropolitains, tandis que ceux-ci attirent en plus grande proportion à la fois les jeunes nouvellement indépendants et les retraités [Robette et al., 2012]. Cet exode des couples actifs avec enfants vers les zones périurbaines entraîne pour les villes des périphéries des difficultés à fournir les aménités nécessaires (par exemple, des crèches et des écoles), mais aussi une augmentation des temps de trajets domicile-travail, générateur de pollution et de congestion du trafic.

Notre objectif est ici de vérifier que notre modèle offre le potentiel pour, à terme, vérifier et comprendre cet état de fait dans l’agglomération grenobloise. À cet effet, nous déroulons deux scénarios d’analyse des données collectées. Avant de présenter ces scénarios, nous montrons d’abord comment acquérir une vue générale des données collectées, puis comment les explorer à différents niveaux de granularité.

**Vue générale des données collectées** Nous montrons ici comment acquérir une vue générale des données présentes dans le *Triple Store*, c’est-à-dire d’obtenir des informations : *i*) sur les individus : genre et âge *ii*) sur les trajectoires elles-mêmes : nombre de trajectoires, d’épisodes, d’événements et de facteurs explicatifs collectés ainsi que leurs répartitions spatiales et temporelles.

— Question 1a : Combien y a-t-il d’individu dans la base de données ?

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/#>
SELECT (count(distinct ?person) as ?personCount)
WHERE {
```

---

4. [purl.org/lto](http://purl.org/lto)

```
?person rdfs:type schema:person .
}
```

La figure 7.7 présente le résultat de cette requête dans le *Triple Store* GraphDB<sup>5</sup>. La variable *persconCount*, de type *xsd:Integer*, pour valeur 31, ce qui signifie que le *Triple Store* abrite 31 *URIs* caractérisant des individus.

Filter query results		Showing results from 1 to 1 of 1. Query took 0.03 s.
		<b>personCount</b>
1	"31"	xsd:integer

Figure 7.7 – Résultat de la requête 1a dans Graphdb

- Question 1b : Quel est l'URI correspondant à chaque individu, et quelle est son année de naissance et son genre ?

```
SELECT ?person ?dateOfBirth ?gender
WHERE {
  ?person rdfs:type schema:person .
  ?person schema:birthdate ?dateOfBirth .
  ?person schema:gender ?gender .
}
```

La figure 7.7 présente un extrait du résultat de cette requête.

	person	dateOfBirth	gender
1	<a href="http://metropolifeline.imag.fr/person/1">http://metropolifeline.imag.fr/person/1</a>	"1981" xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
2	<a href="http://metropolifeline.imag.fr/person/2">http://metropolifeline.imag.fr/person/2</a>	"1958" xsd:integer	<a href="http://schema.org/Female">http://schema.org/Female</a>
3	<a href="http://metropolifeline.imag.fr/person/3">http://metropolifeline.imag.fr/person/3</a>	"1989" xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
4	<a href="http://metropolifeline.imag.fr/person/4">http://metropolifeline.imag.fr/person/4</a>	"1961" xsd:integer	<a href="http://schema.org/Female">http://schema.org/Female</a>
5	<a href="http://metropolifeline.imag.fr/person/5">http://metropolifeline.imag.fr/person/5</a>	"1957" xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
6	<a href="http://metropolifeline.imag.fr/person/6">http://metropolifeline.imag.fr/person/6</a>	"1979" xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
7	<a href="http://metropolifeline.imag.fr/person/8">http://metropolifeline.imag.fr/person/8</a>	"1959" xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
8	<a href="http://metropolifeline.imag.fr/person/9">http://metropolifeline.imag.fr/person/9</a>	"1990" xsd:integer	<a href="http://schema.org/Female">http://schema.org/Female</a>
	...		

Figure 7.8 – Extrait du résultat de la requête 1b dans Graphdb

5. <http://graphdb.ontotext.com/>

- Question 2 : Combien y-a-il d'épisodes, événements et facteurs explicatifs pour un type de trajectoire thématique ? Dans cet exemple, nous nous intéressons à la trajectoire résidentielle.

```

SELECT  (count(distinct ?event) as ?eventCount)
        (count(distinct  ?episode) as  ?episodeCount)
        (count(distinct  ?factor) as  ?factorCount)
WHERE {
  ?episode rdfs:type lto:ResidentialEpisode .
  ?event rdfs:type lto:ResidentialEvent .
  ?factor rdfs:type lto:ExplanatoryFactor .
}

```

La figure 7.7 présente le résultat de cette requête. Le *Triple Store* contient donc 104 épisodes et 104 événements pour lesquels 24 facteurs explicatifs ont été fournis.

	person	dateOfBirth	gender
1	<a href="http://metropolifeline.imag.fr/person/1">http://metropolifeline.imag.fr/person/1</a>	"1981"^^xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
2	<a href="http://metropolifeline.imag.fr/person/2">http://metropolifeline.imag.fr/person/2</a>	"1958"^^xsd:integer	<a href="http://schema.org/Female">http://schema.org/Female</a>
3	<a href="http://metropolifeline.imag.fr/person/3">http://metropolifeline.imag.fr/person/3</a>	"1989"^^xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
4	<a href="http://metropolifeline.imag.fr/person/4">http://metropolifeline.imag.fr/person/4</a>	"1961"^^xsd:integer	<a href="http://schema.org/Female">http://schema.org/Female</a>
5	<a href="http://metropolifeline.imag.fr/person/5">http://metropolifeline.imag.fr/person/5</a>	"1957"^^xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
6	<a href="http://metropolifeline.imag.fr/person/6">http://metropolifeline.imag.fr/person/6</a>	"1979"^^xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
7	<a href="http://metropolifeline.imag.fr/person/8">http://metropolifeline.imag.fr/person/8</a>	"1959"^^xsd:integer	<a href="http://schema.org/Male">http://schema.org/Male</a>
8	<a href="http://metropolifeline.imag.fr/person/9">http://metropolifeline.imag.fr/person/9</a>	"1990"^^xsd:integer	<a href="http://schema.org/Female">http://schema.org/Female</a>
	...		

**Figure 7.9** – Extrait du résultat de la requête 2 dans Graphdb

- Question 3 : Dans quelles communes sont localisés les épisodes résidentiels décrits ?

```

SELECT (count(distinct ?residentialEpisode )
        as ?episodeCount)  ?communeLabel
WHERE {
  ?residentialEpisode rdfs:type lto:ResidentialEpisode.
  ?residentialEpisode lto:isIn ?commune .
  ?commune rdfs:label ?communeLabel .
}
GROUP BY ?communeLabel

```

**Explorer les données à différents niveaux de granularité** Nous nous intéressons dans ces requêtes aux *déplacements* (*géographique* ou *agéographique*) des individus, c'est-à-dire que les trajectoires sont sélectionnées en se basant sur au moins deux épisodes (successifs ou non) contenus dans celle-ci. Afin de situer temporellement ces épisodes l'un par rapport à l'autre, nous utilisons les opérateurs temporels définis dans la section 5 du chapitre VI.

Nous recherchons ici des motifs de déplacements particulier à différents niveaux de granularité. Cela est rendu possible en faisant varier le niveau de granularité, spatial ou thématique, des requêtes. Nous nous intéressons ici aux trajectoires résidentielles et aux trajectoires professionnelles.

**Granularité spatiale** Pour les trajectoires résidentielles nous cherchons à identifier des trajectoires étant passées par *Grenoble*, puis par un canton de la zone péri-urbaine, le canton *Oisans-Romanche*.

- Question 4 : Quelles sont les trajectoires résidentielles passant par Grenoble puis par le canton Oisans-Romanche ?

La requête suivante utilise la fonction `ltofn:before`, définie dans la section 5 du chapitre VI. Celle-ci permet de de s'assurer que l'épisode résidentiel localisé à Grenoble a bien lieu *avant* l'épisode résidentiel localisé dans le canton Oisans-Romanche. Il est également possible d'utiliser ici l'opérateur `ltofn:after`. Nous utilisons ici les données de l'IGN qui caractérise les unités administratives de la France, selon l'ontologie Geofla. Ici, que la localisation du second épisode résidentiel ait été indiqué au niveau de la commune ou au niveau du canton, celle-ci correspond au motif recherché si elle se situe dans le canton désigné. Nous montrons donc grâce à cette requête qu'il est possible d'obtenir, à un niveau de granularité donné caractérisé par l'ontologie, des trajectoires qui ont été collectées à ce niveau ou à un niveau inférieur dans la même requête.

```
PREFIX geofla: <http://data.ign.fr/def/geofla#>
SELECT ?residentialTrajectory
WHERE {
  ?residentialTrajectory rdfs:type lto:ResidentialTrajectory .
  ?residentialTrajectory lto:hasResidentialEpisode ?episode1 .
  ?residentialTrajectory lto:hasResidentialEpisode ?episode2 .
  {?episode1 lto:isIn ?canton .}
UNION
  {?episode1 lto:isIn ?commune .
  ?commune geofla:cant ?canton .}
```

```
?canton rdfs:label 'OISANS ROMANCHE'@fr .
?episode2 lto:isIn ?commune1 .
?commune1 rdfs:label 'GRENOBLE'@fr .
?episode1 ltofn:before ?episode2 .
}
```

La figure 7.10 présente deux URIs correspondant aux résultats de cette requête.

	residentialTrajectory
1	<a href="http://metropolifeline.imag.fr/P34/residentialTraj">http://metropolifeline.imag.fr/P34/residentialTraj</a>
2	<a href="http://metropolifeline.imag.fr/P1/residentialTraj">http://metropolifeline.imag.fr/P1/residentialTraj</a>

**Figure 7.10** – Extrait du résultat de la requête 4 dans Graphdb

Quand cette requête renvoie plusieurs résultats, ce qui est le cas ici, il est possible de s'intéresser à la répartition de ces résultats à un niveau de granularité inférieur. La requête suivante permet de savoir par quelle commune de ce canton passent effectivement ces trajectoires résidentielles.

- Question 5 : Quelles sont les trajectoires résidentielles passant par Grenoble, puis par une commune du canton Oisans-Romanche, et dans chaque cas, quelle est cette commune si elle a été renseignée ?

```
SELECT ?residentialTrajectory ?communeLabel
WHERE {
?residentialTrajectory rdfs:type lto:ResidentialTrajectory .
?residentialTrajectory lto:hasResidentialEpisode ?episode1 .
?residentialTrajectory lto:hasResidentialEpisode ?episode2 .
{?episode1 lto:isIn ?canton .}
UNION
{?episode1 lto:isIn ?commune .
?commune geofla:cant ?canton .
?commune rdfs:label ?communeLabel .}
?canton rdfs:label 'OISANS ROMANCHE'@fr .
?episode2 lto:isIn ?commune1 .
?commune1 rdfs:label 'GRENOBLE'@fr .
?episode1 ltofn:before ?episode2 .
}
```

La figure 7.11 présente deux URIs correspondant aux résultats de cette requête. Pour l'une des deux trajectoires, la commune renseignée pour le second épisode est *Saint-Martin d'Uriage*, pour l'autre l'information n'a pas été renseignée à ce niveau de granularité.

	residentialTrajectory	communeLabel
1	<a href="http://metropolifeline.imag.fr/P34/residentialTraj">http://metropolifeline.imag.fr/P34/residentialTraj</a>	
2	<a href="http://metropolifeline.imag.fr/P1/residentialTraj">http://metropolifeline.imag.fr/P1/residentialTraj</a>	*SAINT-MARTIN-d'URIAGE*@fr

Figure 7.11 – Extrait du résultat de la requête 5 dans Graphdb

**Granularité thématique** Pour la granularité thématique, nous nous intéressons ici aux trajectoires professionnelles. Nous recherchons les individus étant passés du statut *d'employés de commerce* à celui de *commerçants* au cours de leur vie. Nous montrons ensuite comment l'influence de ce motif dans la trajectoire professionnelle sur la trajectoire résidentielle peut être examinée.

- Question 6 : Quelles sont les trajectoires professionnelles des individus ayant été *employés de commerce* puis *commerçants et assimilés* ?

```

SELECT ?professionalTrajectory
WHERE {
    ?professionalTrajectory rdfs:type lto:professionalTrajectory .
    ?professionalTrajectory lto:hasProfessionalEpisode ?episode1 .
    ?professionalTrajectory lto:hasProfessionalEpisode ?episode2 .

    {?episode1 lto:occupationalStatus ?profession .
    ?cspa <http://www.w3.org/2004/02/skos/core#narrower>
        ?profession .}
    UNION
    {?episode1 lto:occupationalStatus ?cspa .}

    ?cspa skos:prefLabel
        'Employés de commerce'@fr .

    ?episode2 lto:occupationalStatus ?pcs2 .
    ?pcs2 skos:prefLabel
        'Commerçants et assimilés'@fr .}

    ?episode1 ltofn:before ?episode2 .}
    
```

Si cette requête ne renvoie pas (suffisamment) de résultats, il est possible d'augmenter son niveau de granularité. Grâce à la requête suivante, on répond à la question précédente en remplaçant *employé de commerce* par *employé*, qui correspond à la catégorie socio-professionnelle contenant les employés de commerce.

- Question 7 : Quelles sont les trajectoires professionnelles des individus ayant été *employées* puis *commerçants et assimilés* ?

