THESE DE DOCTORAT DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE PREPAREE A I3MAINZ

Ecole Doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques (SPIM)

Doctorat d'Informatique

Par

Claire PRUDHOMME

Knowledge-driven multi-agent simulation engineering for assessing the effectiveness of disaster management plans

Ingénierie de simulation multi-agents conduite par la connaissance pour évaluer l'efficacité des plans de gestion de catastrophes

Thèse présentée et soutenue à Dijon, le 17/12/2020

Composition du Jury:

M. GENTIL Christian	Professeur à l'Université Bourgogne Franche-Comté	Président
Mme TEISSEIRE Maguelonne	Professeur à l'Institut INRAE	Rapportrice
M. BILL Ralf	Professeur à l'Université Rostock	Rapporteur
Mme VIDAL Maria-Esther	Professeur à l'Université Gottfried Wilhelm Leibniz de Hanovre	Examinatrice
M. CRUZ Christophe	Professeur à l'Université Bourgogne Franche-Comté	Directeur
Mme ROXIN Ana	Maître de conférences HDR à l'Université Bourgogne Franche-Comté	Co-directrice
M. BOOCHS Frank	Professeur à l'Université des Sciences appliquées de Mayence	Co-directeur
M. SCHAFFERT Markus	Professeur à l'Université des Sciences appliquées de Mayence	Invité

Titre: Ingénierie de simulation multi-agents conduite par la connaissance pour évaluer l'efficacité des plans de gestion de catastrophes

Mots clés: Technologies du Web sémantique, Simulation multi-agent, Architecture conduite par les connaissances, Gestion de catastrophe

Résumé: La protection des personnes contre les catastrophes est une tâche importante des gouvernements et des experts, qui s'effectue en définissant des plans de gestion de catastrophes. Les stratégies de réponse en cas de catastrophe visent à réduire le nombre de victimes et l'impact économique. La sélection du plan de réponse le plus approprié pour des situations de catastrophe spécifiques nécessite une évaluation de ces plans. Toutefois, cette évaluation est limitée par le coût élevé des exercices et la spécificité des modèles de simulation existants. L'approche proposée dans cette thèse combine les techniques du Web Sémantique et la simulation multi-agents pour évaluer les plans de réponse de gestion de catastrophes. Elle est composée de quatre étapes : (1) la modélisation des connaissances en matière de gestion des catastrophes, (2) la modélisation des simulations, (3) la conception des simulations, et (4) l'analyse des résultats des simulations à partir de regroupements. Tout d'abord, les connaissances explicites et les données d'experts sont utilisées pour créer un modèle de connaissances pour la gestion des catastrophes. Deuxièmement, les modèles de simulation sont conçus sur la base du modèle de connaissances. Troisièmement, la programmation générative est utilisée pour la conception des simulations. Enfin, les résultats des simulations sont utilisés pour calculer l'efficacité du plan pour chaque simulation. Le regroupement par apprentissage non supervisé permet d'identifier le contexte d'application lié à l'efficacité calculée. L'efficacité et le contexte d'application associé enrichissent le modèle de connaissance initial. Cette approche a été appliquée à une étude de cas basée sur le plan français NOVI dans la ville de Montbard, en France.

Title: Knowledge-driven multi-agent simulation engineering for assessing the effectiveness of disaster management plans

Keywords: Semantic Web Technologies, Multi-agent Simulation, Knowledge-Driven Architecture, Disaster Management

Abstract: Protecting humans from disasters has been an active mission of governments and experts through the definition of disaster management plans. Defining disaster response strategies is crucial in order to reduce the number of victims and the economic impact. In order to select which response plan is best suited to a specific disaster situation, these plans must be evaluated. However, such evaluation is limited by the high cost of exercises and the specificity of existing simulation models. The approach defended in this thesis combines techniques from Semantic Web and multi-agent simulation to evaluate disaster management response plans. It is composed of four steps : (1) modeling disaster management knowledge, (2) modeling simulations, (3) designing simulations, and (4) analyzing simulation results based on clustering. First, explicit expert knowledge and data is used to create a knowledge model for disaster management. Second, simulation models are conceived based on the knowledge model. Thirdly, generative programming is used for simulation design. Finally, simulation results are used to calculate the plan's effectiveness for each simulation. Unsupervised learning clustering identifies the application context related to the calculated effectiveness. The effectiveness and the associated application context enrich the initial knowledge model. This approach was applied to a case study based on the French NOVI plan in the city of Montbard, France.

Titel: Wissensgesteuerte Simulationen mittels Multiagentenkonzepten für die Bewertung der Wirksamkeit von Plänen zur Katastrophenbewältigung

Schlüsselwörter: Semantic-Web-Technologien, Multiagentensimulation, wissensbasierte Architektur, Katastrophenmanagement

Kurzfassung: Der Schutz der Menschen vor Katastrophen ist eine wichtige Aufgabe von Regierungen und Experten und wird durch die Definition von Katastrophenmanagementplänen wahrgenommen. Für Katastrophenreaktionsstrategien ist dabei von entscheidender Bedeutung die Zahl der Opfer und die wirtschaftlichen Auswirkungen zu verringern. Die Auswahl des für bestimmte Katastrophensituationen bestmöglich geeigneten Reaktionsplans erfordert eine Evaluation dieser Pläne. Eine solche Evaluierung ist jedoch durch die hohen Kosten praktischer Übungen und die Besonderheit der bestehenden Simulationsmodelle begrenzt. Der in dieser Arbeit vorgeschlagene Ansatz kombiniert Techniken des Semantic Web und der Multi-Agenten-Simulation, um Reaktionspläne für das Katastrophenmanagement zu evaluieren. Er besteht aus vier Schritten: (1) Modellierung von Wissen aus dem Katastrophenmanagement, (2) Modellierung von Simulationen, (3) Erstellung von Simulationen und (4) Analyse der Simulationsergebnisse mittels Clustering. Zunächst werden explizites Expertenwissen und Daten verwendet, um ein Wissensmodell für das Katastrophenmanagement zu erstellen. Danach werden Simulationsmodelle auf der Grundlage des Wissensmodells konzipiert. Anschließend wird generative Programmierung zur Erstellung von Simulationen verwendet. Abschließend werden die Simulationsergebnisse verwendet, um die Wirksamkeit des Plans für jede Simulation zu berechnen. Durch unbeaufsichtigtes Lernclustering wird der Anwendungskontext in Bezug auf die berechnete Wirksamkeit identifiziert. Die Wirksamkeit und der damit verbundene Anwendungskontext bereichern das anfängliche Wissensmodell. Dieser Ansatz wurde auf eine Fallstudie auf der Grundlage des französischen NOVI-Plans in der Stadt Montbard, Frankreich, angewandt.

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon

Acknowledgment

I want to thank my director Professor Christophe Cruz and my co-directors Dr. Ana Roxin and Professor Frank Boochs, who supported and wisely advised me throughout my thesis. I want to thank the jury members for their work and their effort that allow me to defend the Ph.D. thesis. I want to express my gratitude to Prof. Teisseire and Prof. Bill, who have reviewed the manuscript in a difficult situation with challenging time constraints.

This thesis has been financed by the German Ministry of Education and Research through the project called "Semantic Geographic Information System" (SemGIS) [grants: 03FH032IX4]. I want to thank my colleagues Dr. Ashish Karmacharya, Dr. Falk Würriehausen, and Timo Homburg for the collaborative work made together. This project has allowed me to exchange and learn from different partners. I want to thank Dr. Stefan Jäger and his team in the Geomer company for their warm welcome during an internship that allowed a better mutual understanding. I want to thank Dr. Marlene Willkomm of the flood protection center in Cologne (Germany) for her explanations and data sharing related to Cologne's flood management. I want to thank the Institute of the Fire Brigade North Rhine-Westphalia and the Unna district's crisis team in Germany for allowing me to participate in a crisis exercise and better understand their expertise domain. I want to thank Laurence Porte, Mayor of Montbard, for her cooperation and explanations about crisis management in a french city.

The Ph.D. thesis was carried out at the institute i3mainz, Institute for Spatial Information and Surveying Technology of the University of Applied Sciences Mainz. This institute's competencies are in the scope of Spatial Information Technology in the Humanities with application fields such as cultural heritage, geology, environment (e.g. disaster management, bioaerosol), and energy. I want to thank the i3mainz institute members, particularly Dr. Jean-Jacques Ponciano and Songül Polat, for their working ethic and their friendship that has allowed me to thrive within this institute.

"Give me six hours to chop down a tree
and I will spend the first four
sharpening the axe."

—Abraham Lincoln

Dédicacé à ma famille de coeur pour son soutien et son amour, avec une pensée toute particulière pour Denise GAVEAU

Knowledge-driven multi-agent simulation engineering for assessing the effectiveness of disaster management plans

Claire PRUDHOMME

17th December 2020

Contents

1 Résumé détaillé de la thèse en français / Detailed summary of the thesis in French 9

1.1 Contexte et motivation 11

1.1.1 Évaluation de plans de gestion des catastrophes 11

1.1.2 Application pratique d'un plan 12

1.1.3 Défis et objectifs 14

1.2 Énoncé du problème 16

1.2.1 Enjeux 16

1.2.2 Besoins 19

1.3 Approche proposée 20

1.3.1 Approches proposées pour répondre aux besoins 20

1.3.2 Approche pour évaluer l'efficacité d'un plan d'intervention 21

1.4 Contributions 22

1.4.1 Contribution pour la communauté de gestion des catastrophes 22

1.4.2 Contribution pour le Web sémantique 23

1.4.3 Contribution pour l'ingénierie de simulation multi-agents 24

1.5 Vue d'ensemble de la thèse 24

2 Introduction 27

2.1 Context and motivation 29

2.1.1 Disaster management plan assessment 29

2.1.2 Practical application of a plan 30

2.1.3 Challenges and objectives 32

2.2 Problem statement 34

2.2.1 Issues 34

2.2.2 Requirements 36

2.3 Proposed approaches 37
2.3.1 Proposed approaches to meet requirements 38
2.3.2 Approach for assessing the effectiveness of a response plan 39
2.4 Contributions 39
2.4.1 Contribution to the disaster management community 39
2.4.2 Contribution to the Semantic Web 40
2.4.3 Contribution to multi-agent simulation engineering 41
2.4.4 Publications 41
2.5 Thesis overview 43

3 Related work **47**
3.1 Disaster management 48
3.1.1 A cycle of four stages 48
3.1.2 Preparedness 51
3.1.3 Simulation approaches supporting preparedness 58
3.2 Multi-agent simulation 62
3.2.1 Multi-agent simulation engineering 63
3.2.2 Multi-agent simulation model components 65
3.2.3 Multi-agent simulation platforms 72
3.2.4 Discussion 76
3.3 Knowledge engineering 78
3.3.1 Knowledge representation for disaster management 82
3.3.2 Knowledge extraction and integration 93
3.4 Discussion 97

4 Conceptual approach **101**
4.1 Requirements overview 102
4.1.1 Lack of change evaluation in the DM plan description 104
4.1.2 Lack of adaptability to simulate and assess the diversity of plans representation 104
4.1.3 Lack of adaptability and reusability of MAS components for disaster management 106
4.1.4 Lack of expressive plans and scenarios 106

4.2 Proposed approaches to meet requirements . . . 107
4.2.1 Assessing a plan's effectiveness and its applicability . . . 109
4.2.2 Adapting the design of simulation experiments . . . 110
4.2.3 Adaptability and reusability of MAS components for disaster management . . . 117
4.2.4 Disaster management knowledge modeling for plan assessment . . . 117
4.3 Approach for assessing the effectiveness of a response plan . . . 124
4.3.1 Modeling of the studied system . . . 126
4.3.2 Simulation modeling . . . 127
4.3.3 Programming simulations and executing the related experiments . . . 129
4.3.4 Assessing the response plan . . . 129
4.4 Summary . . . 130

5 Architecture 133

5.1 Presentation of the architecture . . . 134
5.2 Multi-agent simulation platform . . . 137
5.2.1 GAMA platform and its programmed model . . . 138
5.2.2 MAS components base . . . 139
5.3 Knowledge base . . . 141
5.3.1 SemDM ontology: disaster management plans' assessment . . . 142
5.3.2 SemMAS ontology: multi-agent simulation . . . 146
5.4 Client . . . 153
5.4.1 Disaster management system's modeling . . . 154
5.4.2 Event modeling . . . 155
5.4.3 Geospatial modeling . . . 155
5.5 Processing server . . . 156
5.5.1 Disaster management system's modeling . . . 157
5.5.2 Plan assessment . . . 157
5.6 Summary . . . 159

6 Method's implementation 161

6.1 Disaster management system modeling . . . 162
6.1.1 User interface . . . 163

6.1.2 Extraction and integration of knowledge from data 166
6.1.3 Rule-based reasoning on the ontology SemDM 171
6.2 Simulation modeling 173
6.2.1 Multi-agent simulation modeling from a disaster management model 175
6.2.2 Modeling of the programmed simulation model 184
6.3 Simulation design 187
6.3.1 Generative programming 187
6.3.2 Simulation execution 192
6.4 Clustering-based analysis 194
6.4.1 Effectiveness score-based segmentation 195
6.4.2 Instance case segmentation 196
6.4.3 Knowledge enrichment 197
6.5 Summary 201

7 Use Case 203

7.1 Case study 204
7.1.1 Plan description 204
7.1.2 Scenarios 206
7.2 Modeling of disaster management case study 207
7.2.1 Knowledge integration 208
7.2.2 Reasoning on disaster management model according to an event . . . 219
7.3 Simulation modeling 220
7.3.1 Conceptual simulation modeling from disaster management case study 220
7.3.2 Program modeling from the conceptual simulation model 227
7.4 Simulation design 229
7.4.1 Generative programming 230
7.4.2 Simulation execution 239
7.5 Clustering-based analysis 241
7.5.1 Effectiveness score-based segmentation 242
7.5.2 Instance case segmentation 244
7.5.3 Knowledge enrichment 246
7.5.4 Plan assessment result 248