```

SELECT ?professionalTrajectory ?csp
WHERE {
  ?professionalTrajectory rdfs:type lto:professionalTrajectory .
  ?professionalTrajectory lto:hasProfessionalEpisode ?episode1 .
  ?professionalTrajectory lto:hasProfessionalEpisode ?episode2 .

  {?episode1 lto:occupationalStatus ?profession .
  ?cspd skos:narrower ?profession .
  ?csp skos:narrower ?cspd .
  ?cspa skos:narrower ?csp .}
  UNION
  {?episode1 lto:occupationalStatus ?cspd .
  ?csp skos:narrower ?cspd .
  ?cspa skos:narrower ?csp .}
  UNION
  {?episode1 lto:occupationalStatus ?csp .
  ?cspa skos:narrower ?csp .}
  UNION
  {?episode1 lto:occupationalStatus ?cspa .}

  ?cspa <http://www.w3.org/2004/02/skos/core#prefLabel>
    'Employés'@fr .

  ?episode2 lto:occupationalStatus ?csp2 .
  ?csp2 <http://www.w3.org/2004/02/skos/core#prefLabel>
    'Commerçants et assimilés'@fr .

  ?episode1 ltofn:before ?episode2 .}

```

Nous avons montré ici comment effectuer des requêtes à différents niveaux de granularité sur les trajectoires. Pour un niveau de granularité donné, tous les niveaux de granularité inférieurs à laquelle l'information est susceptible d'avoir été collectée peuvent être considérés.

Dans le scénario suivant, nous montrons comment il est possible d'analyser le parcours résidentiel d'un individu afin de déterminer, par exemple, l'influence de son parcours professionnel sur ces choix résidentiels.

**Construction d'une grille de lecture** Dans ce scénario, nous établissons une grille de lecture des facteurs explicatifs des déménagements. Nous montrons ici comment notre approche peut être utilisée lors d'une phase d'analyse et de construction de la connaissance sociologique [Sophie et al., 2014]. Dans cette phase, il s'agit de fabriquer une grille de lecture applicable aux trajectoires de vie à partir de cas particulier [Sophie et al., 2014]. Notre approche permet ici au sociologue d'activer ses "devoirs comparatistes" [Passeron, 1990] en permettant à celui-ci d'analyser plusieurs trajectoires de vie pour trouver des configurations sociales similaires.

Pour ce faire, nous établissons une grille de lecture pertinente d'une trajectoire de vie à partir d'une seule requête SPARQL, en ayant connaissance de l'*URI* qui caractérise l'individu dans le *Triple Store*. Une telle grille de lecture peut donc être construite automatiquement pour chaque individu présent dans la base. Elle peut également être modifiée en fonction des individus. De plus, les capacités du langage SPARQL, qui permet des regroupements de facteurs explicatifs similaires, peuvent être utilisées pour mettre en évidence des équivalences ou des similitudes entre les cas individuels.

La requête suivante permet la constitution d'une grille de lecture de la trajectoire de l'individu identifié précédemment dont l'*URI* est `<http://metropolifeline.imag.fr/person/1>`.

```
SELECT ?residentialEvent ?communeLabel ?explainingEventType
       ?explainingThematic ?explanation WHERE {
<http://metropolifeline.imag.fr/person/1>
  lto:hasResidentialTrajectory ?residentialTrajectory .
?residentialTrajectory lto:hasResidentialEvent ?residentialEvent .
?residentialEvent lto:hasFactor ?explanatoryFactor .
  ?explanatoryFactor lto:hasExplanation ?explanation .
?explanatoryFactor lto:hasExplainingEvent ?explainingEvent .
?residentialEpisode lto:startWithResidentialEvent ?residentialEvent .
?residentialEpisode lto:isIn ?commune .
?commune rdfs:label ?communeLabel .
?explainingEvent lto:eventType ?explainingEventType .
?ExplainingTrajectory lto:hasEvent ?explainingEvent .
?ExplainingTrajectory rdfs:type ?explainingThematic .
```

La figure 7.12 présente un extrait du résultat de cette requête.

	residentialEvent ↕	communeLabel ↕	explainingEventType ↕	explainingThematic ↕	explanation ↕
1	<a href="http://metropolifeline.imag.fr/P1/residentialTraj/Event1">http://metropolifeline.imag.fr/P1/residentialTraj/Event1</a>	"SAINT-MARTIN-d'URIAGE"@fr	Ito:Couple	Ito:FamilialTrajectory	Emménagement en couple suite à une rencontre
2	<a href="http://metropolifeline.imag.fr/P1/residentialTraj/Event2">http://metropolifeline.imag.fr/P1/residentialTraj/Event2</a>	"GRENOBLE"@fr	Ito:BreakUp	Ito:FamilialTrajectory	Quitter l'appartement occupé en concubinage
3	<a href="http://metropolifeline.imag.fr/P1/residentialTraj/Event2">http://metropolifeline.imag.fr/P1/residentialTraj/Event2</a>	"GRENOBLE"@fr	Ito:Couple	Ito:FamilialTrajectory	Se rapprocher du centre ville pour pouvoir sortir
4	<a href="http://metropolifeline.imag.fr/P1/residentialTraj/Event3">http://metropolifeline.imag.fr/P1/residentialTraj/Event3</a>	"SAINT-MARTIN-d'URIAGE"@fr	Ito:BreakUp	Ito:FamilialTrajectory	
5	<a href="http://metropolifeline.imag.fr/P1/residentialTraj/Event3">http://metropolifeline.imag.fr/P1/residentialTraj/Event3</a>	"SAINT-MARTIN-d'URIAGE"@fr	Ito:StartingBusiness	Ito:ProfessionalTrajectory	Achat d'un magasin, retour au domicile des parents pour raisons financières

**Figure 7.12** – Extrait du résultat de la requête de création de la grille de lecture dans Graphdb

Le tableau 4 reprend les résultats de cette requête dans une grille de lecture. Chaque colonne correspond à une des variables sélectionnées dans la requête (*?residentialEvent*, *?explanatoryFactor*, *?explainingThematic*, *?explainingEventType*, *?explanation* et *?communeLabel*). Nous constatons que trois événements résidentiels de l'individu sont caractérisés dans la base de données. Le premier correspondant à un emménagement à Saint-Martin d'Uriage. Il a un facteur explicatif, qui désigne une raison familiale pour le déménagement, plus exactement *une mise en couple*. Une explication concernant ce déménagement est également fournie par l'individu. Le second événement résidentiel correspond à un emménagement à Grenoble. Il a deux facteurs explicatifs, tous deux pointant des raisons familiales (une *séparation* et une *mise en couple*).

Finalement, le troisième déménagement a deux facteurs explicatifs, une *séparation* et la *création d'un commerce*. L'explication liée au second facteur explicatif nous permet de comprendre, qu'il s'agit d'un retour au domicile des parents lié à des raisons financières en lien avec l'événement professionnel de création d'un commerce. Les données sur que présente cette grille de lecture sont inspirées d'un cas réel. Il est possible de choisir d'intégrer davantage d'informations dans la grille (statut résidentiel lors de l'emménagement, statut professionnel de l'individu...) en fonction de l'intérêt du spécialiste souhaitant la construire.

Événement	Commune d'emménagement	Thématiques explicatives	Types d'événement explicatif	Explications
1	Saint-Martin d'Uriage	Familiale	Mise en coupe	<i>Emménagement en couple suite à une rencontre</i>
2	Grenoble	Familiale	Séparation	<i>Quitter l'appartement occupé en concubinage</i>
		Familiale	Mise en couple	<i>Se rapprocher du centre ville pour pouvoir sortir</i>
3	Saint-Martin d'Uriage	Familiale	Séparation	
		Professionnelle	Création d'un commerce	<i>Achat d'un magasin, retour au domicile des parents pour raisons financières</i>

**Table 7.1** – Les événements et les caractéristiques d'épisodes influençant les déménagements, classés par thématique

Notre approche permet ainsi d'effectuer une étape d'analyse croisée des cas particuliers, sans passer par une analyse manuelle de chaque cas, par exemple la création et l'analyse de représentation graphique pour chaque trajectoire de vie [Sophie et al., 2014]. La constitution de telles grilles de lecture permet de mettre en évidence des similitudes pouvant correspondre à des catégories de motivations pour les déménagements, voire à différentes stratégies dans les trajectoires résidentielles. Les tableaux de ce type sont des outils qui peuvent être utilisés par la suite pour d'autres analyses de trajectoires de vie. Ils permettent également aux sociologues d'interagir entre eux, en sachant à quel type d'informations un jeu de données permet d'accéder.

**Aller plus loin dans la compréhension des choix résidentiels** Dans le second scénario, nous recherchons tout d'abord les caractéristiques familiales et

professionnelles des épisodes décrivant une période d'habitation *i*) dans les villes-centres des métropoles *ii*) dans les zones péri-urbaines. Il est ainsi possible de vérifier la situation précédemment décrite concernant la proportion de jeunes sans enfants et de retraités et celle de couples avec enfants. Ensuite, nous nous intéressons aux facteurs explicatifs des déménagements fournis par les individus déménageant de la ville-centre vers la zone péri-urbaine, puis nous montrons comment ces facteurs explicatifs peuvent être analysés grâce aux fonctions génériques que nous avons proposées. Pour cet exemple, les questions se concentrent sur Grenoble, la ville-centre, et sur Saint-Martin d'Uriage, une ville de la zone péri-urbaine.

- Question 8 : Quelles sont les situations familiales et professionnelles des individus pendant leur(s) épisode(s) résidentiel(s) à Grenoble?

```
SELECT ?familyEpisode ?professionalEpisode .
WHERE {
  ?person rdfs:type schema:Person .

  ?person lto:dateOfBirth ?dateOfBirth .
  ?person lto:hasResidentialTrajectory ?residentialTrajectory .
  ?residentialTrajectory lto:hasResidentialEpisode
      ?residentialEpisode .
  ?ResidentialEpisode LTO:hasLocation ?Location .
  ?Location LTO:isIn ?commune .
  ?commune rdfs:label 'GRENOBLE'@fr .

  ?person lto:hasProfessionalTrajectory ?professionalTrajectory .
  ?professionalTrajectory lto:hasProfessionalEpisode
      ?professionalEpisode .

  ?person lto:hasFamilyTrajectory ?familyTrajectory .
  ?familyTrajectory lto:hasFamilyEpisode ?familyEpisode .

  ?residentialEpisode ltofn:overlaps ?professionalEpisode .
  ?residentialEpisode ltofn:overlaps ?familyEpisode .
}
```

Cette requête utilise un opérateur générique `ltofn:overlaps`, qui vérifie que les périodes de validité respectives de deux épisodes se chevauchent.

La requête précédente est possible pour une ville de la zone péri-urbaine, par exemple Saint-Martin d'Uriage.

- Question 9 : Quels sont les facteurs explicatifs des déménagements de Grenoble à Saint Martin d'Uriage?

```
SELECT ?explanatoryFactor
WHERE {
?explanatoryFactor rdfs:type lto:ExplanatoryFactor .

?explanatoryFactor ltofn:isFactorOutFrom ?commune1 .
?commune1 rdfs:label 'GRENOBLE'@fr .

?explanatoryFactor ltofn:isFactorEnterIn ?commune2 .
?commune2 rdfs:label 'SAINT-MARTIN D'URIAGE'@fr .
}
```

Pour répondre à cette question, l'opérateur `ltofn:isFactorOutFrom` filtre les facteurs explicatifs pour ne conserver que l'ensemble de ceux qui concernent un événement marquant la fin d'un épisode localisé à Grenoble. L'opérateur `ltofn:isFactorEnterIn` filtre les facteurs explicatifs pour ne conserver que l'ensemble de ceux qui concernent un événement marquant le début d'un épisode localisé à Saint-Martin d'Uriage. L'utilisation conjointe de ces deux opérateurs permet donc de conserver uniquement les facteurs explicatifs d'un événement (un déménagement) marquant le passage de Grenoble à Saint-Martin d'Uriage.

- Question 10 : Quand ces individus fournissent ces facteurs explicatifs, quelle est leur situation professionnelle?

```
SELECT ?explanatoryFactor ?professionalEpisode
WHERE {
?explanatoryFactor rdfs:type lto:ExplanatoryFactor .

?explanatoryFactor ltofn:isFactorOutFrom ?commune1 .
?commune1 rdfs:label 'GRENOBLE'@fr .

?explanatoryFactor ltofn:isFactorEnterIn ?commune2 .
?commune2 rdfs:label 'SAINT-MARTIN D'URIAGE'@fr .

?professionalEpisode rdfs:type lto:ProfessionalEpisode .
?professionalEpisode ltofn:episodeAtFactorDate
    ?explanatoryFactor .
}
```

Pour répondre à cette question, nous associons à la requête précédente l'opérateur `ltofn:episodeAtFactorDate`. Il est utilisé ici pour filtrer, parmi les épisodes professionnels, celui qui *i*) appartient à l'individu qui a fourni le facteur *ii*) est valide à la date du déménagement expliqué par ce facteur.

Nous cherchons maintenant à savoir, dans le cas où un événement est en partie expliqué par une naissance, quels sont les autres facteurs explicatifs qui lui sont associés (si ils existent).

- Question 11 : Parmi ces facteurs explicatifs, quels sont ceux qui expliquent des événements déjà en partie expliqués par une naissance ?

```
SELECT ?explanatoryFactor ?professionalEpisode
WHERE {
?explanatoryFactor rdfs:type lto:ExplanatoryFactor .

?explanatoryFactor ltofn:isFactorOutFrom ?commune1 .
?commune1 rdfs:label 'GRENOBLE'@fr .

?explanatoryFactor ltofn:isFactorEnterIn ?commune2 .
?commune2 rdfs:label 'SAINT-MARTIN D'URIAGE'@fr .

?explanatoryFactor ltofn:isFactorOfExplainedEventType lto:ChildBirth
}
```

- Question 12 : Quelle est la situation résidentielle de ces individus dans leur nouveau logement, autrement dit quelles sont les caractéristiques des logements qui sont alors choisis ainsi que le nouveau statut résidentiel (locataire, propriétaire...) des individus ?

```
SELECT ?explanatoryFactor ?residentialEpisode
WHERE {
?explanatoryFactor rdfs:type lto:ExplanatoryFactor .

?explanatoryFactor ltofn:isFactorOutFrom ?commune1 .
?commune1 rdfs:label 'GRENOBLE'@fr .

?explanatoryFactor ltofn:isFactorEnterIn ?commune2 .
?commune2 rdfs:label 'SAINT-MARTIN D'URIAGE'@fr .

?residentialEpisode rdfs:type lto:ResidentialEpisode .
```

```
?residentialEpisode ltofn:episodeAfterFactor ?explanatoryFactor .  
}
```

Cette requête utilise l'opérateur `ltofn:episodeAfterFactor` afin de trouver l'épisode qui correspond à la nouvelle situation résidentielle après l'événement expliqué par ce facteur explicatif. Une requête semblable peut être faite en remplaçant l'opérateur `ltofn:episodeAfterFactor` par `ltofn:episodeBeforeFactor` pour connaître la situation de l'individu *avant* le facteur explicatif (lorsqu'il habitait donc à Grenoble, avant de déménager).

## 5 Conclusion

Nous avons obtenu grâce à notre modèle des résultats conformes à des critères spatiaux, temporels et/ou thématiques exploitant l'ontologie de trajectoire de vie. Nous avons montré que des requêtes peuvent être effectuées à différents niveaux de granularité et obtenir des résultats à un niveau de granularité donné, même si les informations ont été collectées à un niveau de granularité inférieur. Nous avons ensuite montré comment notre approche peut être utilisée lors d'une phase de construction de la connaissance sociologique, en construisant une grille de lecture à partir des données présentes dans le *Triple Store*. Nous avons également utilisé les opérateurs SPARQL génériques que nous avons définis précédemment, pour exploiter les facteurs explicatifs collectés auprès des individus.

# Conclusion



# Conclusion et perspectives

---

## 1 Conclusion

Nous avons proposé une approche globale d'étude de *trajectoires de vie*. Cette approche permet de créer un *modèle de trajectoire de vie* valable pour tout cas d'application, ainsi que de collecter et d'analyser des données conformes au modèle créé. Nous avons fourni une méthodologie et une architecture générique permettant de mettre en place cette approche. Celle-ci met à profit le Web sémantique grâce à la réutilisation d'ontologies existantes pour la modélisation et à la réutilisation de données externes pour l'analyse des trajectoires.

Nos contributions s'articulent au sein d'une approche générique d'étude des trajectoires de vie qui comprend leur modélisation, leur collecte et leur analyse. Cette approche intègre des contributions concernant la modélisation, la collecte et l'analyse *i)* de trajectoires *multipoints de vue ii)* de trajectoires *métaphoriques iii)* de *facteurs explicatifs* d'une trajectoire *iv)* de trajectoires multigranulaires.

Afin d'assurer la généralité de notre approche, nous avons proposé *i)* un patron de conception d'ontologie et des algorithmes paramétrables lors de la phase de modélisation *ii)* l'utilisation de patrons de requêtes lors de la phase de collecte *iii)* des fonctions génériques d'exploitation des trajectoires lors de la phase d'analyse. Ces fonctions ont été implémentées dans le langage SPARQL et prennent la forme de propriétés utilisables dans les requêtes.

Notre approche repose sur un patron de conception d'ontologie original, qui permet la création d'un modèle qui intègre de multiples perspectives. La trajectoire de vie est alors composée de plusieurs trajectoires thématiques qui sont autant de points de vue thématiques selon lesquels elle peut être observée. Ces trajectoires thématiques sont *métaphoriques* et peuvent être *géographiques* ou *agéographiques*.

Nous avons proposé un modèle de facteurs explicatifs. Ceux-ci se déclinent en une typologie et peuvent être intégrés au modèle final pour y ajouter des éléments de compréhension des événements de vie des individus. Ils permettent de caractériser les éléments qui influencent les événements de vie,

qu'ils soient internes à la trajectoire de l'individu, ou externes à celle-ci, c'est-à-dire dépendant de l'environnement dans lequel il évolue, ou bien qu'ils soient issus de la trajectoire d'un autre individu. Ces facteurs explicatifs ont été collectés et analysés dans les phases correspondantes.

Nous avons également montré comment notre modèle permet de modéliser, collecter, et analyser une information multigranulaire sur les trajectoires de vie. L'utilisation de vocabulaires contrôlés pour la description de l'information thématique [Peuquet, 1994] permet de décrire et de stocker cette information à différents niveaux de granularité.

Nous avons validé notre approche pour le cas d'application de l'étude des trajectoires résidentielles. Pour ce cas d'application, nous avons proposé une ontologie de trajectoire de vie, puis une interface de collecte de données. Enfin, nous avons obtenu, grâce à des requêtes exploitant cette ontologie, des résultats conformes à des critères spatiaux, temporels et/ou thématiques exploitant l'ontologie de trajectoire de vie et nous avons utilisé les fonctions précédemment définies pour mettre en place une analyse du choix résidentiel.

## 2 Perspectives

Nous dressons dans cette section les perspectives ouvertes par nos travaux en distinguant *i*) celles qui s'inscrivent dans leur continuité afin de permettre l'amélioration de notre approche *ii*) celles qui concernent, plus généralement, la modélisation, l'acquisition et l'exploitation de trajectoires.

### 2.1 Validation de l'approche et évaluation des opérateurs

Dans l'objectif de poursuivre la validation de notre approche, nous proposons d'effectuer tout d'abord une collecte de données plus importante pour le cas spécifique de l'étude des trajectoires résidentielles, grâce à l'interface de collecte que nous avons développée. Cette collecte doit permettre d'effectuer un travail sur un échantillon représentatif (à l'échelle de la métropole grenobloise). Nous souhaitons également appliquer notre approche à un autre cas d'étude de trajectoires de vie. Ensuite, nous proposons d'évaluer les performances des fonctions proposées et particulièrement de leur implémentation dans le langage SPARQL.

## 2.2 Création d'une interface d'étude des trajectoires de vie

Les autres travaux futurs concernant notre approche s'inscrivent dans la création d'une interface permettant la modélisation, la collecte et l'exploitation de trajectoires de vie. L'architecture que nous avons présentée ici constitue une base concrète pour mettre en place une telle interface. Concernant la modélisation, une interface permettant au modélisateur de paramétrer une ontologie doit être développée. Cette interface peut s'appuyer sur les algorithmes permettant la création de l'ontologie de trajectoires de vie à partir du patron de conception. Concernant la collecte de données, des travaux doivent être menés pour permettre la génération d'une interface de collecte à partir de l'ontologie créée. Les outils que nous proposons pour *faciliter* la construction de cette interface devront être utilisés dans cette optique. Concernant l'exploitation des données, nous envisageons la création d'une interface de construction de requêtes SPARQL. Des travaux doivent être menés en tenant compte des spécificités du modèle de trajectoire de vie proposé. Cette interface de création de requête doit notamment permettre la construction d'une (partie de) trajectoire de vie sur la base de laquelle le *matching* peut être opéré. Une étape intermédiaire est la mise en place d'un SPARQL *endpoint*, permettant d'interroger les données qui implémentent les fonctions que nous avons proposées. Il est d'ores et déjà possible de développer une telle interface en utilisant *Apache Jena Fuseki*<sup>1</sup>, le serveur SPARQL fourni avec l'API *Jena*.

## 2.3 Modélisation de trajectoires

**Trajectoires sémantiques** Nous avons constaté dans l'état de l'art que les différents travaux sur les trajectoires sémantiques s'intéressent de plus en plus à la sémantique des trajectoires. Ceci amène les chercheurs à considérer la sémantique selon de multiples points de vue. En reportant leur intérêt sur la sémantique de ces trajectoires, les chercheurs considèrent en fait de plus en plus à l'utilisation *métaphorique* du terme de *trajectoire* : la position spatiale de l'individu n'est plus qu'un élément parmi d'autres pour comprendre le *sens* de sa trajectoire. Notre modèle de trajectoires pourra être utilisé à une échelle plus fine, pour caractériser par exemple, des déplacements quotidiens, et la viabilité des concepts que nous proposons (épisodes, événements, facteurs explicatifs) à cette échelle devra être évaluée.

---

1. <https://jena.apache.org/documentation/fuseki2/>

## 2.4 Acquisition des données

**Motivation des contributeurs** La complexité du modèle vers laquelle on peut rapidement tendre (en multipliant les thématiques, les attributs descriptifs de celles-ci, *etc.*) ne doit pas être un frein à la participation des contributeurs. Il convient donc de s'attarder sur les moyens qui permettront de garantir l'alimentation du modèle en données. Au delà d'une réflexion sur les bonnes propriétés des interfaces proposées (intuitive, efficace, *etc.*), une première voie est celle de la mise en exergue des paramètres qui susciteront la participation des contributeurs. Autrement dit, que faut-il apporter ou offrir à l'individu pour l'amener à participer ? Si, pour certains utilisateurs-contributeurs, la cause scientifique peut suffire à motiver la participation, il est patent que d'autres ne seront incités que par un intérêt plus personnel lié à la participation au projet. Un travail devra donc être mené sur ce point pour par exemple dresser une typologie des attentes, en fonction de profils d'habitants qui serviront de guide pour l'adaptation des outils développés aux différentes populations sollicitées lors de campagnes d'enquêtes.

**Diversification des sources de données** Finalement, d'autres sources de données que la collecte auprès des individus devront être envisagées. L'une des voies à suivre est celle de l'interopérabilité avec les réseaux sociaux (Facebook, LinkedIn, *etc.*), car ceux-ci couvrent de fait des champs informationnels pertinents en matière de trajectoire de vie. En effet, ces réseaux concernent de nombreuses thématiques de la vie, pouvant aussi bien être consacrés à la vie personnelle, professionnelle, ou aux loisirs. Un moissonnage des informations contenues dans ces réseaux, effectué avec l'accord des individus pourrait donc être une source de données importante. L'hypothèse formulée ici est que si l'on est capable d'extraire et de formater automatiquement des éléments de trajectoires contenus dans les réseaux sociaux pour les rendre compatibles avec notre modèle, alors on réduit d'autant la charge incombant à un contributeur. Il est cependant ici évident que les questions de fiabilité des données moissonnées, ainsi que de confidentialité et de protection de la vie privée ne pourront être ignorées. Notre approche, qui permet d'offrir des requêtes d'insertion dans le *Triple Store* pour tous modèles de trajectoires de vie issus de notre patron de conception, pourra être utile lors de la conception d'une approche de moissonnage de ces données.

## 2.5 Exploitation des trajectoires

**Opérateurs d'exploitation des trajectoires** D'autres opérateurs génériques d'exploitation des trajectoires doivent être proposés. À ce titre un travail exploratoire doit être mené pour définir les opérateurs les plus adaptés pour l'étude des trajectoires. Pour définir les opérateurs temporels, on pourra se baser notamment sur l'algèbre de [Allen, 1984].

**Opérateurs multi-granulaires générique** Des opérateurs permettant d'augmenter ou de diminuer le niveau de granularité des résultats d'une requête sur les trajectoires doivent être proposés. La définition de prédicats génériques indiquant dans une requête un lien entre deux ressources de type *plus fin que*, ou *plus grossier que*, doit par exemple être envisagée.

Dans le cadre de notre approche, il est possible de mettre à profit le fait qu'elle comprend une phase de configuration de notre architecture qui pourrait également être utilisée pour paramétrer ces opérateurs. Les travaux de [Silva et al., 2015] qui permettent de modéliser la notion de granularité à la fois dans le temps, l'espace ou l'information thématique constituent des outils conceptuels intéressants dans une perspective de définition de tels opérateurs.



# Annexes



# Le patron de conception

---

Patron de conception de trajectoire de vie décrit en utilisant la sérialisation Turtle du langage RDF.