7.6 Summary 253

8 Evaluation **255**

8.1 Evaluation methods 256

8.1.1 Criteria metrics for high-level metrics computation 256

8.1.2 Computation of metrics 260

8.2 Evaluation application 261

8.2.1 Evaluation of environment complexity 262

8.2.2 Evaluation of agent's rationality 264

8.2.3 Evaluation of agent's autonomy 266

8.2.4 Evaluation of agent's reactivity 268

8.2.5 Evaluation of agent's adaptability to the Environment 269

8.3 Discussion 272

8.3.1 Rationality vs. Environment Complexity 272

8.3.2 Autonomy vs. Environment Complexity 273

8.3.3 Reactivity vs. Environment Complexity 274

8.3.4 Adaptability vs. Environment Complexity 274

8.3.5 Conclusion 275

9 Conclusion **277**

9.1 Contributions 278

9.1.1 Multi-agent simulation adaptation 279

9.1.2 Plan representation 280

9.1.3 Representation of the plan's effectiveness associated with its applicability context 280

9.2 Discussions 281

9.3 Future work 283

9.3.1 Short term perspective 284

9.3.2 Medium-term perspective 285

9.3.3 Long term perspective 285

Bibliography **308**

List of figures **313**

List of tables **317**

A Related work appendix **319**

A.1 Knowledge engineering 319

A.1.1 Knowledge representation 319

A.1.2 Knowledge base 328

A.2 Spatial data models 334

A.3 Practice from different countries 340

A.3.1 Disaster management in France 340

A.3.2 Disaster management in Germany 343

A.3.3 Comparison of disaster management strategy 346

A.3.4 Discussion 347

B Method's implementation appendix **351**

B.1 Processes intervening in the approach of extraction and integration of knowledge from data (section 6.1.2) 351

B.1.1 Geometry processing 351

B.1.2 Feature value analysis 353

B.1.3 Feature description analysis 355

B.2 Example of a SHACL (Shapes Constraint Language) rule to explain its structure 356

B.3 Prefixes 358

1 Résumé détaillé de la thèse en français / Detailed summary of the thesis in French

Les catastrophes naturelles ont toujours eu un impact sur les êtres vivants de la terre, apportant le chaos et la mort jusqu'à parfois provoquer l'extinction. Rapidement, les humains ont essayé d'atténuer l'impact des catastrophes naturelles et d'origine humaine. Des solutions d'atténuation ont été trouvées à travers la création de nouvelles technologies pour gérer les effets des catastrophes (par exemple, un projet de roues hydrauliques pour conduire l'eau des inondations du Nil dans un lac, le pharaon Amenemhat III entre 1817 et 1722 avant J.-C. en Egypte), par la création d'experts en gestion des catastrophes (par exemple, l'unité citoyenne créée par l'empereur Auguste, dans l'armée pour combattre le feu après l'incendie qui a dévasté Rome en 64 après J.C.) ou par le développement de stratégies pour réduire la vulnérabilité à une catastrophe et son risque (par exemple, la construction de villes incas dans les montagnes et l'adaptation du mode de vie pour limiter les attaques ennemies) [Coppola, 2011]. Aujourd'hui, nous continuons à développer des stratégies pour réduire les risques et atténuer les effets des catastrophes. Cependant, l'importance de ces stratégies augmente avec la récente augmentation d'environ 61,9 % du nombre moyen de catastrophes naturelles par an, entre la période 1983-1999 (256 catastrophes/an) et 2000-2016 (413 catastrophes/an) [Magdelaine, 2009]. Depuis la conférence internationale de mai 1994 à Yokohama (Japon), la stratégie mondiale de gestion des catastrophes est un cycle composé de quatre étapes: l'atténuation, la préparation, la réponse et le rétablissement [Coppola, 2011]. Le rôle de ce cycle vise à améliorer et à adapter la gestion des catastrophes de manière continue afin de réduire l'impact d'une catastrophe. L'étape d'atténuation vise à évaluer le risque de catastrophe afin de réduire la vulnérabilité et d'accroître la résilience des zones à risque. L'étape de préparation vise à être prêt à répondre à une catastrophe en (1) élaborant des plans de gestion des catastrophes, (2) en formant les intervenants et la population à faire face aux catastrophes, et (3) en surveillant les éléments annonciateurs d'une catastrophe en vue d'une alerte précoce et rapide. L'étape de réponse correspond à l'étape d'urgence et consiste à limiter les blessés, les décès et la quantité de dommages par les interventions de différentes organisations. L'étape de rétablissement vise à revenir à une situation normale. Parmi toutes ces étapes, la phase de réponse est la plus critique en termes de temps. L'efficacité de la réponse exige que les actions et la coordination entre les différentes parties prenantes soient mises en place rapidement et efficacement. Les intervenants doivent savoir quoi faire dans quelle circonstance et être prêts à le faire, pour obtenir cette facilité pendant la réponse. L'étape de préparation est donc essentielle et cruciale. L'élaboration de plans est à la fois au cœur de la préparation, sur laquelle repose la formation des intervenants, et la source d'une réponse efficace.

Les plans de gestion des catastrophes décrivent les actions à entreprendre en fonction de la situation, les responsables et les acteurs de ces actions, ainsi que les ressources nécessaires (telles que les équipements ou les véhicules) pour réaliser ces actions. L'élaboration de plans est une tâche complexe. Les plans doivent faire face à une diversité de cas, de conditions et de scénarios, ce qui signifie qu'ils doivent s'adapter à une diversité de situations (par exemple, type de catastrophe, situation géospatiale et financière). Par conséquent, leur élaboration exige de prendre en compte un ensemble complexe d'informations et la diversité des situations à affronter. Il existe différents niveaux de planification pour faciliter l'élaboration du plan (c'est-à-dire stratégique, tactique et opérationnel). Le niveau le plus élevé fournit des lignes directrices pour l'élaboration des plans, tandis que le niveau le plus bas fournit des plans précis et adaptés à la situation. L'adaptation à la situation nécessite la collecte de données géospatiales telles que des informations sur les bâtiments, les routes, les infrastructures critiques, les réseaux fluviaux et les zones à risque d'une localité. La collecte de ces données géospatiales pour l'élaboration des plans fait face à des problèmes d'interopérabilité entre les différents systèmes et formats utilisés par les différents acteurs. De plus, l'élaboration d'un plan nécessite une étape d'évaluation pour déterminer l'efficacité d'un plan, surtout pour les plans de niveau inférieur qui sont les plus difficiles à élaborer pour apporter une réponse efficace et spécifique à la situation. L'évaluation du plan nécessite de nombreuses ressources pour tester les plans dans une grande diversité de situations et nécessite la collecte de données hétérogènes.

Pour résoudre le problème de l'interopérabilité des données dans le contexte de la gestion des catastrophes et faire face aux défis de la préparation, l'institut i3mainz de l'Université des Sciences Appliquées de Mayence (Allemagne) a créé le projet SemGIS[1], financé par le ministère allemand de l'éducation et de la recherche. Ce projet a concentré ses recherches sur (1) la collecte et le choix des données à utiliser et (2) l'élaboration de plans. Le premier aspect aborde la question de l'évaluation de la qualité des données et de leur évolution pour soutenir le choix des données utilisables dans le contexte de la gestion des catastrophes. Cet aspect n'est pas présenté dans ce manuscrit. Le deuxième aspect concerne l'évaluation de l'efficacité des plans en fonction de l'ensemble des données intégrées, dont les plans des différentes parties prenantes. Ce deuxième aspect est le sujet de ce manuscrit.

[1] Site Web du projet SemGIS : http://i3mainz.hs-mainz.de/de/projekte/semanticgis

1.1 Contexte et motivation

La préparation à la gestion des catastrophes repose sur la préparation des plans. Elle consiste à organiser les responsabilités, la collaboration et la planification des tâches, des actions et des ressources nécessaires pour faire face aux risques réels de catastrophe. Les plans visent à savoir ce qu'il faut faire et comment le faire lorsqu'une catastrophe se produit. Ces plans permettent également aux parties prenantes de déterminer le type de formation nécessaire, mais aussi ce qu'il faut surveiller, quand et qui alerter pour une alerte précoce adéquate. La formation et la surveillance sont deux autres étapes principales de la préparation [Coppola, 2011], dont l'efficacité dépend de la préparation des plans. La préparation des plans est au cœur de la gestion des catastrophes, et leur efficacité est cruciale. C'est pourquoi il faut les évaluer et les améliorer jusqu'à ce qu'ils soient efficaces.

1.1.1 Évaluation de plans de gestion des catastrophes

L'évaluation de plans est une étape essentielle pour la préparation. Elle vise à savoir dans quelles situations un plan est efficace ou non et à définir dans quelles conditions un plan est applicable. Comme l'affirme l'auteur de [Larsson, 2008], "*un plan particulier de préparation doit être motivé par au moins une appréhension de son efficacité*". Ces connaissances sur l'applicabilité et l'efficacité des plans fournissent une base pour améliorer les plans et les compléter par de nouveaux plans afin de faire face à des situations pour lesquelles tous les plans sont inefficaces. Ce processus récursif entre l'élaboration et l'évaluation des plans vise à aider la communauté de gestion des catastrophes à être bien préparée.

L'évaluation d'un plan peut être réalisée à l'aide de plusieurs critères comme les critères d'organisation, de maturité et d'efficacité [Larsson, 2008]. Parmi ces critères, seuls les critères d'efficacité fournissent une valeur numérique. Les autres critères comme l'organisation et la maturité sont basés sur le contenu du plan et la préparation générale (par exemple, le système d'alerte, le respect des lois). Ces critères peuvent être utilisés pour vérifier l'élaboration du plan, mais peuvent également servir de ligne directrice non seulement pendant l'élaboration du plan mais aussi pendant la préparation générale. Au contraire, la mesure numérique de l'efficacité anticipe le niveau d'efficacité d'un plan exécuté dans un scénario spécifique. Ce critère est celui qui fournit la connaissance des situations dans lesquelles un plan est efficace ou non et permet donc de l'améliorer. L'efficacité du plan dépend de la réalisation de ses objectifs. L'auteur de [Larsson, 2008] précise que "*l'efficacité est évaluée en fonction de la mesure dans laquelle les objectifs sont atteints, sans se préoccuper des activités spécifiques entreprises pour atteindre les objectifs*". Par conséquent, l'évaluation de l'efficacité du plan nécessite (1) l'exécution du plan, et (2) la définition des objectifs pour adapter son évaluation. La définition de l'objectif du plan fait partie de l'élaboration du plan. En effet, l'élaboration de plans nécessite de définir ce que le plan adresse et ce que l'on attend de lui. Cependant, l'exécution d'un plan est une autre étape qui nécessite une application pratique du plan. L'application pratique du plan dans le cadre de son évaluation présente deux défis principaux. Le premier défi est le réalisme d'une application pratique, et le second défi concerne la nécessité d'une grande diversité de situations définies par des applications pratiques. Une grande diversité de situations est nécessaire pour permettre une évaluation exhaustive de l'efficacité.

1.1.2 Application pratique d'un plan

Il existe deux approches principales pour l'application pratique d'un plan. La première est un exercice réel et la seconde est une simulation informatique.

1.1.2.1 Exercice réel vs Simulation informatique

Les exercices réels nécessitent de reproduire des scénarios de catastrophe et de réunir tous les acteurs pour simuler et tester la mise en œuvre des plans en fonction de situations réelles. Cela nécessite la mise en place de beaucoup de moyens et d'argent pour reproduire des scénarios de catastrophe. Plus les exercices doivent être réalistes, plus les coûts sont élevés. Les exercices ont l'avantage de permettre à la fois l'évaluation de l'efficacité du plan et la formation des acteurs simultanément. Bien qu'il y ait quelques biais lors d'un exercice en raison de l'état d'esprit de l'intervenant qui sait que ce n'est pas réel, les exercices réels sont assez précis. Son principal inconvénient est un coût d'organisation élevé pour réunir toutes les parties prenantes et mettre en place les situations de catastrophe les plus réelles.

La simulation informatique est utilisée pour évaluer un système par le biais d'expériences. Dans le but d'évaluer un plan, les expériences peuvent simuler le système de gestion des catastrophes basé sur un plan dans une grande diversité de scénarios. Une telle simulation nécessite un modèle du système, exprimé par des paradigmes informatiques.

> *"Un modèle est une représentation d'un système qui peut être défini et étudié indirectement en aidant à fournir des réponses à son sujet". [Rodrigues Da Silva, 2015]*

Trois étapes de conception constituent le processus de modélisation de la simulation : premièrement, la conception d'un modèle conceptuel (c'est-à-dire la conceptualisation du modèle), deuxièmement, la conception du modèle communicatif (c'est-à-dire la représentation de la conceptualisation indépendante d'une plateforme), troisièmement, la conception du modèle programmé (c'est-à-dire la programmation du modèle), et enfin, la conception du modèle d'expérience (c'est-à-dire la mise en place des expériences). Selon [Nance, 1994], le modèle conceptuel est le "*modèle qui existe dans l'esprit du modélisateur*", alors que le modèle communicatif est une représentation du modèle qui peut être communiquée à d'autres. Dans la littérature [Benjamin et al., 2006], la distinction entre le modèle conceptuel et sa représentation (c'est-à-dire le modèle communicatif) n'est pas toujours

faite. Par conséquent, le terme de modèle conceptuel est plus souvent utilisé pour décrire sa représentation que son abstraction dans l'esprit du concepteur. C'est pourquoi le terme "modèle conceptuel" est utilisé dans le reste de la thèse pour décrire la représentation du modèle conceptuel (c'est-à-dire que l'on utilisera "modèle conceptuel" au lieu de "modèle communicatif").
Le modèle programmé, également appelé modèle de calcul, "*est une représentation du modèle qui admet l'exécution par un ordinateur pour produire des résultats de simulation*" [Nance, 1994]. Enfin, le modèle expérimental est formé par l'ajout de descriptions exécutables de l'environnement de test au modèle programmé qui permet l'exécution de la simulation. Il est basé sur différentes configurations de paramètres permettant la simulation du système modélisé dans différents scénarios. La simulation informatique a l'avantage d'être une approche peu coûteuse pour tester un plan dans une grande diversité de scénarios. Bien que d'un bon niveau de réalisme, les simulations peuvent être moins précises qu'un exercice réel.