```
@prefix geosparql: <http://schemas.opengis.net/
  geosparql/1.0/geosparql_vocab_all.rdf#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
  ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
.
@prefix LTO: <http://www.semanticweb.org/david/
  ontologies/2016/9/LTO#> .
@prefix OWL-Time: <https://www.w3.org/TR/owl-time/#> .
@prefix foaf: <http://xmlns.com/foaf/spec/#> .
@prefix schema: <http://schema.org/#> .
@prefix geofla: <http://data.ign.fr/def/geofla#> .

<http://www.semanticweb.org/david/ontologies/2016/9/
  LTO>
  a owl:Ontology .

OWL-Time:Interval
  a owl:Class ;
  rdfs:subClassOf OWL-Time:Temporal-Thing .

LTO:hasTrajectory
  a owl:ObjectProperty ;
  rdfs:domain schema:Person ;
  rdfs:range LTO:Trajectory .

LTO:Episode
```

```
    a owl:Class ;
    rdfs:label "Episode" .

geosparql:SpatialObject
  a owl:Class .

LTO:endWithEvent
  a owl:ObjectProperty ;
  rdfs:domain LTO:Episode ;
  rdfs:range LTO:Event .

OWL-Time:Instant
  a owl:Class ;
  rdfs:subClassOf OWL-Time:Temporal-Thing .

geosparql:geometry
  a owl:Class ;
  rdfs:subClassOf geosparql:SpatialObject .

LTO:validyPeriod
  a owl:ObjectProperty ;
  rdfs:domain LTO:Episode ;
  rdfs:range OWL-Time:Interval .

LTO:Event
  a owl:Class .

LTO:SpatialEpisode
  a owl:Class ;
  rdfs:subClassOf LTO:Episode .

LTO:hasLocation
  a owl:ObjectProperty ;
  rdfs:domain LTO:SpatialEpisode ;
  rdfs:range geosparql:geometry ;
  rdfs:subPropertyOf owl:topObjectProperty .

LTO:hasEpisode
  a owl:ObjectProperty ;
```

---

```
rdfs:domain LTO:Trajectory ;
rdfs:range LTO:Episode .

LTO:hasSpatialEpisode
  a owl:ObjectProperty ;
  rdfs:domain LTO:SpatialTrajectory ;
  rdfs:range LTO:SpatialEpisode .

geofla:UniteAdministrative
  a owl:Class .

LTO:Trajectory
  a owl:Class .

LTO:SpatialTrajectory
  a owl:Class ;
  rdfs:subClassOf LTO:Trajectory .

LTO:hasEvent
  a owl:ObjectProperty ;
  rdfs:domain LTO:Trajectory ;
  rdfs:range LTO:Event .

schema:Person
  a owl:Class ;
  owl:equivalentClass foaf:Person .

LTO:atTime
  a owl:ObjectProperty ;
  rdfs:domain LTO:Event ;
  rdfs:range OWL-Time:Instant .

LTO:isIn
  a owl:ObjectProperty ;
  rdfs:domain LTO:SpatialEpisode ;
  rdfs:range geofla:UniteAdministrative .

LTO:startWithEvent
  a owl:ObjectProperty ;
```

```
rdfs:domain LTO:Episode ;  
rdfs:range LTO:Event ;  
rdfs:subPropertyOf owl:topObjectProperty .
```

```
OWL-Time:Temporal-Thing  
a owl:Class .
```

# Le modèle de facteur explicatif

---

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
.
```

```
@prefix lto: <http://www.purl.org/lto#> .
```

```
@prefix dcterms: <http://purl.org/dc/terms/> .
```

```
#Classes
```

```
lto:Event
```

```
    dcterms:description "This class characterizes a  
        life event." ;  
    a owl:Class .
```

```
lto:Factor
```

```
    dcterms:description "This class characterizes  
        an explanatory factor of a life event." ;  
    a owl:Class .
```

```
lto:TrajectoryEventFactor
```

```
    dcterms:description "This class characterizes an  
        explanatory factor of a life event that is  
        related to another event of the life  
        trajectory." ;  
    a owl:Class ;  
    rdfs:subClassOf lto:Factor .
```

```
lto:TrajectoryStateFactor
```

```
    dcterms:description "This class characterizes an  
        explanatory factor of a life event that is
```

```
related to a state of the life trajectory." ;  
  a owl:Class ;  
  rdfs:subClassOf lto:Factor .
```

```
lto:NetworkEventFactor  
  dcterms:description "This class characterizes an  
  explanatory factor of a life event that is  
  related to an event of another life trajectory  
  ." ;  
  a owl:Class ;  
  rdfs:subClassOf lto:TrajectoryEventFactor .
```

```
lto:InternalEventFactor  
  dcterms:description "This class characterizes an  
  explanatory factor of a life event that is  
  related to an event of the same life  
  trajectory." ;  
  a owl:Class ;  
  rdfs:subClassOf lto:TrajectoryEventFactor .
```

#Object properties

```
lto:hasFactor  
  a owl:ObjectProperty ;  
  dcterms:description "Relates a life event to  
  an explanatory factor." ;  
  rdfs:domain lto:Event ;  
  rdfs:range lto:Factor .
```

```
lto:hasExplainingEvent  
  a owl:ObjectProperty ;  
  dcterms:description "Relates an explanatory  
  factor to its explaining event." ;  
  rdfs:domain lto:TrajectoryEventFactor ;  
  rdfs:range lto:Event .
```

```
lto:hasExplainingStatement  
  a owl:ObjectProperty ;  
  dcterms:description "Relates an explanatory
```

```
        factor to its explaining statement" ;  
        rdfs:domain lto:TrajectoryStateFactor .
```

```
#Datatype properties
```

```
lto:hasExplanation  
    a owl:DatatypeProperty ;  
    dcterms:description "Relates an explanatory  
        factors to its explanation" ;  
    rdfs:domain lto:Factor ;  
    rdfs:range xsd:string .
```

```
lto:hasWeight  
    a owl:DatatypeProperty ;  
    dcterms:description "Relates an explanatory  
        factors to its weight" ;  
    rdfs:domain lto:Factor ;  
    rdfs:range xsd:integer .
```



# L'implémentation des fonctions SPARQL avec l'API Jena

## 1 Fonction ltofn :before

```

public class BeforePropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node episode1, final Node
                predicate, final Node episode2, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX time: <https://www.w3.org/TR/owl-time/#> "
                    + "ASK {<"+episode1.getURI()+> lto:validityPeriod ?epi1ValidityPeriod . "
                    + "?epi1ValidityPeriod time:hasEnd ?epi1DateEnd ."
                    + "?epi1DateEnd time:inXSDDate ?epi1DateEndXSD ."
                    + "<"+episode2.getURI()+> lto:validityPeriod ?epi2ValidityPeriod ."
                    + "?epi2ValidityPeriod time:hasBeginning ?epi2DateBegin ."
                    + "?epi2DateBegin time:inXSDDate ?epi2DateBeginXSD ."
                    + "FILTER (?epi1DateEndXSD <= ?epi2DateBeginXSD)}";
                Query query = QueryFactory.create(queryString);
                Boolean b;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, model)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        }
    }
}

```

Figure 3.1 – Capture d'écran Eclipse de la Classe BeforePropertyFunction

## 2 Fonction ltofn :after

```

public class AfterPropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node episode1, final Node
                predicate, final Node episode2, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX time: <https://www.w3.org/TR/owl-time/#> "
                    + "ASK {<"+episode1.getURI()+> lto:validityPeriod ?epi1ValidityPeriod ."
                    + "?epi1ValidityPeriod time:hasEnd ?epi1DateEnd ."
                    + "?epi1DateEnd time:inXSDDate ?epi1DateEndXSD ."
                    + "<"+episode2.getURI() + "> lto:validityPeriod ?epi2ValidityPeriod ."
                    + "?epi2ValidityPeriod time:hasBeginning ?epi2DateBegin ."
                    + "?epi2DateBegin time:inXSDDate ?epi2DateBeginXSD ."
                    + "FILTER (?epi1DateEndXSD <= ?epi2DateBeginXSD)}";

                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, model)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        }
    }
}

```

Figure 3.2 – Capture d'écran Eclipse de la Classe AfterPropertyFunction

### 3 Fonction ltofn :overlaps

```

public class OverlapsPropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node episode1, final Node
                predicate, final Node episode2, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX time: <https://www.w3.org/TR/owl-time/#> "
                    + "ASK {< + episode1.getURI() + "> lto:validityPeriod ?epi1ValidityPeriod ."
                    + "?epi1ValidityPeriod time:hasBeginning ?epi1DateBegin ."
                    + "?epi1DateBegin time:inXSDDate ?epi1DateBeginXSD ."
                    + "?epi1ValidityPeriod time:hasEnd ?epi1DateEnd ."
                    + "?epi1DateEnd time:inXSDDate ?epi1DateEndXSD ."
                    + "<" + episode2.getURI() + "> lto:validityPeriod ?epi2ValidityPeriod ."
                    + "?epi2ValidityPeriod time:hasBeginning ?epi2DateBegin ."
                    + "?epi2DateBegin time:inXSDDate ?epi2DateBeginXSD ."
                    + "?epi2ValidityPeriod time:hasEnd ?epi2DateEnd ."
                    + "?epi2DateEnd time:inXSDDate ?epi2DateEndXSD ."
                    + "FILTER (?epi1DateBeginXSD < ?epi2DateEndXSD "
                    + "    && ?epi1DateEndXSD < ?epi2DateBeginXSD)}";
                Query query = QueryFactory.create(queryString);
                Boolean b;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, model)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

**Figure 3.3** – Capture d'écran Eclipse de la Classe OverlapsPropertyFunction

## 4 Fonction ltofn :isFactorOutFrom

```

public class IsFactorOutFromPropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node factor, final Node
                predicate, final Node status, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // This function need an inference model to take into account lto:endsWith child properties
                InfModel modelInf = ModelFactory.createRDFSModel(model);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX owl: <http://www.w3.org/2002/07/owl#> "
                    + "ASK {?episode ?statusProperty <" + status.getURI() + ">. "
                    + "?episode lto:endsWith ?event . "
                    + "?event lto:hasFactor <" + factor.getURI() + "> . }" ;

                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, modelInf)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

Figure 3.4 – Capture d'écran Eclipse de la Classe IsFactorOutFromPropertyFunction

## 5 Fonction ltofn :isFactorEnterIn

```

public class IsFactorEnterInPropertyFunction implements PropertyFunctionFactory {}
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node factor, final Node predicate,
                final Node status, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // This function need an inference model to take into account lto:startWith child properties
                InfModel modelInf = ModelFactory.createRDFSModel(model);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX owl: <http://www.w3.org/2002/07/owl#> "
                    + "ASK {?episode ?statusProperty <" + status.getURI() + ">. "
                    + "?episode lto:startWith ?event . "
                    + "?event lto:hasFactor <" + factor.getURI() + ">. }";
                Query query = QueryFactory.create(queryString);
                Boolean b;

                try (QueryExecution qexec = QueryExecutionFactory.create(query, modelInf)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

**Figure 3.5** – Capture d'écran Eclipse de la Classe IsFactorEnterInPropertyFunction

## 6 Fonction ltofn :isFactorOfExplainingEventType

```

public class IsFactorOfExplainingEventTypePropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node factor, final Node
                predicate, final Node eventType, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "ASK {< + factor.getURI() + "> lto:hasExplainingEvent ?event . "
                    + "?event lto:hasType < + eventType.getURI() + "> . }" ;
                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, model)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

**Figure 3.6** – Capture d'écran Eclipse de la Classe IsFactorOfExplainingEventPropertyFunction

## 7 Fonction ltofn :isFactorOfExplainedEventType

```

public class IsFactorOfExplainedEventTypePropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node factor, final Node
                predicate, final Node eventType, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();
                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
                    + "ASK { ?event lto:hasFactor <" + factor.getURI() + "> . "
                    + "?event lto:hasType <" + eventType.getURI() + "> . }" ;

                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, model)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

Figure 3.7 – Capture d'écran Eclipse de la Classe IsFactorOfExplainedEventTypePropertyFunction

## 8 Fonction ltofn :IsFactorOfExplainedTrajectory- Type

```

public class IsFactorOfExplainedTrajectoryTypePropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node factor, final Node
                predicate, final Node trajectoryType, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // Inference model to take into account lto:hasEvent child properties
                InfModel modelInf = ModelFactory.createRDFSModel(model);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
                    + "ASK { ?event lto:hasFactor <" + factor.getURI() + "> . "
                    + " ?trajectory lto:hasEvent ?event . "
                    + "?trajectory rdfs:type <" + trajectoryType.getURI() + "> . }" ;
                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, modelInf)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

**Figure 3.8** – Capture d'écran Eclipse de la Classe IsFactorOfExplained-TrajectoryTypePropertyFunction

## 9 Fonction ltofn :IsFactorOfExplainingTrajectoryType

```

public class IsFactorOfExplainingTrajectoryTypePropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node factor, final Node
                predicate, final Node trajectoryType, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // Inference model to take into account lto:hasEvent child properties
                InfModel modelInf = ModelFactory.createRDFSModel(model);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
                    + "ASK { < + factor.getURI() +> lto:hasExplainingEvent ?event . "
                    + " ?trajectory lto:hasEvent ?event . "
                    + "?trajectory rdfs:type < + trajectoryType.getURI() + "> . }" ;

                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, modelInf)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

**Figure 3.9** – Capture d'écran Eclipse de la Classe IsFactorOfExplaining-TrajectoryTypePropertyFunction

## 10 Fonction ltofn :episodeBeforeFactor