Une évaluation correcte des plans doit être basée sur plusieurs exercices dans différentes situations afin de fournir un retour d'information complet pour tester les plans en fonction de la situation et les évaluer. Néanmoins, le coût élevé des exercices réels limite le nombre d'itérations, la complexité et le réalisme des scénarios. Bien que les simulations soient moins précises que les exercices réels, la précision des simulations peut atteindre un niveau suffisamment élevé pour évaluer l'efficacité d'un plan. C'est pourquoi les simulations informatiques sont les plus adaptées aux applications pratiques des plans afin d'évaluer leur efficacité. Ainsi, les simulations informatiques permettent de tester les plans dans une grande variété de scénarios complexes pour un faible coût.

1.1.2.2 Technique de simulation adaptée aux besoins de la solution

Les simulations informatiques regroupent plusieurs types de techniques. Il est essentiel de choisir la technique de simulation en fonction de l'objectif de la simulation et du système à modéliser. Les auteurs de [Mishra et al., 2019] passent en revue les techniques de simulation utilisées dans la littérature pour la gestion des catastrophes et examinent les problèmes adressés. Les techniques identifiées comme les plus couramment utilisées sont la simulation d'événements discrets, la dynamique des systèmes, la simulation à base d'agents et la simulation de Monte Carlo. L'étude des problèmes adressés par ces techniques montre que la simulation basée sur des agents est la plus utilisée lors de la préparation pour évaluer les stratégies et les plans de réponse aux catastrophes [Mishra et al., 2019].

Une simulation à base d'agents est une simulation basée sur un système multi-agents. Selon l'auteur de [Ferber, 1997], un système multi-agents est :

1. "*Un Environnement E, qui est en général un espace avec une métrique.*
2. *Un ensemble d'objets O. Ces objets sont localisés, cela signifie que pour tous les objets, on*

peut associer une position dans E à un temps spécifique. Ces objets sont passifs, ils peuvent être perçus, créés, détruits, et modifiés par des Agents.

3. *Un ensemble d'agents A, qui sont des objets particuliers. Ce sont les entités actives du système.*

4. *Un ensemble de relations R, qui relient les objets (et donc les agents) entre eux.*

5. *Un ensemble d'opérations Op permettant aux agents de percevoir, produire, consommer, transformer, et manipuler les objets dans O.*

6. *Opérateurs, qui représentent l'application des opérations et des réactions du monde selon la modification (on les appelle les lois de l'univers)*".

Ils sont également appelés simulation multi-agents. Cette technique est utilisée pour observer l'émergence d'un système complexe à partir de la modélisation de ses composants, correspondant principalement aux comportements des agents et à leurs interactions. L'évaluation de l'efficacité d'un plan nécessite l'observation du système de gestion des catastrophes. Le système de gestion des catastrophes est un système complexe, dont les composants sont les intervenants et les autres parties prenantes. Les plans décrivent le rôle, les actions et l'interaction entre les acteurs de la réponse. Ainsi, la technique de simulation à base d'agents est la plus adaptée pour simuler un système de gestion des catastrophes à partir de la modélisation de ses composants basée sur la description d'un plan.

1.1.3 Défis et objectifs

La simulation multi-agents est la technique la plus appropriée pour réaliser l'application pratique des plans dans diverses situations. Une telle technique nécessite des étapes de modélisation et de conception pour produire des résultats d'expériences de simulation. Ces résultats servent de base pour étudier la réalisation des objectifs des plans et évaluer leur efficacité. Cependant, la simulation de la diversité des plans et des situations implique un changement des objectifs, des parties prenantes, des actions et des interactions. Selon les objectifs visés, ces changements impliquent une adaptation de l'application pratique par la simulation et une adaptation de l'évaluation de l'efficacité.

Adaptation de la simulation Un certain niveau d'adaptation de la simulation peut être géré par l'adaptation de la variation des paramètres du modèle. Les caractéristiques de certaines situations peuvent être configurées en tant que paramètres pour permettre l'expérimentation de la simulation dans une diversité de situations. La majorité des approches de simulation de la littérature soutenant la préparation à la gestion des catastrophes est spécifique à un plan ou à une catégorie de plan. Selon la catégorie ou le plan concerné, elles ont un objectif spécifique, la conception d'un modèle de simulation, et un ensemble de configurations de paramètres représentant les expériences de simulation pour atteindre leur but. Cependant, les changements d'objectifs, d'acteurs, d'actions et d'interactions ne peuvent pas être gérés uniquement par des variations de paramètres. Ils impliquent un changement dans la modélisation et la conception de la simulation en raison de la variation des objectifs des expériences et des composants du système. Par conséquent, le premier défi de l'évaluation de l'efficacité du plan est l'adaptation de la modélisation et de la conception des expériences de simulation en fonction de la représentation du plan, en décrivant son objectif, les parties prenantes, leurs actions et leurs interactions. Ce défi regroupe deux sous-défis. Au niveau de la conceptualisation, le défi est ***l'adaptation du modèle conceptuel et de la représentation des expériences à la diversité des plans***. Il consiste à fournir un modèle multi-agent qui peut représenter la diversité des plans dans le paradigme agent et représenter la diversité des scénarios par des expériences pour permettre l'évaluation des plans. Cette thèse vise à relever ce défi en *augmentant la flexibilité pour permettre l'évaluation des divers plans de gestion des catastrophes à travers leur simulation*. Au niveau de la programmation, le défi consiste à disposer de ***composants de simulation multi-agents adaptables et réutilisables pour la gestion des catastrophes***, permettant l'adaptation du modèle programmé. L'objectif pour relever ce défi est *l'augmentation de l'extensibilité en permettant de traiter la diversité des plans*.

Représentation des plans Ces deux défis liés à l'adaptation de la simulation dépendent de la définition d'un plan. Son niveau de détail définit le niveau de détail de la simulation. La conceptualisation de la simulation est généralement faite de manière ad hoc à partir des plans et des données connexes fournies par les experts en gestion des catastrophes. La conceptualisation est réalisée dans le paradigme agent qui nécessite un concepteur informaticien. Cette exigence implique qu'un concepteur informaticien doit effectuer une nouvelle modélisation à chaque nouveau niveau de description du plan ou à chaque nouvelle modification du plan. Les problèmes de représentation du plan directement dans le paradigme agent sont (1) l'inaccessibilité de la modélisation de l'expert en gestion des catastrophes pour modifier certains aspects de la représentation du plan et (2) les biais de conceptualisation générés par la nouvelle modélisation qui ne garantit pas une conceptualisation de simulation uniforme pour tous les plans. Une conceptualisation de simulation non uniforme pour tous les plans peut produire des biais dans leur évaluation et, par conséquent, un non-sens dans la comparaison des valeurs d'efficacité. Par conséquent, le premier défi énoncé est d'adapter le modèle conceptuel et la représentation des expériences à la diversité des plans et de garantir une conceptualisation uniforme. Une telle uniformité de conceptualisation pourrait être assurée par un processus automatique de conceptualisation qui suivrait la même méthode de conceptualisation. Les problèmes énoncés génèrent un nouveau défi lié à la nécessité d'une représentation des ***plans qui doit être comprise à la fois par l'expert du domaine pour les modéliser et par la machine pour être traiter***. Cette thèse vise à *augmenter l'expressivité et l'interopérabilité pour rassembler les connaissances en matière de gestion des catastrophes et permettre aux experts en gestion des catastrophes d'évaluer* *les plans en fonction de leur définition* pour relever ce défi.

Adaptation de l'évaluation Des expériences de simulation fournissent l'application pratique, dont les résultats sont nécessaires pour évaluer l'efficacité du plan. Le quatrième défi de l'évaluation de l'efficacité du plan est ***l'adaptation de l'analyse des résultats en fonction des objectifs du plan pour fournir une valeur d'efficacité***. Une évaluation exhaustive du plan dans une grande diversité de situations peut fournir une efficacité globale caractérisant la catégorie de situation. Dans le cas d'une efficacité différente et spécifique pour les différentes situations, fournir une valeur d'efficacité globale n'aurait aucun sens. Dans ce cas, il convient de fournir des valeurs d'efficacité associées à la description de la situation caractérisant la valeur d'efficacité. Par conséquent, le dernier objectif de cette thèse est de *représenter les valeurs d'efficacité associées à leur contexte d'applicabilité.*

1.2 Énoncé du problème

À partir des défis et des objectifs énoncés dans la section précédente, il est pertinent de rechercher une méthode qui adapte automatiquement la simulation multi-agents en

- conceptualisant le modèle de simulation multi-agents et
- concevant le modèle programmé basé sur des composants de simulation multi-agents adaptables et réutilisables,
- selon une représentation de plan compréhensible par la communauté de gestion des catastrophes et les machines
- permettant d'évaluer et de représenter l'efficacité d'un plan en fonction de son contexte d'application.

Par conséquent, cette thèse aborde la problématique suivante :

Comment adapter automatiquement la simulation multi-agents à partir de la représentation sémantique des plans de gestion des catastrophes pour évaluer leur efficacité ?

1.2.1 Enjeux

Les enjeux proviennent des limites des travaux sur:

1. l'adaptation de la simulation multi-agents,
2. la représentation du plan, et
3. la représentation de l'efficacité du plan associée à une définition de la situation.

1.2.1.1 Adaptation de la simulation multi-agents

Dans la littérature, les approches liées à l'augmentation de l'adaptabilité des simulations multi-agents proposent des méta-modèles et des ontologies permettant de concevoir une diversité de modèles de simulation multi-agents. Un méta-modèle est un modèle de modélisation :

> *"qui définit la structure d'un ensemble de modèles conformes à une syntaxe et une sémantique données." [Rodrigues Da Silva, 2015]*

Une ontologie est définie comme

> *"une spécification formelle et explicite d'une conceptualisation partagée" [Studer et al., 1998].*

Ces méta-modèles et ontologies sont souvent utilisés comme base pour la programmation générative.

> *"La programmation générative est une tentative de fabriquer des composants logiciels de manière automatisée en développant des programmes qui synthétisent d'autres programmes." [Cointe, 2005]*

La programmation générative est basée sur un ensemble de composants de simulation multi-agents combinés et paramétrés pour concevoir le code de simulation. Ces approches sont utilisées pour automatiser le développement de la simulation et ont l'avantage de faciliter l'interopérabilité avec d'autres systèmes.

Conceptualisation du modèle de simulation multi-agents Parmi les méta-modèles et les ontologies pour la simulation multi-agents de la littérature, l'ontologie présentée par les auteurs de [Christley et al., 2004] est la plus pertinente pour adapter la simulation à l'évaluation des plans. Cette ontologie a l'avantage de fournir des concepts de haut niveau pour la simulation multi-agents, permettant la représentation d'une diversité de cas d'application. Cependant, elle ne contient pas de concepts spécifiques pour la simulation de gestion des catastrophes.

Conception du modèle programmé basé sur des composants de simulation multi-agents adaptables et réutilisables Un modèle programmé dépend d'une plateforme de simulation avec laquelle il peut être exécuté. Parmi les différentes plateformes de simulation multi-agents étudiées, la plateforme GAMA [Taillandier et al., 2019] semble être la plus adaptée à la simulation des plans. Elle présente l'avantage de permettre la représentation du monde réel et des systèmes d'information géographique (SIG), les simulations à grande échelle, la simulation scientifique, la planification et l'ordonnancement de simulations à base d'agents polyvalents, les ressources naturelles et l'environnement. Toutefois, la plateforme GAMA ne dispose pas d'un ensemble de comportements d'agents spécifiques au domaine de la gestion des catastrophes.

1.2.1.2 Représentation de plans

Les ontologies et les méta-modèles fournissent un modèle sémantique qui permet la représentation d'une diversité de plans de gestion des catastrophes. Par conséquent, les ontologies et les méta-modèles liés à la gestion des catastrophes sont examinés pour l'évaluation des plans. En plus d'une formalisation, la représentation des plans nécessite la représentation des plans spécifiques. Une grande partie des connaissances relatives à ces plans est stockée dans des données géospatiales hétérogènes. Par conséquent, l'augmentation de l'interopérabilité pour la collecte de connaissances sur la gestion des catastrophes et la représentation des plans nécessite l'extraction de connaissances à partir de données géospatiales hétérogènes.

Formalisation des plans Parmi les ontologies et les méta-modèles examinés, deux approches semblent les plus pertinentes pour la représentation des plans : le méta-modèle présenté par [Othman et al., 2014] et l'ontologie Emergel [Casado et al., 2015]. Le méta-modèle de [Othman et al., 2014] a l'avantage de fournir des concepts de haut niveau permettant la définition d'une grande variété de plans. Cependant, ce méta-modèle n'est pas exprimé en OWL et ne contient pas de concepts spécifiques pour définir le contenu des plans. L'ontologie Emergel [Casado et al., 2015] fournit un vocabulaire complet pour le contenu des plans mais ne permet pas la représentation des plans.

Extraction de connaissances à partir de données géospatiales hétérogènes Parmi les approches étudiées d'extraction de connaissances à partir de données géospatiales hétérogènes, le projet Datalift [Scharffe et al., 2012] propose l'approche automatique la plus intéressante. Cependant, Datalift crée des individus avec des annotations, ce qui n'est pas la représentation RDF la plus adaptée pour intégrer les connaissances. Une représentation RDF avec des individus liés par des propriétés serait plus adaptée.

1.2.1.3 Représentation de l'efficacité du plan associée à son contexte d'applicabilité

La représentation de l'efficacité d'un plan associée à son contexte d'applicabilité nécessite de regrouper les situations pour lesquelles un plan a une efficacité similaire. Dans le cas d'une application de plan à grande échelle ou de tests approfondis de ces plans, le regroupement doit être non supervisé afin d'éviter une analyse humaine sujette à erreur lorsque le nombre de situations et de caractéristiques est important. Parmi les regroupements non supervisés, l'approche CURE [Guha et al., 2001] semble présenter le meilleur compromis entre la complexité de calcul et la qualité du regroupement. Cependant, l'application du regroupement sur les résultats de simulation conduit à une sur-segmentation en considérant des critères qui n'ont pas d'incidence sur l'efficacité du plan.

1.2.2 Besoins

La résolution de la problématique de cette thèse nécessite de combler les limites des travaux connexes présentés précédemment. Le dépassement de ces limites implique un ensemble de besoins liés à chaque limite identifiée.

1.2.2.1 Adaptation de la simulation multi-agents

Les travaux liés à l'adaptation de la simulation multi-agents ont permis d'identifier deux approches pertinentes : l'ontologie de la modélisation de la simulation multi-agents de [Christley et al., 2004] et la plateforme GAMA [Taillandier et al., 2019]. La limite de l'approche de la [Christley et al., 2004] exige une ***spécification des concepts pour la simulation de gestion des catastrophes*** pour compléter les concepts de haut niveau de la simulation multi-agents. Le manque de compétences des agents pour la gestion des catastrophes de la plateforme GAMA [Taillandier et al., 2019] nécessite une ***extension des comportements des agents pour les actions de gestion des catastrophes*** par l'ajout d'un plugin externe appelé "skills". Ces deux premières exigences correspondent à des composants essentiels pour l'adaptation de la simulation. Cependant, l'adaptation de la simulation nécessite également ***un processus automatique pour générer le modèle de simulation conceptuel selon la définition du plan*** et ***une approche automatisée pour développer le modèle de simulation programmé***.

1.2.2.2 Représentation de plans

Les travaux étudiés relatifs à la représentation de plans ont permis d'identifier deux approches pertinentes pour sa formalisation : le méta-modèle présenté par [Othman et al., 2014] et l'ontologie Emergel [Casado et al., 2015]. En plus de ces approches, le projet Datalift [Scharffe et al., 2012] a été identifié comme intéressant pour extraire des connaissances des données. Les approches de [Othman et al., 2014] et [Casado et al., 2015] ont des limites complémentaires : la première n'a pas de concepts de bas niveau pour décrire le contenu d'un plan, tandis que la seconde n'a pas de concept de haut niveau pour la description de plans. Ces approches ne permettent pas de formaliser la combinaison des concepts de haut niveau et de bas niveau pour représenter les plans de gestion des catastrophes. Par conséquent, il est nécessaire de fournir un modèle de connaissance de la gestion des catastrophes pour l'évaluation de plans afin de fixer la limite de la combinaison des concepts de haut niveau et de bas niveau pour représenter un plan de gestion des catastrophes. La limite du projet Datalift [Scharffe et al., 2012] rend difficile l'intégration des connaissances extraites dans l'ontologie pour l'évaluation des plans de gestion des catastrophes. Par conséquent, ***une nouvelle approche automatique pour l'extraction de connaissances à partir de données*** est nécessaire pour permettre leur intégration.

1.2.2.3 Représentation de l'efficacité du plan associée à son contexte d'applicabilité

L'approche de regroupement hiérarchique non supervisé [Guha et al., 2001], identifiée comme la plus appropriée, pose un problème de sur-segmentation. Ce problème provient de la prise en compte de critères qui n'ont pas d'incidence sur l'efficacité du plan. Par conséquent, une ***identification des critères pertinents ayant un impact sur l'efficacité du plan*** est nécessaire pour surmonter ce problème.

1.3 Approche proposée

Ce manuscrit propose une approche d'ingénierie de simulation multi-agents basée sur la connaissance pour évaluer l'efficacité des plans de gestion des catastrophes afin de résoudre le problème mentionné dans la section 2.2.1 en répondant à l'exigence présentée dans la section 2.2.2. La première sous-section présente les approches proposées pour répondre aux exigences, et la seconde décrit l'approche pour évaluer l'efficacité d'un plan d'intervention.

1.3.1 Approches proposées pour répondre aux besoins

Les approches proposées pour répondre aux besoins sont liées (1) à l'adaptation de la simulation multi-agents, (2) à la représentation de plans et (3) à la représentation de l'efficacité d'un plan associée à son contexte d'applicabilité.

1.3.1.1 Adaptation de la simulation multi-agents

Adaptabilité pour simuler et évaluer la diversité de représentation des plans Cette thèse propose de répondre au besoin d'adaptabilité de la simulation pour évaluer la diversité de représentation des plans (1) en précisant les concepts de simulation de gestion des catastrophes, (2) en générant un modèle conceptuel de simulation en fonction de la définition d'un plan et (3) en générant le modèle programmé de simulation correspondant. Trois approches sont présentées pour répondre à chacune de ces exigences. La première approche est l'ontologie SemMAS (Semantic Multi-Agent Simulation), qui combine des concepts de haut niveau représentant le domaine de la simulation multi-agents. Elle s'inspire des travaux de [Christley et al., 2004] complétés par des concepts de bas niveau représentant la simulation de gestion des catastrophes et liés à la plateforme GAMA. La deuxième approche est une modélisation de simulation basée sur la connaissance qui utilise la représentation d'un plan pour générer la conceptualisation de la simulation dans l'ontologie SemMAS. La troisième approche est une programmation de simulation basée sur la connaissance pour la plateforme GAMA qui génère le code de simulation à partir du contenu de l'ontologie SemMAS afin d'exécuter des simulations avec la plateforme GAMA.

Adaptabilité et réutilisation des composants de simulation multi-agents pour la gestion des catastrophes Une extension des comportements d'agents pour les actions de gestion des catastrophes est proposée pour résoudre les problèmes d'adaptabilité et de réutilisation des composants de simulation multi-agents. L'approche pour répondre à cette exigence correspond au développement de nouvelles compétences d'agents pour la plateforme GAMA. Ces nouvelles compétences et leurs actions associées sont représentées dans l'ontologie SemMAS.

1.3.1.2 Représentation de plans

Cette thèse propose de résoudre le problème de la représentation des plans et des scénarios exprimée directement dans un paradigme de simulation multi-agents en les représentant à travers une ontologie pour l'évaluation des plans de gestion des catastrophes et en intégrant les connaissances extraites de données hétérogènes. L'ontologie SemDM (Semantic Disaster Management) combinant les concepts de gestion des catastrophes de haut et de bas niveau est proposée pour répondre à la première exigence. Les concepts de haut niveau de cette ontologie sont inspirés du méta-modèle présenté par [Othman et al., 2014], tandis que les concepts de bas niveau sont inspirés de l'ontologie Emergel [Casado et al., 2015]. Une approche d'intégration automatique des connaissances extraites de données hétérogènes est proposée pour intégrer les connaissances stockées à travers des données dans l'ontologie SemDM. Cette approche est basée sur le traitement du langage naturel et la dimension géospatiale.

1.3.1.3 Représentation de l'efficacité d'un plan associée à son contexte d'applicabilité

Enfin, cette thèse fixe la limite liée à l'évaluation des changements dans la description des plans en identifiant les critères pertinents ayant un impact sur l'efficacité du plan. Une analyse basée sur une combinaison de regroupement appliquée aux différents critères de simulation est proposée pour satisfaire ce besoin.

1.3.2 Approche pour évaluer l'efficacité d'un plan d'intervention

L'approche défendue dans cette thèse combine les approches présentées précédemment pour évaluer les plans d'intervention de gestion des catastrophes. Elle se compose de quatre étapes : (1) la modélisation des connaissances en matière de gestion des catastrophes, (2) la modélisation des simulations, (3) la conception des simulations, et (4) l'analyse des résultats des simulations sur la base de regroupements. Tout d'abord, les connaissances explicites et les données des experts sont utilisées pour créer un modèle de connaissances pour la gestion des catastrophes. Deuxièmement, les modèles de simulation sont conçus sur la base du modèle de connaissances. Troisièmement, la programmation générative est utilisée pour la conception des simulations. Enfin, les résultats des simulations sont utilisés pour calculer l'efficacité du plan pour chaque simulation. Le regroupement par apprentissage non supervisé permet d'identifier le contexte d'application lié à l'efficacité calculée. L'efficacité et le contexte d'application associé enrichissent le modèle de connaissance initial.

1.4 Contributions

Les approches proposées dans la section précédente apportent principalement des contributions dans le domaine de l'ingénierie de simulation multi-agents. Elles apportent également quelques contributions dans le domaine du Web sémantique et dans celui de la gestion des catastrophes.

1.4.1 Contribution pour la communauté de gestion des catastrophes

L'ensemble des travaux de cette thèse est appliqué au domaine de la gestion des catastrophes. C'est pourquoi ce travail apporte deux contributions dans ce domaine : la première est de **faciliter la collecte des connaissances sur la gestion des catastrophes pour le travail en collaboration**, et la seconde est **l'enrichissement des connaissances à partir de l'évaluation des plans de gestion des catastrophes**.

Tout d'abord, il contribue à fournir des outils, en facilitant le partage des informations et des connaissances avec d'autres systèmes, grâce aux technologies sémantiques. Les auteurs de [Bharosa et al., 2010] soulignent les avantages des systèmes de partage d'informations inter-organisationel pour résoudre les problèmes de coordination de la gestion des catastrophes. La conception de tels systèmes nécessite de résoudre des problèmes techniques d'interopérabilité. Le processus d'intégration automatique [Prudhomme et al., 2017a, Prudhomme et al., 2020a] et les capacités fournies par l'utilisation de la sémantique dans les SIG [Homburg et al., 2017] apportent une valeur ajoutée pour l'interopérabilité des systèmes. Cette approche proposant l'intégration de connaissances extraites de données hétérogènes permet d'utiliser des données provenant d'autres systèmes comme entrées pour l'évaluation d'un plan. Ainsi, elle facilite le travail de collaboration de la communauté de gestion des catastrophes et peut encourager les parties prenantes à partager des informations pour résoudre les difficultés d'intégration et de synergie entre les réseaux institutionnels.

Deuxièmement, il fournit un système qui soutient la préparation à la gestion des catastrophes [Prudhomme et al., 2017c, Prudhomme et al., 2018]. La contribution de ce système est de soutenir l'évaluation de l'efficacité des plans en fournissant sa représentation avec son contexte d'applicabilité. L'utilisation de la simulation pour soutenir l'évaluation des plans permet une préparation active et peu coûteuse de la communauté de gestion des catastrophes. Son avantage est de permettre une meilleure préparation de la communauté de gestion des catastrophes, qui est souvent plus une préparation symbolique qu'un reflet des réalités opérationnelles [McConnell and Drennan, 2006].

1.4.2 Contribution pour le Web sémantique

Le système basé sur la simulation dirigée par les connaissances utilise des technologies sémantiques pour concevoir et manipuler des modèles ontologiques de gestion des catastrophes et de simulation multi-agents. Toutefois, les approches existantes ont certaines limites pour répondre aux exigences du système. Ces limites précédemment énoncées ont conduit à des contributions dans le domaine de la sémantique. Les contributions de ce domaine améliorent la collecte d'informations et de connaissances sur la gestion des catastrophes et étendent les capacités de raisonnement pour permettre la conceptualisation de la simulation.

Le besoin de rassembler toutes les connaissances et informations pertinentes relatives aux plans de gestion des catastrophes dans une base de connaissances entraîne la nécessité de récupérer des données hétérogènes à partir d'une grande variété de sources de données. La première contribution dans ce domaine consiste à **faciliter l'accès à une grande variété de sources de données par l'utilisation d'un catalogue sémantique** [Prudhomme et al., 2016]. La deuxième contribution est **un processus d'intégration automatique des connaissances extraites de données hétérogènes qui facilite l'interopérabilité entre les systèmes** [Prudhomme et al., 2017a, Prudhomme et al., 2020a]. Cette contribution faite en collaboration avec d'autres facilite l'enrichissement entre les données et le Web sémantique.

Le processus de conceptualisation de la simulation est réalisé dans l'ontologie SemMAS à partir de l'ontologie SemDM et par raisonnement. Les **ontologies SemDM et SemMAS** sont deux contributions au Web sémantique. La première permet la représentation et le partage d'un plan de gestion des catastrophes, tandis que la seconde permet la représentation de modèles de simulation de gestion des catastrophes et de ses configurations [Prudhomme et al., 2019b].

L'ensemble des règles [Prudhomme et al., 2019b], sur lesquelles repose le raisonnement de la conceptualisation de la simulation, utilise de nouvelles implémentations de "built-ins"[2] de règles. Ces **built-ins** de règles sont utilisés pour augmenter les capacités de raisonnement pour la conceptualisation de la simulation [Prudhomme et al., 2019a].

[2] Un "built-in" est une fonction intégrée aux règles et utilisée par le raisonnement à base de règles

1.4.3 Contribution pour l'ingénierie de simulation multi-agents

La principale contribution de cette thèse se situe dans l'ingénierie de la simulation multi-agents. En effet, cette thèse apporte trois contributions, qui sont (i) **une approche dirigée par la connaissance pour la conceptualisation et la conception de la simulation**, (ii) **une programmation générative de simulation pour la plateforme GAMA, dirigée par les connaissances**, et (iii) **une extension des compétences d'agents en matière de gestion des catastrophes pour la plateforme GAMA**.

La modélisation de simulations basée sur les connaissances, proposée dans cette thèse permet d'automatiser la conceptualisation et la conception de la simulation [Prudhomme et al., 2019b]. Cette approche produit un modèle conceptuel de simulation indépendant de la plateforme et une représentation du modèle programmé pour la plateforme GAMA. Cette contribution permet l'adaptation de la modélisation à partir de diverses représentations de plans par une méthode uniforme évitant les différents biais de conceptualisation de la simulation pour l'évaluation du plan. De plus, l'utilisation d'une ontologie apporte des avantages pour faciliter le partage et la réutilisation des modèles de simulation dans la communauté de simulation multi-agents.