```

public class EpisodeBeforeFactorPropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node episode, final Node
                predicate, final Node factor, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // This function need an inference model to take into account lto:endWith child properties
                InfModel modelInf = ModelFactory.createRDFSModel(model);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "ASK { <" + episode.getURI() + "> lto:endWith ?event . "
                    + "?event lto:hasFactor <" + factor.getURI() + "> .}" ;
                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, modelInf)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

**Figure 3.10** – Capture d'écran Eclipse de la Classe EpisodeBeforeFactor-PropertyFunction

## 11 Fonction ltofn :episodeAfterFactor

```

public class EpisodeAfterFactorPropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node episode, final Node
                predicate, final Node factor, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // This function need an inference model to take into account lto:startWith child properties
                InfModel modelInf = ModelFactory.createRDFSModel(model);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "ASK { <" + episode.getURI() + "> lto:startWith ?event . "
                    + "?event lto:hasFactor <" + factor.getURI() + "> .}" ;
                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, modelInf)) {
                    b = qexec.execAsk();
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        };
    }
}

```

**Figure 3.11** – Capture d'écran Eclipse de la Classe EpisodeAfterFactor-PropertyFunction

## 12 Fonction ltofn :episodeAtFactorDate

```

public class EpisodeAtFactorDatePropertyFunction implements PropertyFunctionFactory {
    @Override
    public PropertyFunction create(final String uri)
    {
        return new PFuncSimple()
        {
            @Override
            public QueryIterator execEvaluated(final Binding parent, final Node episode, final Node
                predicate, final Node factor, final ExecutionContext execCxt)
            {
                //Retrieve current Graph.
                Graph graph = execCxt.getActiveGraph();

                //Wrap Graph in a Model.
                Model model = ModelFactory.createModelForGraph(graph);

                // Test if the required pattern is in the graph
                String queryString = "PREFIX lto: <http://www.purl.org/lto#> "
                    + "PREFIX time: <https://www.w3.org/TR/owl-time/#> "
                    + "ASK { ?event lto:hasFactor <" + factor.getURI() + "> . "
                    + "?event lto:atTime ?eventDate . "
                    + "?eventDate time:inXSDDate ?eventDateXSD . "
                    + "<" + episode.getURI() + "> lto:validityPeriod ?epiValidityPeriod . "
                    + "?epiValidityPeriod time:hasBeginning ?epiDateBegin . "
                    + "?epiValidityPeriod time:hasEnd ?epiDateEnd . "
                    + "?epiDateBegin time:inXSDDate ?epiDateBeginXSD . "
                    + "?epiDateEnd time:inXSDDate ?epiDateEndXSD . "
                    + "FILTER (?eventDateXSD >= ?epiDateBeginXSD && ?eventDateXSD <= ?epiDateEndXSD) }" ;
                Query query = QueryFactory.create(queryString) ;
                Boolean b ;
                try (QueryExecution qexec = QueryExecutionFactory.create(query, model)) {
                    b = qexec.execAsk();
                    System.out.println(b);
                    qexec.close();
                }
                // If yes return the parent binding
                if(b) {
                    return QueryIterSingleton.create(parent, execCxt);
                } else { // If no don't return the binding
                    return QueryIterNullIterator.create(execCxt);
                }
            }
        }
    }
}

```

Figure 3.12 – Capture d'écran Eclipse de la Classe EpisodeAtFactorDatePropertyFunction

# L'ontologie LTO pour l'étude du choix résidentiels

---

```
@prefix geosparql: <http://schemas.opengis.net/
  geosparql/1.0/geosparql_vocab_all.rdf#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
  ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
.
@prefix lto: <http://www.purl.org/lto#> .
@prefix time: <https://www.w3.org/TR/owl-time/#> .
@prefix foaf: <http://xmlns.com/foaf/spec/#> .
@prefix schema: <http://schema.org/#> .
@prefix geofla: <http://data.ign.fr/def/geofla#> .
@prefix skos: <http://www.w3.org/TR/skos-reference#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

#Ontology

<http://www.purl.org/lto#>
  a owl:Ontology ;
  dcterms:title "Life trajectory ontology" ;
  dcterms:description "This ontology of life
  trajectory allow to characterize
  residential choices over time and space and
  according to multiple thematic viewpoints
  (residential, professional, family)" ;
  owl:versionInfo "Version 0.1 - 2017-03-20" .

#Classes
```

```
lto:ResidentialTrajectory
    dcterms:description "This class characterizes
        the residential trajectory of an
        individual." ;
    a owl:Class .
```

```
lto:ProfessionalTrajectory
    dcterms:description "This class characterizes
        a professional episode of the life
        trajectory." ;
    a owl:Class .
```

```
lto:FamilyTrajectory
    dcterms:description "This class characterizes
        the family trajectory of an individual." ;
    a owl:Class .
```

```
lto:Episode
    a owl:Class ;
    dcterms:description "This class characterizes
        an episode of the life trajectory. An
        episode is a stable state of the life
        trajectory seen from a given viewpoint." ;
    rdfs:label "Episode" .
```

```
lto:SpatialEpisode
    dcterms:description "This class characterizes
        a spatial episode of the life trajectory.
        A spatial episode has one attribute
        location" ;
    a owl:Class ;
    rdfs:subClassOf lto:Episode .
```

```
lto:ResidentialEpisode
    dcterms:description "This class characterizes
        a residential episode of the life
        trajectory." ;
    a owl:Class ;
```

---

```
    rdfs:subClassOf lto:Episode , lto:SpatialEpisode
    .

lto:ProfessionalEpisode
    dcterms:description "This class characterizes
        a professional episode of the life
        trajectory." ;
    a owl:Class ;
    rdfs:subClassOf lto:Episode , lto:SpatialEpisode
    .

lto:FamilyEpisode
    dcterms:description "This class characterizes
        a family episode of the life trajectory."
    ;
    a owl:Class ;
    rdfs:subClassOf lto:Episode .

lto:Event
    dcterms:description "This class characterizes
        an event of the life trajectory." ;
    a owl:Class .

lto:ResidentialEvent
    dcterms:description "This class characterizes
        a residential event of the life trajectory
        ." ;
    a owl:Class ;
    rdfs:subClassOf lto:Event .

lto:ProfessionalEvent
    a owl:Class ;
    dcterms:description "This class characterizes
        a professional event of the life
        trajectory." ;
    rdfs:subClassOf lto:Event .

lto:FamilyEvent
    dcterms:description "This class characterizes
```

```
        a family event of the life trajectory." ;  
a owl:Class ;  
rdfs:subClassOf lto:Event .
```

```
lto:Factor
```

```
    dcterms:description "This class characterizes  
        an explanatory factor of a life event." ;  
    a owl:Class .
```

```
lto:TrajectoryEventFactor
```

```
    dcterms:description "This class characterizes an  
        explanatory factor of a life event that is  
        related to another event of the life  
        trajectory." ;  
    a owl:Class ;  
    rdfs:subClassOf lto:Factor .
```

```
lto:TrajectoryStateFactor
```

```
    dcterms:description "This class characterizes an  
        explanatory factor of a life event that is  
        related to a state of the life trajectory." ;  
    a owl:Class ;  
    rdfs:subClassOf lto:Factor .
```

```
lto:NetworkEventFactor
```

```
    dcterms:description "This class characterizes an  
        explanatory factor of a life event that is  
        related to an event of another life trajectory  
        ." ;  
    a owl:Class ;  
    rdfs:subClassOf lto:TrajectoryEventFactor .
```

```
lto:InternalEventFactor
```

```
    dcterms:description "This class characterizes an  
        explanatory factor of a life event that is  
        related to an event of the same life  
        trajectory." ;  
    a owl:Class ;  
    rdfs:subClassOf lto:TrajectoryEventFactor .
```

---

```
#properties with internal classes as range

lto:hasResidentialTrajectory
  a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:hasTrajectory ;
    dcterms:description "Residential trajectory a
      of a person." ;
    rdfs:domain schema:Person ;
    rdfs:range lto:ResidentialTrajectory .

lto:hasFamilyTrajectory
  a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:hasTrajectory ;
    dcterms:description "Family trajectory a of a
      person." ;
    rdfs:domain schema:Person ;
    rdfs:range lto:FamilyTrajectory .

lto:hasProfessionalTrajectory
  a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:hasTrajectory ;
    dcterms:description "Professional trajectory
      a of a person." ;
    rdfs:domain schema:Person ;
    rdfs:range lto:ProfessionalTrajectory .

lto:hasResidentialEpisode
  a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:hasEpisode ;
    dcterms:description "Residential episode that
      belongs to a residential trajectory." ;
    rdfs:domain lto:ResidentialTrajectory ;
    rdfs:range lto:ResidentialEpisode .

lto:hasProfessionalEpisode
  a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:hasEpisode ;
```

```

    dcterms:description "Professional episode
        that belongs to a professional trajectory."
    ;
    rdfs:domain lto:ProfessionalTrajectory ;
    rdfs:range lto:ProfessionalEpisode .

lto:hasFamilyEpisode
    a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:hasEpisode ;
    dcterms:description "Family episode that
        belongs to a family trajectory." ;
    rdfs:domain lto:FamilyTrajectory ;
    rdfs:range lto:FamilyEpisode .

lto:hasEvent
    a owl:ObjectProperty ;
    dcterms:description "Relates a trajectory to
        an event." ;
    rdfs:domain lto:Trajectory ;
    rdfs:range lto:Event .

lto:hasResidentialEvent
    a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:hasEvent ;
    dcterms:description "Residential event that
        belongs to a residential trajectory." ;
    rdfs:domain lto:ResidentialTrajectory ;
    rdfs:range lto:ResidentialEvent .

lto:hasProfessionalEvent
    a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:hasEvent ;
    dcterms:description "Professional event that
        belongs to a professional trajectory." ;
    rdfs:domain lto:ProfessionalTrajectory ;
    rdfs:range lto:ProfessionalEvent .

lto:hasFamilyEvent
    a owl:ObjectProperty ;
```

---

```
    rdfs:subPropertyOf lto:hasEvent ;
    dcterms:description "Family event that
        belongs to a family trajectory." ;
    rdfs:domain lto:FamilyTrajectory ;
    rdfs:range lto:FamilyEvent .
```

```
lto:startWith
```

```
    a owl:ObjectProperty ;
    dcterms:description "Relates an episode to
        its starting event." ;
    rdfs:domain lto:Episode ;
    rdfs:range lto:Event .
```

```
lto:startWithResidentialEvent
```

```
    a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:startWith ;
    dcterms:description "Residential event that
        starts an episode." ;
    rdfs:domain lto:ResidentialEpisode ;
    rdfs:range lto:ResidentialEvent .
```

```
lto:startWithProfessionalEvent
```

```
    a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:startWith ;
    dcterms:description "Professional event that
        starts an episode." ;
    rdfs:domain lto:ProfessionalEpisode ;
    rdfs:range lto:ProfessionalEvent .
```

```
lto:startWithFamilyEvent
```

```
    a owl:ObjectProperty ;
    rdfs:subPropertyOf lto:startWith ;
    dcterms:description "Family event that starts
        an episode." ;
    rdfs:domain lto:FamilyEpisode ;
    rdfs:range lto:FamilyEvent .
```

```
lto:endWith
```

```
    a owl:ObjectProperty ;
```

```
dcterms:description "Relates an episode to  
  its ending event." ;  
rdfs:domain lto:Episode ;  
rdfs:range lto:Event .
```

```
lto:endWithResidentialEvent  
  a owl:ObjectProperty ;  
  rdfs:subPropertyOf lto:endWith ;  
  dcterms:description "Residential event that  
    ends an episode." ;  
  rdfs:domain lto:ResidentialEpisode ;  
  rdfs:range lto:ResidentialEvent .
```

```
lto:endWithProfessionalEvent  
  a owl:ObjectProperty ;  
  rdfs:subPropertyOf lto:endWith ;  
  dcterms:description "Professional event that  
    ends an episode." ;  
  rdfs:domain lto:ProfessionalEpisode ;  
  rdfs:range lto:ProfessionalEvent .
```

```
lto:endWithFamilyEvent  
  a owl:ObjectProperty ;  
  rdfs:subPropertyOf lto:endWith ;  
  dcterms:description "Family event that ends  
    an episode." ;  
  rdfs:domain lto:FamilyEpisode ;  
  rdfs:range lto:FamilyEvent .
```

```
lto:hasFactor  
  a owl:ObjectProperty ;  
  dcterms:description "Relates a life event to  
    an explanatory factor." ;  
  rdfs:domain lto:Event ;  
  rdfs:range lto:Factor .
```

```
lto:hasExplainingEvent  
  a owl:ObjectProperty ;  
  dcterms:description "Relates an explanatory
```