La plateforme choisie pour réaliser les expériences de simulation est la plateforme GAMA [Taillandier et al., 2019]. Un processus de programmation générative guidée par les connaissances a été développé pour implémenter le code correspondant à sa représentation faite dans l'ontologie SemMAS et spécifique à cette plateforme. Le code généré permet l'exécution d'expériences de simulation [Prudhomme et al., 2019a, Prudhomme et al., 2019a].

Cette programmation générative est basée sur les compétences et les réflexes d'un agent. Une extension des compétences d'un agent en matière de gestion des catastrophes pour la plateforme GAMA a été développée pour répondre aux exigences de la programmation générative. Ces contributions apportent des avantages en termes de productivité du développement de la simulation et évitent les différents biais de programmation pour l'évaluation des plans.

1.5 Vue d'ensemble de la thèse

La thèse est décrite à travers huit chapitres en anglais, dont le contenu est résumé dans cette section.

Le **chapitre 2** présente en premier lieu, le contexte général de la gestion des catastrophes dont les points critiques d'activité métier . Il explique ensuite le contexte de l'évaluation des plans de gestion des catastrophes et les besoins d'applications pratiques qui motivent la simulation multi-agents. L'évaluation d'un plan par la simulation multi-agents génère des défis et des objectifs sur lesquels repose l'énoncé du problème. L'énoncé du problème présente les enjeux associés aux objectifs et aux besoins pour résoudre la problématique. Ensuite, les approches proposées et les contributions de cette thèse sont présentées. Enfin, il donne un aperçu du contenu de la thèse.

Le **chapitre 3** vise à présenter les travaux relatifs à l'évaluation des plans de gestion des catastrophes. Il présente le domaine d'application de cette thèse, qui est la gestion des catastrophes, pour détailler le problème métier de l'évaluation des plans et mettre en évidence les limites des approches existantes. Elle passe ensuite en revue les techniques d'ingénierie, les composants et les plateformes de simulation multi-agents, qui sont les approches les plus appropriées pour expérimenter et évaluer les plans de gestion des catastrophes. Ces études visent à rechercher des approches appropriées et à identifier leurs limites pour atteindre les objectifs. Ce chapitre présente ensuite, le domaine de l'ingénierie de la connaissance pour permettre la représentation des connaissances en matière de gestion des catastrophes, dont la description des plans. Il consiste à passer en revue les ontologies existantes de ce domaine et les approches permettant d'intégrer les connaissances extraites des données. Enfin, le chapitre conclut sur les limites à surmonter pour une approche de bout en bout permettant à la communauté de gestion des catastrophes d'évaluer leurs plans.

Le **chapitre 4** vise à présenter l'analyse conceptuelle réalisée pour résoudre la problématique de cette thèse. Il expose d'abord les besoins en matière d'évaluation des plans découlant des limites des travaux actuels. Ce chapitre explique ensuite les approches proposées pour répondre aux besoins identifiés. Enfin, il explique comment et pourquoi ces différentes approches sont organisées ensemble pour fournir une méthode d'évaluation de l'efficacité d'un plan.

Le **chapitre 5** vise à présenter l'architecture utilisée pour la mise en œuvre de la méthode présentée dans le chapitre précédent. Ce chapitre présente ses quatre composants : une plateforme de simulation, une base de connaissances, un client et un serveur de traitement. La description de la plateforme de simulation explique les spécificités de la plateforme GAMA et les nouvelles compétences d'agents développées pour la gestion des catastrophes. Il décrit le contenu de la base de connaissances par la modélisation des connaissances des ontologies SemDM et SemMAS. Il présente ensuite le client permettant l'intégration des connaissances et l'évaluation d'un plan à travers des requêtes au serveur de traitement. Enfin, il explique le rôle du serveur de traitement, qui interagit et utilise les autres composants.

Le **chapitre 6** vise à présenter l'implémentation de la méthode réalisée par le serveur de traitement. La méthode comprend quatre étapes principales : la modélisation du système de gestion des catastrophes étudié, la modélisation de la simulation, la conception de la simulation et l'analyse des résultats des expériences de simulations, basée sur le regroupement. Le chapitre présente tout d'abord la modélisation du système étudié, réalisée par l'intégration des connaissances extraites des données et le raisonnement à base de règles sur l'ontologie SemDM. Ensuite, la modélisation de la simulation est expliquée à travers les processus de génération du modèle de simulation conceptuel à partir de l'ontologie SemDM et la représentation du modèle de simulation programmé pour la plateforme GAMA. Le chapitre présente ensuite l'implémentation du processus de programmation générative et l'exécution des simulations qui permettent de concevoir les simulations. Enfin, il explique le processus d'évaluation d'un plan basé sur l'analyse par regroupement et l'enrichissement de l'ontologie SemDM avec la représentation de l'efficacité du plan.

Le **chapitre 7** vise à illustrer les résultats de l'implémentation de la méthode sur un cas d'utilisation, étape par étape. L'étude de cas de ce chapitre est le plan français NOVI, dont l'objectif est de gérer de nombreuses victimes d'une catastrophe (plus de cent victimes). Ce plan est évalué à travers différents scénarios basés sur trois configurations d'une catastrophe affectant la ville de Montbard (France). Ce chapitre explique l'étude de cas tout d'abord à travers la description du plan NOVI et des trois scénarios. Ensuite, il présente la modélisation de l'étude de cas de gestion des catastrophes, par l'intégration des connaissances et le raisonnement à base de règles sur l'ontologie SemDM. À partir de l'ontologie SemDM, les résultats de la modélisation conceptuelle et de la simulation programmée sont présentés. Ensuite, il montre le programme, les expériences et les résultats obtenus par l'étape de conception de la simulation. Enfin, il décrit les résultats fournis par le processus d'analyse basé sur les regroupements.

Le **chapitre 8** vise à évaluer l'approche proposée dans cette thèse. En l'absence de travaux similaires pour l'évaluation des plans de gestion des catastrophes, ce chapitre concentre son évaluation sur le modèle de simulation multi-agents de l'approche proposée. Le modèle de simulation est le pivot de l'approche : il est le résultat de la conceptualisation et de la conception de la simulation à partir de l'intégration des connaissances en matière de gestion des catastrophes, et ses résultats sont la source de l'analyse basée sur le regroupement pour l'évaluation du plan. Par conséquent, ce chapitre présente, tout d'abord, la méthode d'évaluation à travers les critères de mesure utilisés et leur calcul. Ensuite, il explique et décrit l'application de la méthode d'évaluation à l'étude de cas. Enfin, il examine les forces et les faiblesses du modèle de simulation généré.

Le **chapitre 9** résume les contributions apportées par ce manuscrit. Il aborde ensuite les avantages et les limites de l'approche proposée. Enfin, il présente les perspectives à court, moyen et long terme.

2 Introduction

Contents

2.1 Context and motivation 29

2.1.1 Disaster management plan assessment 29

2.1.2 Practical application of a plan 30

2.1.3 Challenges and objectives 32

2.2 Problem statement 34

2.2.1 Issues 34

2.2.2 Requirements 36

2.3 Proposed approaches 37

2.3.1 Proposed approaches to meet requirements 38

2.3.2 Approach for assessing the effectiveness of a response plan 39

2.4 Contributions 39

2.4.1 Contribution to the disaster management community . . . 39

2.4.2 Contribution to the Semantic Web 40

2.4.3 Contribution to multi-agent simulation engineering 41

2.4.4 Publications 41

2.5 Thesis overview 43

The natural disasters have always impacted the living beings on the earth, bringing chaos and death until sometimes causing extinction. Quickly, humans have tried to mitigate the impact of natural and human-made disasters. The mitigation solutions have been found through the creation of new technologies to manage the disaster effects (e.g., a project of water wheels to conduct the water from the floods of Nil into a lake, Pharaoh Amenemhat III between 1817 and 1722 before J.-C. in Egypt), through the creation of experts in disaster management (e.g., a citizen unit created by the emperor August, into the army to fight fire after the fire that devastated Rome in 64 after J.C.) or through the development of strategies to reduce the vulnerability to a disaster and its risk (e.g., the building of Inca cities in mountains and lifestyle adaptation to limit enemy attacks) [Coppola, 2011]. Nowadays, we continue to develop strategies to reduce risks and mitigate disaster effects.

However, the importance of these strategies grows up with the recent increase around 61,9% of the average number of natural disasters per year, between the period of 1983-1999 (256 disasters/year) and 2000-2016 (413 disasters/year) [Magdelaine, 2009]. Since the international conference in May 1994 in Yokohama (Japan), the global strategy of disaster management (DM) is a cycle composed of Mitigation, Preparedness, Response, and Recovery [Coppola, 2011]. The role of this cycle aims at improving and adapting the disaster management continuously to reduce the impact of a disaster. The mitigation step aims at assessing the risk of disaster to decrease the vulnerability and increase the resilience of risk areas. The preparedness step aims at being prepared to respond to a disaster by (1) elaborating disaster management plans, (2) training responders and population to face disaster, and (3) monitoring the elements heralding of a disaster for early warning.

The response step corresponds to the emergency step and consists of the limitation of injured, death, and damage amount by different organizations' interventions. The recovery step aims at going back to a normal situation. Among all these steps, the response is the most time-critical. The response's effectiveness requires that actions and the coordination between the different stakeholders be fast and efficiently set up. Responders need to know what to do in what circumstance and be prepared to do it, to obtain such ease during the response. Thus, the preparedness step is essential and crucial. Elaborating plans is both at the heart of the preparedness, on which the training is based and the source of an effective response.

Disaster management plans describe actions to be undertaken according to the situation, the responsible and actors of these actions, and the needed resources (such as equipment or vehicles) to achieve these actions. The plan elaboration is a complicated task. The plans must face a diversity of cases, conditions, and scenarios, meaning they must adapt to a diversity of situations (e.g., disaster type, geospatial, and financial situation). Therefore, their elaboration requires to take into account a complex set of information and the diversity of situations to face. Different planning levels exist to facilitate the development of the plan (i.e., strategic, tactical, and operational). The highest level provides guidelines to elaborate plans, whereas the lowest level provides precise plans adapted to the situation. The adaptation to the situation requires gathering geospatial data such as information about buildings, roads, critical infrastructures, river networks, and risk areas for a locality. Gathering such geospatial data to elaborate plans makes a face to interoperability problems between the different systems and formats used by the different stakeholders. Moreover, the plan elaboration requires a step of assessment to determine the effectiveness of a plan, above all for the lowest level plans that are the most difficult to elaborate for providing an effective response to the situation. The plan assessment requires many resources to test the plans in a broad diversity of situations and require gathering heterogeneous data.

To address the problem of data interoperability in the context of disaster management and face to preparedness challenges, the institute i3mainz in the University of Applied

Sciences in Mainz (Germany) has created the SemGIS[1] project financed by the German Ministry of Education and Research. This project has focused its research on this disaster management step on supporting (1) the gathering and the choice of data to use and (2) the plan elaboration. The first aspect addresses the question of data quality assessment and data evolution to support the choice of usable data in the context of disaster management. This aspect is not presented in this manuscript. The second aspect addresses the assessment of the plan's effectiveness according to all integrated data, whose plans from the different stakeholders. This second aspect is the subject of this manuscript.

2.1 Context and motivation

Disaster management preparedness is based on the preparation of the plans. It consists of organizing responsibilities, collaboration, and planning tasks, actions, and resources required to address the real risks of disaster. The plans aim at knowing what and how to do when a disaster happens. These plans also allow stakeholders to determine the type of training needed, but also what to monitor, when, and whom to alert for adequate early warning. Training and monitoring are two other primary steps of preparedness [Coppola, 2011], whose effectiveness depends on the plan preparation. The plan preparation is at the heart of disaster management, and their effectiveness is crucial. Therefore, they must be evaluated and improved until they are effective.

2.1.1 Disaster management plan assessment

Plan assessment is an essential step for the preparedness. It aims to know in which situations a plan is effective or not and define in which conditions a plan is applicable. As claimed by the author of [Larsson, 2008], "*a particular preparedness plan is to be motivated by means of at least an apprehension of its effectiveness*". This knowledge on plan applicability and effectiveness provides a base for improving the plans and completing them with new plans to face situations for which all plans are ineffective. This recursive process between plan elaboration and assessment aims at supporting the disaster management community to be well prepared.

A plan's assessment can be achieved through several criteria as organizational, maturity, and effectiveness criteria [Larsson, 2008]. Among these criteria, only the effectiveness criteria provide a numerical value. The other criteria as organizational and maturity are based on plan content and the general preparation (e.g., warning system, respect of laws). These criteria can be used to check the plan elaboration, but can also be used as a guideline not only during the plan elaboration but during the overall preparedness. On the contrary, the numerical measure of effectiveness anticipates an executed plan's effectiveness level in

[1] SemGIS website: `http://i3mainz.hs-mainz.de/de/projekte/semanticgis`

a specific scenario. This criterion is the one that provides the knowledge in which situations a plan is effective or not and, thus, allows its improvement. The plan's effectiveness depends on the achievement of its goals. The author of [Larsson, 2008] specifies that "*the effectiveness is assessed as to what extent the goals are accomplished, not being concerned with the specific activities undertaken in order to achieve the goals*". Therefore, the plan's effectiveness assessment requires (1) executing the plan, and (2) the goal definition to adapt its assessment. The definition of the plan's goal is a part of the plan's elaboration. Indeed, the plan elaboration requires to define what the plan addresses and what is expected from it. However, the execution of a plan is another step that requires a practical application of the plan. The practical application of the plan in the context of its assessment has two main challenges. The first challenge is the realism of a practical application, and the second challenge concern the need for a broad diversity of situations defined by practical applications. Broad diversity of situations is required to allow an exhaustive effectiveness assessment.