---

```
        factor to its explaining event." ;
        rdfs:domain lto:TrajectoryEventFactor ;
        rdfs:range lto:Event .

lto:hasExplainingStatement
    a owl:ObjectProperty ;
    dcterms:description "Relates an explanatory
        factor to its explaining statement" ;
    rdfs:domain lto:TrajectoryStateFactor .

#properties with external classes as range

lto:atTime
    a owl:ObjectProperty ;
    dcterms:description "Instant of a life event"
        ;
    rdfs:domain lto:Event ;
    rdfs:range time:Instant .

lto:isIn
    a owl:ObjectProperty ;
    dcterms:description "Administrative Area to
        which belongs a geometry." ;
    rdfs:domain lto:SpatialEpisode ;
    rdfs:range geofla:UniteAdministrative .

lto:hasLocation
    a owl:ObjectProperty ;
    dcterms:description "Location of the spatial
        episode." ;
    rdfs:domain lto:SpatialEpisode ;
    rdfs:range geosparql:geometry .

lto:validityPeriod
    a owl:ObjectProperty ;
    dcterms:description "Validity period of the
        episode" ;
    rdfs:domain lto:Episode ;
    rdfs:range time:Interval .
```

```
#properties with skos concept as range
```

```
lto:hasType
```

```
    a owl:ObjectProperty ;  
    rdfs:domain lto:Event ;  
    rdfs:range lto:EventTypeConcept .
```

```
lto:residentialStatus
```

```
    a owl:ObjectProperty ;  
    rdfs:domain lto:ResidentialEpisode ;  
    rdfs:range lto:ResidentialStatusConcept .
```

```
lto:housingType
```

```
    a owl:DatatypeProperty ;  
    rdfs:domain lto:ResidentialEpisode ;  
    rdfs:range lto:HousingTypeConcept .
```

```
lto:rentOrLoan
```

```
    a owl:ObjectProperty ;  
    rdfs:domain lto:ResidentialEpisode ;  
    rdfs:range lto:RentOrLoanRangeConcept .
```

```
lto:occupationnalStatus
```

```
    dcterms:description "Occupationnal status of  
        the individual belongs during an episode" ;  
    a owl:ObjectProperty ;  
    rdfs:domain lto:ProfessionalEpisode ;  
    rdfs:range lto:OccupationnalStatusConcept .
```

```
lto:typeOfContract
```

```
    a owl:ObjectProperty ;  
    rdfs:domain lto:ProfessionalEpisode ;  
    rdfs:range lto:TypeOfContractConcept .
```

```
lto:salary
```

```
    a owl:ObjectProperty ;  
    rdfs:domain lto:ProfessionalEpisode ;  
    rdfs:range lto:SalaryRangeConcept .
```

---

```
lto:maritalStatus
    a owl:DatatypeProperty ;
    rdfs:domain lto:FamilyEpisode ;
    rdfs:range lto:MaritalStatusConcept .

#Datatype properties

lto:hasExplanation
    a owl:DatatypeProperty ;
    dcterms:description "Explanation of an
        explanatory factor" ;
    rdfs:domain lto:Factor ;
    rdfs:range xsd:string .

lto:hasWeight
    a owl:DatatypeProperty ;
    dcterms:description "Weight of an explanatory
        factor" ;
    rdfs:domain lto:Factor ;
    rdfs:range xsd:integer .

lto:numberOfRoom
    a owl:DatatypeProperty ;
    dcterms:description "Number of room of the
        housing during a residential episode" ;
    rdfs:domain lto:ResidentialEpisode ;
    rdfs:range xsd:integer .

lto:numberOfChildren
    a owl:DatatypeProperty ;
    dcterms:description "Number of children
        during a family episode" ;
    rdfs:domain lto:FamilyEpisode ;
    rdfs:range xsd:integer .
```



# Vocabulaires contrôlés utilisés

---

## 1 Vocabulaire contrôlé des types de logement

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
```

```
@prefix lto: <http://www.purl.org/lto#>.
```

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
.
```

```
lto:residentialStatusConceptScheme
```

```
  a skos:ConceptScheme ;
```

```
  skos:prefLabel "Vocabulaire contrôlé des  
  statuts résidentiels" .
```

```
lto:residentialStatusConcept rdfs:subClassOf skos:  
  Concept ;
```

```
  skos:prefLabel "Ensemble des statuts ré  
  sidentiels" .
```

```
lto:residentialStatusOwner
```

```
  a lto:residentialStatusConcept ;
```

```
  skos:prefLabel "Propriétaire" ;
```

```
  skos:inScheme lto:residentialStatus .
```

```
lto:residentialStatusPartOwner
```

```
  a lto:residentialStatusConcept ;
```

```
  skos:prefLabel "Co-propriétaire" ;
```

```
  skos:inScheme lto:residentialStatus .
```

```
lto:residentialStatusAccessing
```

```
  a lto:residentialStatusConcept ;
```

```
  skos:prefLabel "Propriétaire accédant" ;
```

```
  skos:inScheme lto:residentialStatus .
```

```
lto:residentialStatusTenant
  a lto:residentialStatusConcept ;
  skos:prefLabel "Locataire" ;
  skos:inScheme lto:residentialStatus .
```

```
lto:residentialStatusCoTenant
  a lto:residentialStatusConcept ;
  skos:prefLabel "Co-Locataire" ;
  skos:inScheme lto:residentialStatus .
```

```
lto:residentialStatusFreeHosting
  a lto:residentialStatusConcept ;
  skos:prefLabel "Hébergé-e gratuitement" ;
  skos:inScheme lto:residentialStatus .
```

## 2 Vocabulaire contrôlé des types de logement

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix lto: <http://www.purl.org/lto#>.
```

```
lto:typeOfHousingConceptScheme
  a skos:ConceptScheme ;
  skos:prefLabel "Vocabulaire contrôlé des types
    de logement" .
```

```
lto:typeOfHousingConcept
  rdfs:subClassOf skos:Concept ;
  skos:prefLabel "Ensemble des statuts ré
    sidentiels" .
```

```
lto:apartment
  a skos:typeOfHousingConcept ;
  skos:prefLabel "Appartement" ;
  skos:inScheme lto:typeOfHousing .
```

```
lto:house
  a skos:typeOfHousingConcept ;
  skos:prefLabel "Maison" ;
```

```
skos:inScheme lto:typeOfHousing .

lto:studio
  a skos:typeOfHousingConcept ;
  skos:prefLabel "Studio" ;
  skos:inScheme lto:typeOfHousing .

lto:HLM
  a skos:typeOfHousingConcept ;
  skos:prefLabel "HLM" ;
  skos:inScheme lto:typeOfHousing .

lto:othersTypeOfHousing a skos:typeOfHousingConcept ;
  skos:prefLabel "Autre type de logement" ;
  skos:inScheme lto:typeOfHousing .
```

### 3 Vocabulaire contrôlé des intervalles de loyers

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix lto: <http://www.purl.org/lto#>.
```

```
lto:rentOrLoanRangeConceptScheme
  a skos:ConceptScheme ;
  skos:prefLabel "Vocabulaire contrôlé caractérisant des intervalles correspondant au montant du loyer ou du prêt" .

lto:rentOrLoanRangeConcept rdfs:subClassOf skos:Concept ;
  skos:prefLabel "Ensemble des intervalles" .

lto:rentOrLoanRangeLessthan400
  a lto:rentOrLoanRangeConcept ;
  skos:prefLabel "Moins de 400" ;
  skos:inScheme lto:rentOrLoanRangeConceptScheme
  .

lto:rentOrLoanRange400to600
  a lto:rentOrLoanRangeConcept ;
```

```
skos:prefLabel "Entre 400 et 600" ;
skos:inScheme lto:rentOrLoanRangeConceptScheme
```

.

```
lto:rentOrLoanRange600to800
```

```
  a lto:rentOrLoanRangeConcept ;
  skos:prefLabel "Entre 600 et 800" ;
  skos:inScheme lto:rentOrLoanRangeConceptScheme
```

.

```
lto:rentOrLoanRange800to1000
```

```
  a lto:rentOrLoanRangeConcept ;
  skos:prefLabel "Entre 800 et 1000" ;
  skos:inScheme lto:rentOrLoanRangeConceptScheme
```

.

```
lto:rentOrLoanRange1000to1200
```

```
  a lto:rentOrLoanRangeConcept ;
  skos:prefLabel "Entre 1000 et 1200" ;
  skos:inScheme lto:rentOrLoanRangeConceptScheme
```

.

```
lto:rentOrLoanRangeMorethan1200
```

```
  a lto:rentOrLoanRangeConcept ;
  skos:prefLabel "Plus de 1200" ;
  skos:inScheme lto:rentOrLoanRangeConceptScheme
```

.

## 4 Vocabulaire contrôlé des types de contrat

```
#types of contract
```

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
```

```
@prefix lto: <http://www.purl.org/lto#>.
```

```
lto:typeOfContractConceptScheme
```

```
  a skos:ConceptScheme ;
  rdfs:label "Vocabulaire contrôlé des types de
  contrat" .
```

```
lto:typeOfContractConcept
    rdfs:subClassOf skos:Concept ;
    rdfs:label "Ensemble des types de contrat" .

lto:fixedTermContractFullTime
    a skos:typeOfContractConcept ;
    rdfs:label "CDD temps plein" ;
    skos:topConceptOf lto:
        typeOfContractConceptScheme .

lto:fixedTermContractPartTime
    a skos:typeOfContractConcept ;
    rdfs:label "CDD temps partiel" ;
    skos:topConceptOf lto:
        typeOfContractConceptScheme .

lto:permanentContractFullTime
    a skos:typeOfContractConcept ;
    rdfs:label "CDI temps plein" ;
    skos:topConceptOf lto:
        typeOfContractConceptScheme .

lto:permanentContractPartTime
    a skos:typeOfContractConcept ;
    rdfs:label "CDI temps partiel" ;
    skos:topConceptOf lto:
        typeOfContractConceptScheme .

lto:Interim
    a skos:typeOfContractConcept ;
    rdfs:label "Interim" ;
    skos:topConceptOf lto:
        typeOfContractConceptScheme .

lto:workStudyContract a skos:typeOfContractConcept ;
    rdfs:label "Alternance" ;
    skos:topConceptOf lto:
        typeOfContractConceptScheme .
```

## 5 Vocabulaire contrôlé des intervalles de salaires

@prefix skos: <http://www.w3.org/2004/02/skos/core#>.

@prefix lto: <http://www.purl.org/lto#>.

lto:salaryRangeConceptScheme

  a skos:ConceptScheme ;

  skos:prefLabel "Vocabulaire contrôlé caractérisant des intervalles correspondant au montant du salaire annuel" .

lto:salaryRangeConcept rdfs:subClassOf skos:Concept ;

  skos:prefLabel "Ensemble des intervalles" .

lto:salaryRangeLessthan10000

  a lto:salaryRangeConcept ;

  skos:prefLabel "Moins de 10000" ;

  skos:inScheme lto:salaryRangeConceptScheme .

lto:salaryRange10000to20000

  a lto:salaryRangeConcept ;

  skos:prefLabel "Entre 10000 et 20000" ;

  skos:inScheme lto:salaryRangeConceptScheme .

lto:salaryRange20000to30000

  a lto:salaryRangeConcept ;

  skos:prefLabel "Entre 20000 et 30000" ;

  skos:inScheme lto:salaryRangeConceptScheme .

lto:salaryRange30000to40000

  a lto:salaryRangeConcept ;

  skos:prefLabel "Entre 30000 et 40000" ;

  skos:inScheme lto:salaryRangeConceptScheme .

lto:salaryRange40000to50000

  a lto:salaryRangeConcept ;

  skos:prefLabel "Entre 40000 et 50000" ;

  skos:inScheme lto:salaryRangeConceptScheme .