2.1.2 Practical application of a plan

It exists two main approaches to the practical application of a plan. The first one is a real exercise, and the second one is a computer simulation.

2.1.2.1 Real exercise vs Computer simulation

The real exercises require to reproduce scenarios of disaster and gather all stakeholders to simulate and test the plans' implementation according to real situations. It requires the set up of a lot of means and money to reproduce disaster scenarios. The higher the degree of realism is required for the exercises, the higher the costs are. The exercises have the advantage of allowing both the assessment of the plan's effectiveness and stakeholders' training simultaneously. Although there are a few biases during an exercise due to the responder's mindset that knows it is not real, real exercises are quite precise. Its primary disadvantage is a high organizing cost to gather all stakeholders and set up the most real disaster situations.

Computer simulation is used to assess a system through experiments. For the goal of plan assessment, experiments can simulate the plan-based disaster management system in a broad diversity of scenarios. Such simulation requires a system model expressed through informatics paradigms.

> *"A model is a representation of a system that can be defined and studied indirectly by helping to provide answers about it". [Rodrigues Da Silva, 2015]*

Three design steps constitute the simulation modeling process: firstly, the design of a conceptual model (i.e., model conceptualization), secondly, the design of the communicative model (i.e., representation of the conceptualization independent of a platform), thirdly,

the design of the programmed model (i.e., programming of the model), and finally, the design of the experiment model (i.e., experiments set up). According to [Nance, 1994], the conceptual model is the "*model, which exists in the mind of the modeler*", whereas the communicative model is a model representation that can be communicated to others. In the literature [Benjamin et al., 2006], the distinction between the conceptual model and its representation (i.e., the communicative model) is not always done. Therefore, the term conceptual model is more often used to describe its representation than its abstraction in the designer's mind. That is why the term conceptual model is used in the rest of the thesis to describe the representation of the conceptual model (i.e., "conceptual model" will be used instead of "communicative model").
The programmed model, also called the computational model, "*is a model representation that admits execution by a computer to produce simulation results*" [Nance, 1994]. Finally, the experimental model is formed by adding executable descriptions of the test environment to the programmed model that allows the simulation execution. It is based on different parameter configurations allowing the simulation of the system model in different scenarios. Computer simulation has the advantage of being a low-cost approach for testing a plan in a broad diversity of scenarios. Although good realism, simulations can be less precise than real exercise.

A proper evaluation of plans needs to be based on several exercises in different situations to provide extensive feedback to test plans addressing the situation and evaluate them. Nevertheless, the high cost of real exercises limits the number of iterations, the complexity, and the realism of scenarios. Although simulations have less precision than real exercises, the simulation precision can reach enough high precision to assess the plan's effectiveness. Therefore computer simulations are the most suited for practical plan applications in order to evaluate their effectiveness. Thus, computer simulations provide plan testing capability in a large and complex variety of scenarios for a low cost.

2.1.2.2 Suitable simulation technique and requirements

Computer simulations gather several types of techniques. It is essential to choose the simulation technique according to the simulation goal and the system to model. The authors of [Mishra et al., 2019] review the simulation techniques used in the literature for disaster management and examine the issues they address. The more commonly used identified techniques are discrete-event simulation, system dynamics, agent-based simulation, and Monte Carlo simulation. The examination of the issues addressed by these techniques shows the agent-based simulation as the most used during the preparedness to assess strategies and plans of disaster response [Mishra et al., 2019].

An agent-based simulation is a simulation based on a multi-agent system. According to the author of [Ferber, 1997], a multi-agent system is:

1. "*An Environnement E, it is in general a space with a metric.*
2. *A set of objects O. These objects are located, it means that for all objects, a specific time can associate a position in E. These objects are passive, they can be perceived, created, destroyed, and modified by Agents.*
3. *A set of agents A, which are particular objects. They are the active entities of the system.*
4. *A set of relations R, which link objects (and so, agents) between them.*
5. *A set of operations Op allowing for agents to perceive, produce, consume, transform, and manipulate objects in O.*
6. *Operators, which represent the application of operations and reactions of the world according to the modification (they are called universe laws).*"

They are also called multi-agent simulation (MAS). This technique is used to observe a complex system's emergence from modeling its components, corresponding mainly to agent behaviors and their interactions. Plan's effectiveness assessment requires the observation of the disaster management system. The disaster management system is a complex system, whose components are the responders and other stakeholders. Plans describe the role, the actions, and the interaction between the stakeholders. Thus, the agent-based simulation technique is the most suited to simulate a disaster management system from modeling its components based on plan description.

2.1.3 Challenges and objectives

Multi-agent simulation is the most appropriate technique to achieve the practical application of plans in diverse situations. Such a technique requires modeling and design steps to produce simulation experiments results. These results are the basis for studying the achievement of the plans' goals and evaluate their effectiveness. However, simulating the diversity of plans and situations implies a change of goals, stakeholders, actions, and interactions. According to the addressed goals, these changes imply an adaptation of the practical application through simulation and an adaptation of effectiveness assessment.

Simulation adaptation A certain level of simulation adaptation can be managed through the adaptation of the model parameter's variation. Some situation's characteristics can be configured as parameters to allow a simulation experiment in a diversity of situations. The majority of literature simulation approaches supporting disaster management preparedness are specific to a plan or a plan category. According to the category or plan addressed, they have a specific goal, design a simulation model, and a set of parameter configurations representing the simulation experiments to achieve their goal. However, the changes in goals, stakeholders, actions, and interactions cannot be managed only through parameter variations. They imply a change in simulation modeling and design due to the variation of experiment goals and the system's components. Therefore, the plan's effectiveness assessment's first challenge is the adaptation of simulation experiments modeling and design according to plan representation, describing its goal, the stakeholders, their actions, and interactions. This challenge gathers two sub-challenges. At the level of conceptualization, the challenge is ***the adaptation of the conceptual model and experiments representation to the diversity of plans***. It consists of providing a multi-agent model that can represent the diversity of plans in the agent paradigm and represent the diversity of scenarios through experiments to allow plan assessment. This thesis aims at addressing this challenge by *increasing the flexibility to allow the assessment of the diverse plans of disaster management through their simulation*. At the level of programming, the challenge is to have ***adaptable and reusable multi-agent simulation components for disaster management***, allowing the adaptation of the programmed model. The objective to address this challenge is *the increase of the extensibility by allowing the diversity of plans to be processed*.

Plan representation These two challenges linked to the simulation adaptation depend on the plan definition. Its detail level defines the detail level of the simulation. The simulation conceptualization is generally made ad hoc from plans and related data provided by disaster management experts. The conceptualization is achieved in the agent paradigm that requires a computer scientist designer. This requirement implies that a computer scientist designer must make new modeling at each new description level of plan or a new change in plan. The plan representation issues directly in agent paradigm are (1) the no accessibility of modeling from disaster management expert to modify some aspects of the plan representation and (2) the conceptualization biases generated by new modeling that does not guarantee a uniform simulation conceptualization for all plans. A no uniform simulation conceptualization for all plans can produce biases in their assessment and, thus, a meaningless comparison between their effectiveness values. Therefore, the first enunciated challenge is to adapt the conceptual model and experiments representation to the diversity of plans and guarantee a uniform conceptualization. Such uniformity of conceptualization could be provided through an automatic process of conceptualization that would follow the same conceptualization method. The enunciated issues generate a new challenge related to the need for ***plan representation that must be understood both by the domain expert to model them and by the machine to be processed***. This thesis aims at *increasing the expressivity and interoperability to gather disaster management knowledge and allow disaster management experts to assess plans according to their plan definition* to address this challenge.

Assessment adaptation Simulation experiments provide the practical application, whose results are necessary to assess the plan's effectiveness. The fourth challenge of the plan's effectiveness assessment is ***the adaptation of results analysis according to the plan's goals to provide an effectiveness value***. An exhaustive plan assessment in a broad diversity of situations can provide extensive effectiveness characterizing the situation category. In the case of different and extensive effectiveness, providing a global effectiveness value would be meaningless. In this case, it is suitable to provide effectiveness values associated with the situation description characterizing the effectiveness value. Therefore, the last objective of this thesis is to *represent the effectiveness values associated with their applicability context.*

2.2 Problem statement

From the challenges and objectives enunciated in the previous section, it is relevant to search a method that adapts multi-agent simulation automatically by

- conceptualizing the multi-agent simulation model and
- designing the programmed model based on adaptable and reusable multi-agent simulation components,
- according to a plan representation understandable by disaster management community and machines
- to assess and represent the plan's effectiveness associated with its applicability context.

Therefore, this thesis addresses the following problem question:

How to adapt multi-agent simulation automatically from the semantic representation of disaster management plans for their effectiveness assessment?

2.2.1 Issues

Issues to solve the problem question comes from limits of works related to:

1. the multi-agent simulation adaptation,
2. the plan representation, and
3. the representation of the plan's effectiveness associated to a situation definition.

2.2.1.1 Multi-agent simulation adaptation

In the literature, approaches related to the increase of multi-agent simulation adaptability propose meta-models and ontologies to allow the design of a diversity of multi-agent simulation models. A meta-model is a modeling model:

> *"it defines the structure of a set of models conforming to a given syntax and semantics." [Rodrigues Da Silva, 2015]*

An ontology is defined as

> *"a formal, explicit specification of a shared conceptualization" [Studer et al., 1998].*

These meta-models and ontologies are often used as a base for generative programming.

> *"Generative Programming (GP) is an attempt to manufacture software components in an automated way by developing programs that synthesize other programs." [Cointe, 2005]*

The generative programming is based on a set of multi-agent simulation components combined and parameterized to design the simulation code. These approaches are used to automate simulation development and have the advantage of facilitating interoperability with other systems.

Conceptualization of the multi-agent simulation model Among the meta-models and ontologies for multi-agent simulation of the literature, the ontology presented by the authors of [Christley et al., 2004] is the most relevant to adapt simulation for plan assessment. This ontology has the advantage of providing high-level concepts for multi-agent simulation, allowing the representation of an application case diversity. However, it does not contain specific concepts for disaster management simulation.

Designing the programmed model based on adaptable and reusable multi-agent simulation components A programmed model depends on a simulation platform with which it can be executed. Among the different studied multi-agent simulation platforms, the GAMA platform [Taillandier et al., 2019] appears to be the most suitable for the simulation of the plans. It has the advantage of enabling real-world and geographic information systems (GIS) representation, large scale simulations, scientific simulation, general-purpose agent-based simulation scheduling and planning, natural resources, and environment. However, the GAMA platform does not have an agent behavioral set specific to the disaster management domain.

2.2.1.2 Plan representation

Ontologies and meta-models provide a semantic model that allows the representation of a diversity of disaster management plans. Therefore, ontologies and meta-models related to disaster management are reviewed for plan assessment. Besides of a formalization, plan representation requires the representation of the specific plans. Much knowledge related to these plans is stored into heterogeneous geospatial data. Therefore, the interoperability increase for gathering disaster management knowledge and representing plans requires the extraction of knowledge from heterogeneous geospatial data.

Plan formalization Among the reviewed ontologies and meta-models, two approaches appear as the most relevant for plan representation: the meta-model presented by [Othman et al., 2014] and the ontology Emergel [Casado et al., 2015]. The meta-model of

[Othman et al., 2014] has the advantage of providing high-level concepts allowing the definition of a wide variety of plans. However, this meta-model is not expressed in OWL and does not contain specific concepts to define plan content. The ontology Emergel [Casado et al., 2015] provides a complete vocabulary for plan contents but does not allow the representation of plans.

Knowledge extraction from heterogeneous geospatial data Among the studied approaches of knowledge extraction from data, the Datalift project [Scharffe et al., 2012] proposes the most interesting automatic approach. However, Datalift creates individuals with annotations, which is not the most adapted RDF representation to integrate knowledge. An RDF representation with individuals linked through properties would be more suitable.

2.2.1.3 Representation of the plan's effectiveness associated with its applicability context

The representation of the plan's effectiveness associated with its applicability context requires to cluster situations for which a plan has similar effectiveness. In the case of a large-scale plan application or extensive testing of such plans, the clustering should be unsupervised to avoid human analysis prone to error when the number of situations and characteristics is significant. Among unsupervised clustering, the cure approach [Guha et al., 2001] appears to have the best compromise between computational complexity and clustering quality. However, this clustering application on simulation results leads to over-segmentation by considering criteria that do not impact the plan's effectiveness.

2.2.2 Requirements

Solving the problem question of this thesis requires to fulfill the limits of related works presented previously. Fulfilling these limits implies a set of requirements related to each identified limit.

2.2.2.1 Multi-agent simulation adaptation

The works related to the multi-agent simulation adaptation have identified two relevant approaches: the ontology of multi-agent simulation modeling of [Christley et al., 2004] and the GAMA platform [Taillandier et al., 2019]. The limit of the approach of [Christley et al., 2004] requires a ***specification of concepts for disaster management simulation*** to complete high-level concept of multi-agent simulation. The lack of agent's skills for disaster management of the GAMA platform [Taillandier et al., 2019] requires an ***extension of agent's behaviors for disaster management actions*** through the addition of an external plugin call "skills". These two first requirements correspond to essential components for the simulation adaptation. However, the simulation adaptation also requires

an automatic process to generate the conceptual simulation model according to the plan definition and ***an automated approach to develop the programmed simulation model***.

2.2.2.2 Plan representation

The studied works related to the plan representation have identified two relevant approaches for its formalization: the meta-model presented by [Othman et al., 2014] and the ontology Emergel [Casado et al., 2015]. In addition to these approaches, the Datalift project [Scharffe et al., 2012] has been identified as interesting to extract knowledge from data. The approaches of [Othman et al., 2014] and [Casado et al., 2015] have complementary limits: the first one has no low-level concepts to describe the plan's content, whereas the second one has no high-level concept for plan description. These approaches do not provide a formalization to combine high-level and low-level concepts to represent disaster management plans. Therefore, it is necessary to provide ***a disaster management knowledge model for plan assessment*** to fix the limit of high-level and low-level concepts combination to represent the disaster management plan. The limit of the Datalift project [Scharffe et al., 2012] makes it difficult to integrate extracted knowledge into the ontology for disaster management plan assessment. Therefore, ***a new automatic approach for knowledge extraction from data*** is required to allow their integration.