```
lto:salaryRangeMorethan50000
  a lto:salaryRangeConcept ;
  skos:prefLabel "Plus 50000" ;
  skos:inScheme lto:salaryRangeConceptScheme .
```

## 6 Vocabulaire contrôlé des statuts maritiaux

```
lto:maritalStatusConceptScheme
  a skos:ConceptScheme ;
  rdfs:label "Vocabulaire controlé des statuts
  maritiaux" .
```

```
lto:maritalStatusConcept
  a skos:Concept ;
  rdfs:label "Ensemble des statuts maritiaux" .
```

```
lto:fixedTermContractFullTime
  a skos:maritalStatusConcept ;
  rdfs:label "CDD temps plein" ;
  skos:topConceptOf lto:
  maritalStatusConceptScheme .
```

```
lto:fixedTermContractPartTime
  a skos:maritalStatusConcept ;
  rdfs:label "CDD temps partiel" ;
  skos:topConceptOf lto:
  maritalStatusConceptScheme .
```

```
lto:permanentContractFullTime
  a skos:maritalStatusConcept ;
  rdfs:label "CDI temps plein" ;
  skos:topConceptOf lto:
  maritalStatusConceptScheme .
```

```
lto:permanentContractPartTime
  a skos:maritalStatusConcept ;
  rdfs:label "CDI temps partiel" ;
  skos:topConceptOf lto:
  maritalStatusConceptScheme .
```

```
lto:Interim
    a skos:maritalStatusConcept ;
    rdfs:label "Interim" ;
    skos:topConceptOf lto:
        maritalStatusConceptScheme .

lto:workStudyContract a skos:maritalStatusConcept ;
    rdfs:label "Alternance" ;
    skos:topConceptOf lto:
        maritalStatusConceptScheme .
```

## 7 Vocabulaire contrôlé des événements résidentiels

```
#residential event
```

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix lto: <http://www.purl.org/lto#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
.
```

```
lto:residentialEventConceptScheme
    a skos:ConceptScheme ;
    skos:prefLabel "Vocabulaire contrôlé des évé-
        nements résidentiel" .
```

```
lto:EventTypeConcept
    rdfs:subClassOf skos:Concept .
```

```
lto:residentialEventMove
    a skos:EventTypeConcept ;
    skos:prefLabel "Déménagement" ;
    skos:inScheme lto:professionalEvent .
```

```
lto:residentialEventMovein
    a skos:EventTypeConcept ;
    skos:prefLabel "Départ d'un logement" ;
    skos:inScheme lto:professionalEvent .
```

```
lto:residentialEventMoveout
  a skos:EventTypeConcept ;
  skos:prefLabel "Emménagement dans un logement"
  ;
  skos:inScheme lto:professionalEvent .
```

```
lto:residentialEventOther
  a skos:EventTypeConcept ;
  skos:prefLabel "Autre" ;
  skos:inScheme lto:professionalEvent .
```

## 8 Vocabulaire contrôlé des événements professionnel

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix lto: <http://www.purl.org/lto#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
.
```

```
lto:professionalEventConceptScheme
  a skos:ConceptScheme ;
  skos:prefLabel "Vocabulaire contrôlé des événements
  professionnels" .
```

```
lto:EventTypeConcept
  rdfs:subClassOf skos:Concept .
```

```
lto:professionalEventChange
  a skos:EventTypeConcept ;
  skos:prefLabel "Changement d'emploi" ;
  skos:inScheme lto:
  professionalEventConceptScheme .
```

```
lto:professionalEventTraining
  a skos:EventTypeConcept ;
  skos:prefLabel "Formation continue" ;
  skos:inScheme lto:
  professionalEventConceptScheme .
```

```
lto:professionalEventStudies
  a skos:EventTypeConcept ;
  skos:prefLabel "Reprise d'études" ;
  skos:inScheme lto:
    professionalEventConceptScheme .

lto:professionalEventStartingBusiness
  a skos:EventTypeConcept ;
  skos:prefLabel "Création d'entreprise" ;
  skos:inScheme lto:
    professionalEventConceptScheme .

lto:professionalEventPromotion
  a skos:EventTypeConcept ;
  skos:prefLabel "Promotion" ;
  skos:inScheme lto:
    professionalEventConceptScheme .

lto:professionalEventResignation
  a skos:EventTypeConcept ;
  skos:prefLabel "Démission" ;
  skos:inScheme lto:
    professionalEventConceptScheme .

lto:professionalEventTermination
  a skos:EventTypeConcept ;
  skos:prefLabel "Licenciment" ;
  skos:inScheme lto:
    professionalEventConceptScheme .

lto:professionalEventOther
  a skos:EventTypeConcept ;
  skos:prefLabel "Autre" ;
  skos:inScheme lto:
    professionalEventConceptScheme .
```

## 9 Vocabulaire contrôlé des événements familiaux

@prefix skos: <<http://www.w3.org/2004/02/skos/core#>>.

@prefix lto: <http://www.purl.org/lto#>.

```
lto:familialEvent
  a skos:ConceptScheme ;
  skos:prefLabel "Vocabulaire controlé des événé
    ments familiaux" .
```

```
lto:familialEventCouple
  a skos:Concept ;
  skos:prefLabel "Mise en couple" ;
  skos:inScheme lto:familialEvent .
```

```
lto:familialEventWedding
  a skos:Concept ;
  skos:prefLabel "Mariage" ;
  skos:inScheme lto:familialEvent .
```

```
lto:familialEventBirth
  a skos:Concept ;
  skos:prefLabel "Naissance" ;
  skos:inScheme lto:familialEvent .
```

```
lto:familialEventBreakUp
  a skos:Concept ;
  skos:prefLabel "Séparation" ;
  skos:inScheme lto:familialEvent .
```

```
lto:familialEventWidowhood
  a skos:Concept ;
  skos:prefLabel "Veuvage" ;
  skos:inScheme lto:familialEvent .
```

```
lto:familialEventOther
  a skos:Concept ;
  skos:prefLabel "Autre" ;
  skos:inScheme lto:familialEvent .
```



# Patrons de conception de requête d'insertion des données

---

## 1 Requête d'insertion d'un épisode d'une trajectoire géographique