2.2.2.3 Representation of the plan's effectiveness associated with its applicability context

The approach of unsupervised hierarchical clustering [Guha et al., 2001], identified as the most suitable, provides an issue of over-segmentation. This issue comes from the consideration of criteria that do not impact the plan's effectiveness. Therefore, an ***identification of relevant criteria impacting the plan's effectiveness*** is required to overcome this issue.

2.3 Proposed approaches

This manuscript proposes an approach of knowledge-driven multi-agent simulation engineering for assessing the effectiveness of disaster management plans to fix the issue mentioned in section 2.2.1 by meeting the requirement presented in the section 2.2.2. The first subsection presents the proposed approaches to meet requirements, and the second one describes the approach for assessing the effectiveness of a response plan.

2.3.1 Proposed approaches to meet requirements

The proposed approaches to meet requirements are related to (1) the multi-agent simulation adaptation, (2) the plan representation, and (3) the representation of the plan's effectiveness associated with its applicability context.

2.3.1.1 Multi-agent simulation adaptation

Adaptability to simulate and assess the diversity of plans representation This thesis proposes to fix the issue of simulation adaptability to assess the diversity of plans representation by (1) specifying concepts for disaster management simulation, (2) generating a conceptual simulation model according to the plan definition and (3) generating the corresponding programmed simulation model. Three approaches are presented to address each of these requirements. The first approach is the SemMAS (Semantic Multi-Agent Simulation) ontology, which combines high-level concepts representing the domain of multi-agent simulation. It is inspired by the work of [Christley et al., 2004] with low-level concepts representing the disaster management simulation and related to the GAMA platform. The second approach is a knowledge-driven simulation modeling that uses the plan representation to generate the simulation conceptualization represented into the SemMAS ontology. The third approach is a knowledge-driven simulation programming for the GAMA platform that generates the simulation code from the content of the SemMAS ontology to execute simulations with the GAMA platform.

Adaptability and reusability of multi-agent simulation components for disaster management An extension of the agent's behaviors for disaster management actions is proposed to solve the adaptability and reusability issues of multi-agent simulation components. The approach to fulfill this requirement corresponds to develop agent's new skills for the GAMA platform. These new skills and their associated actions are represented in the SemMAS ontology.

2.3.1.2 Plan representation

This thesis proposes to solve the issue of plan and scenario representation expressed directly in a multi-agent simulation paradigm by representing them through an ontology for disaster management plan's assessment and integrating knowledge extracted from heterogeneous data. The SemDM (Semantic Disaster Management) ontology combining high-level and low-level disaster management concepts is proposed to fulfill the first requirement. High-level concepts of this ontology are inspired by the meta-model presented by [Othman et al., 2014], whereas low-level concepts are inspired by the ontology Emergel [Casado et al., 2015]. An automatic integration approach of knowledge extracted from heterogeneous data is proposed to integrate knowledge stored through data in the SemDM ontology. This approach is based on natural language processing and the geospatial dimension.

2.3.1.3 Representation of the plan's effectiveness associated with its applicability context

Finally, this thesis fixes the limit related to the evaluation of changes in the description of plans by identifying relevant criteria impacting the plan's effectiveness. An analysis based on a clustering combination applied to the different simulation criteria is proposed to satisfy this requirement.

2.3.2 Approach for assessing the effectiveness of a response plan

The approach defended in this thesis combines the previously presented approaches to assess disaster management response plans. It is composed of four steps : (1) modeling disaster management knowledge, (2) modeling simulations, (3) designing simulations, and (4) analyzing simulation results based on clustering. First, explicit expert knowledge and data is used to create a knowledge model for disaster management. Second, simulation models are conceived based on the knowledge model. Thirdly, generative programming is used for simulation design. Finally, simulation results are used to calculate the plan's effectiveness for each simulation. Unsupervised learning clustering identifies the application context related to the calculated effectiveness. The effectiveness and the associated applicability context enrich the initial knowledge model.

2.4 Contributions

The approaches proposed in the previous section bring its main contributions to multi-agent simulation engineering. They also bring some contributions to the Semantic Web domain and the domain of disaster management.

2.4.1 Contribution to the disaster management community

The whole work of this thesis is applied to the domain of disaster management. That is why this work brings two contributions in this domain: the first one is to **facilitate gathering disaster management knowledge for the collaborative work**, and the second one is **the knowledge enrichment from the assessment of disaster management plans**.

Firstly, it contributes to providing tools, facilitating the information and knowledge shareability with other systems, thanks to the semantic technologies. The authors of [Bharosa et al., 2010] highlight the benefits of Inter-Organizational Information-Sharing

Systems to solve disaster management coordination problems. The design of such systems requires to solve technical problems of interoperability. The automatic integration process [Prudhomme et al., 2017a, Prudhomme et al., 2020a] and abilities provided by the use of semantics in GIS [Homburg et al., 2017] provides an added value for the systems interoperability. This approach proposing the integration of knowledge extracted from heterogeneous data enables the use of data from other systems as inputs for the plan's assessment. Thus, it facilitates the collaborative work of the disaster management community and can promote stakeholders to share information to solve difficulties of integration and synergy across institutional networks [McConnell and Drennan, 2006].

Secondly, it provides a system that supports disaster management preparedness [Prudhomme et al., 2017c, Prudhomme et al., 2018]. This system's contribution is to support the plans' effectiveness assessment by providing its representation with its applicability context. The use of simulation to support plan assessment enables a low-cost and active disaster management community preparation. Its benefit is to enable better preparation for disaster management community, which is often more a symbolic readiness than a reflection of operational realities [McConnell and Drennan, 2006].

2.4.2 Contribution to the Semantic Web

The system based on knowledge-driven simulation uses semantic technologies to design and manipulate ontological models of disaster management and multi-agent simulation. However, existing approaches have some limits to fit the requirements of the system. These limits previously enunciated have led to contributions in the domain of semantics. This domain's contributions improve the gathering of information and knowledge of disaster management and extend reasoning capabilities for allowing simulation conceptualization.

The requirement of gathering all the relevant knowledge and information related to disaster management plans into a knowledge base inducts a need to retrieve heterogeneous data from a large variety of data sources. The first contribution in this domain is to **facilitate the access of a large variety of data sources through the use of a semantic catalog** [Prudhomme et al., 2016]. The second contribution is **an automatic integration process of knowledge extracted from heterogeneous data that facilitates the interoperability between systems** [Prudhomme et al., 2017a, Prudhomme et al., 2020a]. This contribution made in collaboration with others facilitates the enrichment between data and Semantic Web.

The process of simulation conceptualization is achieved in the SemMAS ontology from the SemDM ontology and through reasoning. The **ontologies SemDM and SemMAS** are two contributions to the Semantic Web. The first one allows the representation and the sharing of a disaster management plan, whereas the second one allows the representation of disaster management simulation models and its configurations [Prudhomme et al., 2019b].

The set of rules [Prudhomme et al., 2019b], on which the reasoning for simulation conceptualization is based, uses new implemented rule built-ins. These **rule built-ins** are used to increase reasoning capabilities for the simulation conceptualization [Prudhomme et al., 2019a].

2.4.3 Contribution to multi-agent simulation engineering

The main contribution of this thesis is located in multi-agent simulation engineering. Indeed, this thesis brings three contributions, which are (i) **a knowledge-driven approach for the simulation conceptualization and design**, (ii) **a knowledge-driven generative programming of simulation for the GAMA platform**, and (iii) **an extension of an agent's skills in disaster management for the GAMA platform**.

The knowledge-driven simulation modeling proposed in this thesis provides the capacity to automate the simulation conceptualization and design [Prudhomme et al., 2019b]. This approach produces a conceptual model of simulation independent of the platform and a representation of the GAMA platform's programmed model. This contribution allows the modeling adaptation from diverse plan representations through a uniform method avoiding different simulation conceptualization biases for plan assessment. Besides, the use of ontology brings benefits to facilitate the sharing and the reusability of simulations models in the multi-agent simulation community.

The platform chosen to execute simulation experiments is the GAMA platform [Taillandier et al., 2019]. A knowledge-based generative programming process has been developed to implement the code corresponding to its representation made in the SemMAS ontology and specific to this platform. The generated code allows the execution of simulation experiments [Prudhomme et al., 2019b, Prudhomme et al., 2019a].

This generative programming is based on an agent's skills and reflex. An extension of an agent's skills in disaster management for the GAMA platform has been developed to fulfill the generative programming requirements. These contributions provide benefits in the productivity of simulation development and avoid different programming biases for plan assessment.

2.4.4 Publications

The works of this thesis have been communicated through publications in journals and conferences, but also through presentations both to the scientific community and to the general public.

2.4.4.1 Journals

[Prudhomme et al., 2020a] Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Cruz, C., and Roxin, A.-M. (2020a). Interpretation and automatic integration of geospatial data into the semantic web. *Computing*, 102(2):365–391

[Prudhomme et al., 2020b] Prudhomme, C., Roxin, A., Cruz, C., and Boochs, F. (2020b). Modélisation sémantique et programmation générative pour une simulation multi-agent dans le contexte de gestion de catastrophe. *Revue Internationale de Géomatique*, Lavoisier.

[Prudhomme et al., 2019b] Prudhomme, C., Cruz, C., and Boochs, F. (2019b). Semantic and Logic Modeling of Disaster Simulation for Multi-agent Systems. *International Journal of Modeling and Optimization*, 9(4):198–204.

[Prudhomme et al., 2018] Prudhomme, C., Cruz, C., Roxin, A., and Boochs, F. (2018). A Framework to Improve the Disaster Response Through a Knowledge-Based Multi-Agent System. *International Journal of Information Systems for Crisis Response and Management*, 9(3):96–109.

[Homburg et al., 2017] Homburg, T., Prudhomme, C., Boochs, F., Roxin, A., and Cruz, C. (2017). Integration, quality assurance and usage of geospatial data with semantic tools. *gis.Science - Die Zeitschrift fur Geoinformatik*, 3:91–96.

2.4.4.2 Conferences

[Prudhomme et al., 2019a] Prudhomme, C., Cruz, C., and Boochs, F. (2019a). Modélisation sémantique et logique pour une simulation multi-agent dans le contexte de gestion de catastrophe. In *International Francophone Conference SAGEO (Spatial Analysis and Geomatics)*, Clermont Ferrand, France.

[Prudhomme et al., 2017c] Prudhomme, C., Roxin, A., Cruz, C., and Boochs, F. (2017c). Towards the design of respond action in disaster management using knowledge modeling. In *Lecture Notes in Business Information Processing*, volume 301, pages 168–174. Springer.

[Prudhomme et al., 2017b] Prudhomme, C., Homburg, T., Ponciano, J. J., Boochs, F., Roxin, A., and Cruz, C. (2017b). Automatic integration of spatial data into the semantic web. In *WEBIST 2017 - Proceedings of the 13th International Conference on Web Information Systems and Technologies*, pages 107–115. **Best Paper Award**.

[Homburg et al., 2016] Homburg, T., Prudhomme, C., Würriehausen, F., Karmacharya, A., Boochs, F., Roxin, A., and Cruz, C. (2016). Interpreting heterogeneous geospatial data using semantic web technologies. In *Computational Science and Its Applications – ICCSA 2016*, pages 240–255. Springer International Publishing.

[Prudhomme et al., 2016] Prudhomme, C., Roxin, A., Cruz, C., and Boochs, F. (2016). Semantic Catalogue to Manage Data Sources in Disaster Management System. In *International*

Conference on Informatics in Economy, Ie 2016: Education, Research & Business Technologies, pages 225–230.

2.4.4.3 Other communications

Presentation Prudhomme, C., Roxin, A., Cruz, C., and Boochs, F. (2018). Vers la conception d'une action de réponse pour la gestion de catastrophe, utilisant la modélisation de connaissances. *International Francophone Conference SAGEO (Spatial Analysis and Geomatics) 2018*, Montpellier, France.

Poster Homburg, T., Prudhomme, C., and Boochs, F. (2018). Semantic Geographic Information System: Integration and management of heterogeneous geodata. *Conference for Expert exchange Geoinformation 2018 by GeoNet.MRN.*

Poster Prudhomme, C. (2017). Katastrophenmanagement: Die geflutete Stadt. *Science Market (Wissenschaftsmarkt)*, Mainz.

Presentation Homburg, T. and Prudhomme, C. (2016). Vorstellung SemGIS Projekt - Einblick und Status. *The German Conference of geodesy students (KonGeoS - Konferenz der Geodäsiestudierenden).*

2.5 Thesis overview

This thesis is described through eight chapters whose content is summarized in this section.

Chapter 2 introduces the general context of disaster management with its critical business points. It then explains the context of disaster management plan assessment and the needs of practical applications that motivate multi-agent simulation. The plan assessment through multi-agent simulation generates challenges and objectives on which the problem statement is based. The problem statement presents issues associated with objectives and requirements to solve the issues. Then, the proposed approaches and contributions of this thesis are presented. Finally, it provides an overview of the thesis' content.

Chapter 3 aims to present works related to the assessment of disaster management plans. It presents the application domain of this thesis, which is disaster management, to detail the business problem of plan assessment and highlights the limits of existing approaches related to the plan assessment. It then reviews engineering techniques, components, and platforms for multi-agent simulations, which are the most suitable approaches to experiment and assess disaster management plans. These reviews aim to search suitable approaches and identify the lacks to reach the objectives. This chapter presents then, the knowledge engineering domain to allow the representation of the disaster management knowledge, whose plans' description. It consists of reviewing the existing ontologies of this domain and approaches to integrate knowledge extracted from data. Finally, the chapter concludes on limits to overcome an end-to-end approach allowing the disaster management community to assess their plans.