```
INSERT DATA
{
  <URI-Trajectory> lto:hasEpisode <URI-Episode>
  .
  <URI-Episode> a lto:Episode ;
    lto:startWithEvent <URI-Event1> ;
    lto:endWithEvent <URI-Event2> ;
    lto:validityPeriod <Period> ;
    lto:isIn <URI-AdministrativeUnit> ;
    lto:hasLocation <URI-geometry> .
}
```



ANNEXE G

# **Interfaces de collecte de trajectoire de vie**

---

# 1 Collecte des épisodes et événements professionnels

Enquête - Trajectoire de vie
🔴

Événement professionnel

Date

📅

Précision
Jour ▾

Événement

Faites votre choix...
📌

Situation professionnelle suite à l'événement

Profession

📌

- Ingénieurs et cadres d'étude du bâtiment et des travaux publics
- Ingénieurs et cadres technico-commerciaux en bâtiment, travaux publics
- Ingénieurs et cadres spécialistes des télécommunications
- Ingénieurs, cadres de chantier et conducteurs de travaux (cadres) du bâtiment et des travaux publics
- Ingénieurs et cadres technico-commerciaux en matériel électrique ou électronique professionnelle
- Ingénieurs et cadres d'étude et d'exploitation de l'agriculture, la pêche, les eaux et forêts
- Ingénieurs et cadres d'étude, recherche et développement des Industries de transformation
- Ingénieurs et cadres de fabrication des autres Industries (imprimerie, matériaux souples, etc.)
- Ingénieurs et cadres de fabrication des Industries de transformation (agroalimentaire, chimie)
- Ingénieurs et cadres de fabrication en matériel électrique, électronique

Revenu disponible

Faites votre choix...
📌

Type de contrat

Faites votre choix...
📌

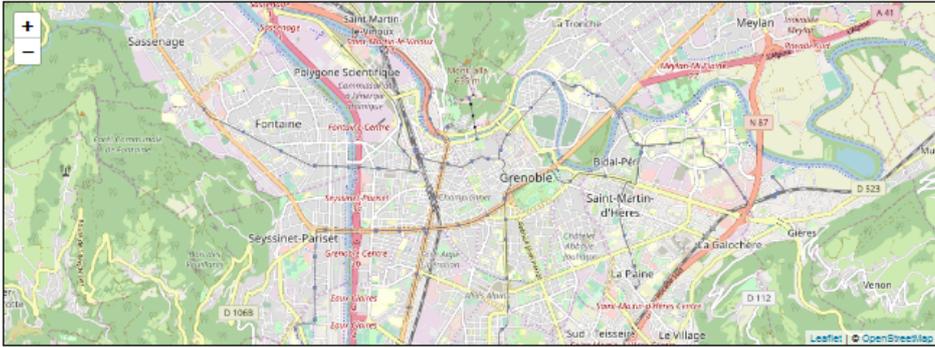
Localisation

🔍
📌

Code postal

Ville

Pays



Annuler
Valider

**Figure 7.1** – Interface de collecte d'un épisode professionnel

La figure 7.1 présente l'interface de collecte d'événement et d'épisodes professionnels. Les différents champs du formulaire correspondent aux attributs présentés dans l'ontologie de trajectoire de vie. De la même manière que pour l'interface de collecte d'épisodes résidentiels, lorsqu'un attribut est décrit

par un vocabulaire contrôlé le contributeur est amené à choisir une valeur dans ce vocabulaire. Dans le cas du choix de la profession, le vocabulaire de l'INSEE étant trop grand pour être affiché intégralement, nous avons mis en place une fonctionnalité d'auto-complétion, qui permet au contributeur de choisir simplement une des valeurs de ce vocabulaire.

## 2 Collecte des épisodes et événements familiaux

Enquête - Trajectoire de vie

Événement familial

Date

Événement

Faites votre choix...

Précision Jour

Situation familiale suite à l'événement

Statut matrimonial

Nombre d'enfants

Mode de garde

Faites votre choix...

0

Faites votre choix...

Annuler Valider

Figure 7.2 – Interface de collecte d'un épisode familial



# Bibliographie

- James F. Allen. Towards a general theory of action and time. *Artificial intelligence*, 23(2) :123–154, 1984. 37, 103, 145
- Luis Otavio Alvares, Vania Bogorny, Bart Kuijpers, Jose Antonio Fernandes de Macedo, Bart Moelans, and Alejandro Vaisman. A model for enriching trajectories with semantic geographical information. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, page 22. ACM, 2007. 17, 21, 23
- Gennady Andrienko, Natalia Andrienko, and Marco Heurich. An event-based conceptual model for context-aware movement analysis. *International Journal of Geographical Information Science*, 25(9) :1347–1370, 2011. 3, 22, 29, 30, 31, 58
- Jean-Yves Authier, Jennifer Bidet, Anaïs Collet, Pierre Gilbert, and H el ene Steinmetz.  Etat des lieux sur les trajectoires r esidentielles. 2012. 11, 14, 78
- Miriam Baglioni, Jos e Macedo, Chiara Renso, and Monica Wachowicz. An ontology-based approach for the semantic modelling and reasoning on trajectories. *Advances in Conceptual Modeling—Challenges and Opportunities*, pages 344–353, 2008. 3, 39, 40, 58, 59
- Arnaud Banos, Sonia Chardonnel, Christophe Lang, Nicolas Marilleau, and Thomas Th evenin. Une approche multi-agents de la ville en mouvement. In *Colloque SMAGEL : Colloque Smaget*, pages 1–17, 2005. 24
- S Becker Howard. Outsiders.  etudes de sociologie de la d eviance. * ditions M etail e*, 1985. 14, 78
- Yvan Bedard and Eveline Bernier. Supporting multiple representations with spatial view management and the concept of” vuel. In *Joint Workshop on Multi-Scale Representations of Spatial Data, ISPRS WG IV/3, ICA Commission on Map Generalisation*, pages 7–8, 2002. 45
- Camille Bernard, Marl ene Villanova-Oliver, J er ome Gensel, and Hy Dao. Modeling changes in territorial partitions overtime : ontologies tsn and tsn-change. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 866–875. ACM, 2018. 47

- Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5) :34–43, 2001. 32
- Claudio Bettini, X Sean Wang, and Sushil Jajodia. A general framework for time granularity and its application to temporal reasoning. *Annals of mathematics and artificial intelligence*, 22(1-2) :29–58, 1998. 47
- Vania Bogorny, Chiara Renso, Artur Ribeiro de Aquino, Fernando de Lucca Siqueira, and Luis Otavio Alvares. CONSTAnT - A Conceptual Data Model for Semantic Trajectories of Moving Objects : CONSTAnT - A Conceptual Data Model for Semantic Trajectories. *Transactions in GIS*, 18(1) : 66–88, February 2014. ISSN 13611682. doi : 10.1111/tgis.12011. URL <http://doi.wiley.com/10.1111/tgis.12011>. 3, 22, 49, 50, 58, 59, 61
- Karla AV Borges, Clodoveu A Davis, and Alberto HF Laender. Omt-g : an object-oriented data model for geographic applications. *GeoInformatica*, 5 (3) :221–260, 2001. 45
- Christophe Claramunt and Marius Thériault. Managing time in GIS an event-oriented approach. In *Recent Advances in Temporal Databases*, pages 23–42. Springer, 1995. URL [http://link.springer.com/chapter/10.1007/978-1-4471-3033-8\\_2](http://link.springer.com/chapter/10.1007/978-1-4471-3033-8_2). 52
- Christophe Claramunt, Marius Thériault, and Christine Parent. A qualitative representation of evolving spatial entities in two-dimensional topological spaces, 1997a. 36
- Christophe Claramunt, Marius Thériault, and Christine Parent. *A qualitative representation of evolving spatial entities in two-dimensional topological spaces*. London, Taylor & Francis, 1997b. 52
- Christophe Claramunt, Christine Parent, and Marius Thériault. Design patterns for spatio-temporal processes. In *Data Mining and Reverse Engineering*, pages 455–475. Springer, 1998a. URL [http://link.springer.com/chapter/10.1007/978-0-387-35300-5\\_19](http://link.springer.com/chapter/10.1007/978-0-387-35300-5_19). 53
- Christophe Claramunt, Christine Parent, and Marius Thériault. Design patterns for spatio-temporal processes. In *Data Mining and Reverse Engineering*, pages 455–475. Springer, 1998b. 52

- Olga Marino Drews. *Raisonnement classificatoire dans une représentation à objets multi-points de vue*. PhD thesis, Université Joseph-Fourier-Grenoble I, 1993. 72
- Jérôme Euzenat. *Représentations de connaissance : de l'approximation à la confrontation*. PhD thesis, Université Joseph-Fourier-Grenoble I, 1999. 44
- Carlos Andres Ferrero, Luis Otávio Alvares, and Vania Bogorny. Multiple aspect trajectory data analysis : research challenges and opportunities. In *GeoInfo*, pages 56–67, 2016. 43, 49, 50, 61, 72
- Renato Fileto, Cleto May, Chiara Renso, Nikos Pelekis, Douglas Klein, and Yannis Theodoridis. The baquara2 knowledge-based framework for semantic enrichment and analysis of movement data. *Data & Knowledge Engineering*, 98 :104–122, 2015. 3, 51, 52, 55, 61
- Ali Frihida, Donia Zheni, Henda Ben Ghezala, and Christophe Claramunt. Modeling trajectories : A spatio-temporal data type approach. In *Database and Expert Systems Application, 2009. DEXA'09. 20th International Workshop on*, pages 447–451. IEEE, 2009. 29, 31
- Michael F. Goodchild. Citizens as sensors : the world of volunteered geography. *GeoJournal*, 69(4) :211–221, August 2007. ISSN 0343-2521, 1572-9893. doi : 10.1007/s10708-007-9111-y. URL <http://link.springer.com/article/10.1007/s10708-007-9111-y>. 11
- Yves Grafmeyer and Jean-Yves Authier. *Sociologie urbaine*. Armand colin, 2011. 14
- Ralf Hartmut Güting and Markus Schneider. *Moving objects databases*. Elsevier, 2005. 9
- Valentine Hélaridot. Parcours professionnels et histoires de santé : une analyse sous l'angle des bifurcations. *Cahiers internationaux de sociologie*, (1) : 59–83, 2006. 9
- Torsten Hägerstraand. What about people in regional science? *Papers in regional science*, 24(1) :7–24, 1970. 22, 24, 26
- Jerry R Hobbs and Feng Pan. Time ontology in owl. *W3C working draft*, 27 : 133, 2006a. 3, 36, 37
- Jerry R Hobbs and Feng Pan. Time ontology in OWL. *W3C working draft*, 27 :133, 2006b. 40

- Kathleen Hornsby and Max J Egenhofer. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence*, 36 (1) :177–194, 2002. 15, 46
- Yingjie Hu, Krzysztof Janowicz, David Carral, Simon Scheider, Werner Kuhn, Gary Berg-Cross, Pascal Hitzler, Mike Dean, and Dave Kolas. A geontology design pattern for semantic trajectories. In *Spatial Information Theory*, pages 438–456. Springer, 2013. URL [http://link.springer.com/chapter/10.1007/978-3-319-01790-7\\_24](http://link.springer.com/chapter/10.1007/978-3-319-01790-7_24). 3, 40, 41, 58, 59
- Christian S Jensen, Curtis E Dyreson, Michael Böhlen, James Clifford, Ramez Elmasri, Shashi K Gadia, Fabio Grandi, Pat Hayes, Sushil Jajodia, Wolfgang Käfer, and others. The consensus glossary of temporal database concepts. In *Temporal Databases : Research and Practice*, pages 367–405. Springer, 1998. 15
- Meihan Jin and Christophe Claramunt. A semantic model for human mobility in an urban region. *Journal on Data Semantics*, 7(3) :171–187, 2018. 64
- A. Krisnadhi and Pascal Hitzler. A core pattern for events. In *Proceedings of the Workshop on Ontology and Semantic Web Patterns (7th edition), Kobe, Japan, 2017*. 3, 38
- Adila A Krisnadhi, Pascal Hitzler, and Krzysztof Janowicz. A spatiotemporal extent pattern based on semantic trajectories. In *Workshop on Ontology and Semantic Web Patterns, 7th edn.–WOP, 2016*. 41
- Patrick Laube and Ross S Purves. An approach to evaluating motion pattern detection techniques in spatio-temporal data. *Computers, environment and urban systems*, 30(3) :347–374, 2006. 9
- Christine Lelévrier. Les trajectoires résidentielles des ménages dans les opérations de rénovation urbaine en Île-de-France. October 2014. URL <http://www.ecoledurenouvellementurbain.com/wp-content/uploads/sites/8/2014/03/DREIF-LELEVRIER-trajectoires-relogement-fev2009.pdf>. 10
- Xiang Li, Christophe Claramunt, Cyril Ray, and Hui Lin. A semantic-based approach to the representation of network-constrained trajectory data. In *Progress in Spatial Data Handling*, pages 451–464. Springer, 2006. 22

- François Madoré and Jean Pihan. Habitat et stratégies résidentielles. *Texte introductif au séminaire ESO, MSH Ange-Guépin*, 2003. 10
- Olga Mariño. *Raisonnement classificatoire dans une représentation à objets multi-points de vue*. PhD thesis, Grenoble 1, 1993. 44
- DM Mark. Geospatial lifelines. Dagstuhl Seminar Report. Integrating Spatial and Temporal Databases 228. 1998. 24
- David Noël, Marlène Villanova-Oliver, and Jérôme Gensel. Recitoire : a tool for qualitative surveys involving citizens in urban planning projects. In *17th AGILE International Conference on Geographic Information Science*, page x, 2014. 11
- David Noel, Marlène Villanova-Oliver, Jérôme Gensel, and Pierre Le Quéau. Design patterns for modelling life trajectories in the semantic web. In *International Symposium on Web and Wireless Geographical Information Systems*, pages 51–65. Springer, 2017. 88
- S Openshaw and P Taylor. A million or so correlation coefficients : three experiments on the modifiable areal unit problem. 127-144. *Statistical Applications in the Spatial Sciences*. Pion, London, 1979. 46
- Christine Parent, Stefano Spaccapietra, and Esteban Zimányi. *Conceptual modeling for traditional and spatio-temporal applications : The MADS approach*. Springer Science & Business Media, 2006a. 27, 45, 47
- Christine Parent, Stefano Spaccapietra, and Esteban Zimányi. The MurMur project : Modeling and querying multi-representation spatio-temporal databases. *Information Systems*, 31(8) :733–769, 2006b. URL <http://www.sciencedirect.com/science/article/pii/S0306437905000165>. 45
- Jean-Claude Passeron. Biographies, flux, itinéraires, trajectoires. *Revue française de sociologie*, 31(1) :3–22, 1990. 132
- Donna J Peuquet. It's about time : A conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers*, 84(3) :441–461, 1994. 16, 52, 53, 142

- Donna J. Peuquet and Niu Duan. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International journal of geographical information systems*, 9(1) :7–24, January 1995. ISSN 0269-3798. doi : 10.1080/02693799508902022. URL <http://www.tandfonline.com/doi/abs/10.1080/02693799508902022>. 52
- Christine Plumejeaud. *Modèles et méthodes pour l'information spatio-temporelle évolutive*. PhD thesis, Université Grenoble Alpes, 2011. 47
- Christine Plumejeaud, Hélène Mathian, Jérôme Gensel, and Claude Grasland. Spatio-temporal analysis of territorial changes from a multi-scale perspective. *International Journal of Geographical Information Science*, 25(10) : 1597–1612, 2011. 52
- Yves Raimond and Samer Abdallah. The event ontology. Technical report, 2007. 3, 39
- Nicolas Robette, Catherine Bonvalet, and Arnaud Bringé. Les trajectoires géographiques des franciliens depuis leur départ de chez les parents. *Catherine Bonvalet et Éva Lelièvre, De la famille à l'entourage, Paris, INED*, pages 177–201, 2012. 10, 125
- Max Schmachtenberg, Christian Bizer, Anja Jentzsch, and Richard Cyganiak. Linking open data cloud diagram 2014. URL <http://lod-cloud.net/>. [Cited at pages ix and 20], 2014. 3, 33, 34
- Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. *IEEE intelligent systems*, 21(3) :96–101, 2006. 31, 32
- Christophe Sidonie, Paule-Annick Davoine, Francis Jambon, Isabelle André-Poyaud, Sonia Chardonnel, Céline Lutoff, and Ahmed Lbath. Acquisition de connaissances sur les déplacements quotidiens des individus dans un contexte de risques naturels/protocoles d'enquête à l'aide de technologies mobiles. In *Spatial Analysis and GEOmatics (SAGEO'2010)*, 2010. 9
- Ricardo Almeida Silva, João Moura Pires, Maribel Yasmina Santos, and Rui Leal. Aggregating Spatio-temporal Phenomena at Multiple Levels of Detail. In *AGILE 2015*, pages 291–308. Springer, 2015. URL [http://link.springer.com/chapter/10.1007/978-3-319-16787-9\\_17](http://link.springer.com/chapter/10.1007/978-3-319-16787-9_17). 3, 15, 48, 49, 145
- Gaurav Sinha and David M. Mark. Measuring similarity between geospatial lifelines in studies of environmental health. *Journal of Geographical*

- Systems*, 7(1) :115–136, May 2005. ISSN 1435-5930, 1435-5949. doi : 10.1007/s10109-005-0153-8. URL <http://link.springer.com/10.1007/s10109-005-0153-8>. 3, 24, 25
- Chaxel Sophie, Fiorelli Cécile, and Moity-Maïzi Pascale. Les récits de vie : outils pour la compréhension et catalyseurs pour l'action. 2014. 132, 134
- Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, Jose Antonio de Macedo, Fabio Porto, and Christelle Vangenot. A conceptual view on trajectories. *Data & knowledge engineering*, 65(1) :126–146, 2008. URL <http://www.sciencedirect.com/science/article/pii/S0169023X07002078>. 3, 11, 13, 22, 26, 27, 28, 31, 40, 41, 58, 59, 61, 62
- Marius Thériault, A.-M. Seguin, Yanick Aubé, and Paul Y. Villeneuve. A spatio-temporal data model for analysing personal biographies. pages 410–418. IEEE, 1999. URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=795202](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=795202). 3, 4, 11, 12, 53, 60, 62, 73
- Marius Thériault, Christophe Claramunt, Anne-Marie Séguin, and Paul Villeneuve. Temporal GIS and Statistical Modelling of Personal Lifelines. In Dr Dianne E. Richardson and Professor Dr Peter van Oosterom, editors, *Advances in Spatial Data Handling*, pages 433–449. Springer Berlin Heidelberg, January 2002. ISBN 978-3-642-62859-7, 978-3-642-56094-1. URL [http://link.springer.com/chapter/10.1007/978-3-642-56094-1\\_32](http://link.springer.com/chapter/10.1007/978-3-642-56094-1_32). 3, 11, 49, 52, 54, 55, 58, 60, 61, 62
- Raphaël Troncy, Ryan Shaw, and Lynda Hardman. Lode : une ontologie pour représenter des événements dans le web de données. In *21es Journées Francophones d'Ingénierie des Connaissances (IC 2010)*, <http://www.ic2010.mines-ales.fr/>, pages 69–80. Ecole des Mines d'Alès, 2010. 39
- JD Ullman. *Principles of Database and Knowledge-Base Systems, Vol I*. Computer Science Press, 1988. 44, 61
- Marie-Hélène Vandersmissen, Anne-Marie Séguin, Marius Thériault, and Christophe Claramunt. Modeling propensity to move after job change using event history analysis and temporal GIS. *Journal of Geographical Systems*, 11(1) :37–65, March 2009. ISSN 1435-5930, 1435-5949. doi : 10.1007/s10109-009-0076-x. URL <http://link.springer.com/10.1007/s10109-009-0076-x>. 54

- Christelle Vangenot, Christine Parent, and Stefano Spaccapietra. Modelling and manipulating multiple representations of spatial data. In *Advances in Spatial Data Handling*, pages 81–93. Springer, 2002. 45, 47, 77, 95
- Robert Weibel and Geoffrey Dutton. Generalising spatial data and dealing with multiple representations. *Geographical information systems*, 1 :125–155, 1999. 47
- Zhixian Yan and Stefano Spaccapietra. Towards Semantic Trajectory Data Analysis : A Conceptual and Computational Approach. In *VLDB PhD Workshop*, 2009. 3, 23
- Nancy J Yattaw. Conceptualizing space and time : a classification of geographic movement. *Cartography and Geographic Information Science*, 26(2) : 85–98, 1999a. 11
- Nancy J. Yattaw. Conceptualizing Space and Time : A Classification of Geographic Movement. *Cartography and Geographic Information Science*, 26 (2) :85–98, January 1999b. ISSN 1523-0406, 1545-0465. doi : 10.1559/152304099782330734. URL <http://www.tandfonline.com/doi/abs/10.1559/152304099782330734>. 15, 25
- Donia Zheni, Ali Frihida, Henda Ben Ghezala, and Christophe Claramunt. A semantic approach for the modeling of trajectories in space and time. In *Advances in Conceptual Modeling-Challenging Perspectives*, pages 347–356. Springer, 2009. URL [http://link.springer.com/chapter/10.1007/978-3-642-04947-7\\_41](http://link.springer.com/chapter/10.1007/978-3-642-04947-7_41). 22