Chapter 4 aims at presenting the conceptual analysis achieved to solve the problem question of this thesis. It first outlines the requirements for plan assessment arising from the limitations of the current works. This chapter then explains the proposed approaches to meet the identified requirements. Finally, it explains how and why these different approaches are organized together to provide a method for assessing the plan's effectiveness.

Chapter 5 aims at presenting the architecture used for the implementation of the method presented in the previous chapter. This chapter presents its four components: a simulation platform, a knowledge base, a client, and a processing server. The simulation platform description explains the GAMA platform's specificities and an agent's developed disaster management skills. It describes the content of the knowledge base through the knowledge modeling of the SemDM and SemMAS ontologies. It then presents the client allowing knowledge integration and the plan's assessment by requesting the processing server. Finally, it explains the role of the processing server, which interacts and uses the other components.

Chapter 6 aims to present the method's implementation achieved by the processing server. The method comprises four main steps: the modeling of the studied disaster management system, the simulation modeling, the simulation design, and a clustering-based analysis of the simulation experiment results. The chapter presents firstly, the studied system modeling achieved by the integration of knowledge extracted from data and the rule-based reasoning on the SemDM ontology. Secondly, the simulation modeling is explained through the processes to generate the conceptual simulation model from the SemDM ontology and the programmed simulation model's representation for the GAMA platform. The chapter then presents the implementation of the generative programming process and the simulation execution that achieve the simulation design. Finally, it explains the process of plan's assessment based on the clustering-based analysis and the enrichment of the SemDM ontology with the plan's effectiveness representation.

Chapter 7 aims at illustrating the results of the method's implementation on a use case, step by step. This chapter's case study is the French NOVI plan, whose goal is to manage numerous victims (more than one hundred victims) of a disaster. This plan is assessed through different scenarios based on three configurations of a disaster event impacting the town Montbard (France). This chapter explains the case study firstly through the description of the NOVI plan and the three scenarios. Secondly, it presents the modeling of the disaster management case study, through knowledge integration and rule-based reasoning on the SemDM ontology. From the SemDM ontology, the results of the conceptual and programmed simulation modeling are presented. Then, it shows the program, experiments, and results obtained by the simulation design step. Finally, it describes the results provided by the clustering-based analysis process.

Chapter 8 aims at evaluating the proposed approach of this thesis. In the absence of similar work for disaster management plan evaluation, this chapter focuses its evaluation on the proposed approach's multi-agent simulation model. The simulation model is the linchpin of the approach: it is the result of the simulation conceptualization and design from the disaster management knowledge integration, and its results are the source of the clustering-based analysis for plan assessment. Therefore, this chapter presents the evaluation method firstly through the used criteria metrics and their computation. Secondly, it explains and describes the application of the evaluation method to the case study. Finally, it discusses the strengths and weaknesses of the generated simulation model.

Chapter 9 summarizes the contributions bring by this manuscript. It then discusses the benefits and limits of the proposed approach. Finally, it presents the perspectives in the short, medium, and long term.

3 Related work

Contents

3.1 Disaster management **48**

3.1.1 A cycle of four stages 48

3.1.2 Preparedness 51

3.1.3 Simulation approaches supporting preparedness 58

3.2 Multi-agent simulation **62**

3.2.1 Multi-agent simulation engineering 63

3.2.2 Multi-agent simulation model components 65

3.2.3 Multi-agent simulation platforms 72

3.2.4 Discussion 76

3.3 Knowledge engineering **78**

3.3.1 Knowledge representation for disaster management 82

3.3.2 Knowledge extraction and integration 93

3.4 Discussion **97**

This chapter presents works related to the evaluation of disaster management plans. It refers to different fields, which cannot all be presented here; therefore, complementary information is given in the appendix A. This complementary information can be consulted before reading this chapter or all along with the reading of this chapter. For further details all along with the reading, appendix's sections are referred to in this chapter according to the need for its understanding.

The first section presents the application domain of this thesis, which is the disaster management, to firstly highlight the business problem of plan assessment and secondly, highlight the limits of existing approaches related to the plan assessment. The second section reviews engineering techniques, components, and platforms for multi-agent simulations, which are the most suitable approaches to experiment and assess disaster management plans. These reviews aim to search suitable approaches and identify lacks of existing approaches to reach the objectives defined in the first section. The third section presents the knowledge engineering domain to allow the representation of the disaster management knowledge, whose plans' description. It firstly reviews the existing ontologies of this domain and secondly reviews approaches to integrate knowledge extracted from data. Finally, the chapter concludes on limits to overcome for an end-to-end approach allowing the disaster management community to assess their plans.

3.1 Disaster management

This section aims to provide necessary background about disaster management. Therefore, the first subsection 3.1.1 presents this domain as a cycle of four steps. From the observed challenges of disaster management, specific attention is given to the Preparedness step in the second subsection 3.1.2. The Preparedness aims at preparing resources, people, organizations, and collaborative work according to risk estimation. This preparation is done through the conception of plans. These plans provide a framework to know who and how to react. To further understand the preparation, it describes the three categories of plans: strategic, operational, and tactical [Federal Emergency Management Agency, 2010]. Each of these categories has its level of detail: the strategic plans are general; the operational plans are specific plans to an administrative area; the tactical plans are situation-specific plans. It then, presents approaches to experiment and assess plans. The third subsection presents the approaches of simulation for disaster management. This subsection aims at identifying limits of these approaches for the experiment and assessment of plans. It allows the introduction of the next sections of the chapter.

3.1.1 A cycle of four stages

The disaster management is an infinite cycle which aims at continuously improving the resilience and efficiency to face disasters. Among the disaster management community, there are some variations on the number and the name of the steps composing the disaster management. These variations are mainly due to diverse levels of description. Our presentation follows the most widespread vision of this cycle presented in [Coppola, 2011, BBK, 2015] and which consists of four steps: mitigation, preparedness, response, and recovery. Figure 3.1 illustrates this cycle.

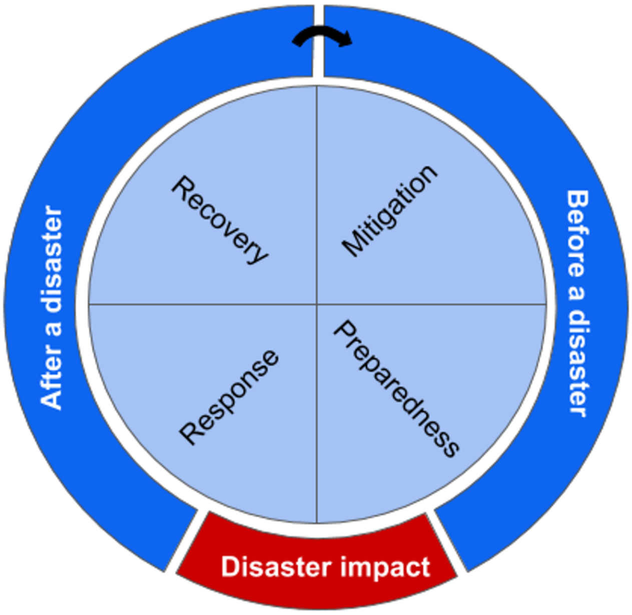


Figure 3.1: Cycle of disaster management

3.1.1.1 Mitigation

The Mitigation step consists of the identification of risks. According to [Coppola, 2011], the risk is assessed according to the hazard likelihood and the hazard impact. The hazard corresponds to a potential disaster. Its likelihood is often calculated according to the historical events of this hazard. The calculation also considers the changes, which could increase or reduce this hazard. The impact depends mainly on the vulnerability of the population and the infrastructure. The risk assessment aims at reducing the vulnerability of the population by the prevention. The Mitigation also aims at lowering infrastructure vulnerability. This vulnerability reduction comes from the creation of the norms (e.g., anti-seismic norms) and structural reinforcements. Its goal is to avoid disaster or at least reduce its impact.

3.1.1.2 Preparedness

The Preparedness step precedes the disaster event. This step aims to elaborate on how to react and act in front of disaster impacts. This step's first activity is the planning of a response that implies using a risk analysis system to create a plan adapted to potential disasters. Each disaster is decomposed into different levels of severity to identify the different needs according to its severity level. The second activity corresponds to the preparation of equipment and resources which have been identified in the action plan. The equipment and resources have a determining role in the disaster; if they are missing, the consequences can severely impact human lives. The third activity is the training of the response actors

according to their role and the actions which could be required during a disaster — some examples of training concern the evacuation, the management of volunteers, or injuries. This training aims at learning the way to act, but the learning is not enough, and practice is required. Consequently, the fourth activity is the exercise, which aims at training the population and the responders, but also assessing plans. The last activity is the monitoring of the potential hazards to early warning.
The activities of training, exercises, resource preparation, and early warning depend on the first activity, which is the conception of plans. This one requires an overview of the risks and available means both in stakeholders, who have specific capacities, and resources that can be used.

3.1.1.3 Response

The response step begins immediately when a disaster happens or is imminent and corresponds to the emergency or crisis management. Its goal is to limit the number of casualties, damages, and impacts on the environment. The response step requires to evaluate the situation: what areas are affected, what type and number of people have been affected, what damages, what needs are present and what organizations, resources, and equipment are available to act. Besides, this step requires to use situation analysis techniques to identify solutions that must be applied. The management between the situation needs and capacities provided by the different actors of the response remains a difficult task. This task consists of decision-making to manage resources and solve problems resulting from a disaster. The collaboration between the different actors plays a crucial role in the response efficiency. Without cooperation, actors would lose precious time.

3.1.1.4 Recovery

Two phases, one in the short term and one in the long term, compose the recovery step. The first one consists of providing the necessary to live for each affected people. This phase is mainly managed by the organizations of humanitarian aid that also consists of logistic work to retrieve a minimum of communication networks. The second one consists of returning to a normal situation. The achievement of this goal requires an evaluation of the loss and damages caused by a disaster. This evaluation allows then cleaning, temporary sheltering of population, and rebuilding. These three stages require an organization and planning to be as efficient as much as possible. Moreover, the planning of rebuilding gathers many constraints like time, cost, and improvement according to past weaknesses.

3.1.1.5 Discussion

The response phase is the most critical in terms of time and challenge since human lives depend on the effectiveness of this phase. This effectiveness depends on the collaboration between the different stakeholders. The collaboration has two aspects: the coordination and the cooperation [Gulati et al., 2012]. Coordination is the act of managing interdependencies between activities performed to achieve a goal [Malone and Crowston, 1990]. The three components of the coordination are activities that correspond to the goal decomposition, actors who are selected to be assigned to an activity, and interdependency, which must be managed [Malone and Crowston, 1990]. The coordination aspect is typically the task carried out during the preparedness. It corresponds to the planning task to organize actions between the different stakeholders. The cooperation is defined as "joint pursuit of an agreed-on goal(s) in a manner corresponding to a shared understanding about contributions and payoffs" [Gulati et al., 2012]. This second aspect is essential to provide a common view both during the preparedness, to identify the resources and capabilities of each stakeholder, and during the response to have a common view of the crisis.

Coordination and cooperation are based on information and knowledge sharing. However, the study of the literature and field exercises made by [Bharosa et al., 2010] highlights the challenges and obstacles in sharing and coordinating information that reduces the effectiveness of disaster response. During a multi-agency disaster response, collaboration obstacles appear at community, agency, and individual levels. Although it exists plans for each of these levels, some problems as organizational silos, conflicting role structure, a mismatch between goals, allocation of responsibility, or inability to determine what should be shared, shows a lack in the preparation process. Indeed such problems appear during the response step and impact its efficiency, but they must be carried out upstream during the preparedness step. To identify the limits of the preparedness, the next subsection explains the preparedness process.

3.1.2 Preparedness

The preparedness step aims at preparing the collaboration between the different stakeholders of disaster management. This collaboration must be anticipated through activities and responsibilities planning to coordinate stakeholders.

3.1.2.1 Preparedness cycle

The Federal Emergency Management Agency (FEMA)[1] [Federal Emergency Management Agency, 2010] presents the preparedness step as a

[1]Federal Emergency Management Agency (FEMA): `https://www.fema.gov/`, visited on 2020-09-22

cycle of five sub-steps: Plan, Organize/Equip, Train, Exercise, Evaluate/Improve. Figure 3.2 illustrates the cycle of preparedness. The heart of the preparedness is the planning sub-step that guides the other sub-steps.

Figure 3.2: Cycle of preparedness according to [Federal Emergency Management Agency, 2010]

Planning aims at managing risks by considering all hazards and threats. It must take into account the whole population and its needs to provide an adapted solution. It should also be flexible enough to address both traditional and catastrophic incidents. Planning must be a collaborative process between all community stakeholders to identify the missions and support goals. Planning identifies tasks, allocates resources to accomplish those tasks, and establishes accountability, according to a description of the anticipated environment for action.

Such collaborative preparation makes the Preparedness crucial for disaster management. The entire structure of this phase stands on the development of plans. Preparedness aims at ensuring a preparation adapted to needs. In disaster management, there are several needs: a need for an organizational structure, a need for risk management, and a need for situation management. Addressing such needs results in different levels of plans. These levels of plans allow providing a common and homogeneous structure between different administrative areas. They aim to facilitate their collaboration and define a specific preparation adapted to both the needs of each of them and the disaster severity. These different levels of plans are detailed in the following subsection.

3.1.2.2 Plan preparation

The phase of planning consists firstly of determining the general needs of disaster management in terms of organizational structure (e.g., command post, responsibilities), tasks (e.g., informing the population, closing roads, build walls against water), and roles (e.g., communication management, security management, flood protection management). Secondly, it is necessary to plan the disaster management operations and their coordination according to levels of risk (e.g., orange and red risk levels, which are the higher risk levels of the four French risk levels) specific to an administrative area. The last step is to plan the implementation of operations adapted to the current disaster situation. These three steps correspond to planning at three levels of detail: strategic, operational, and tactical.

Strategic plan The strategic plans are guidelines and general plans defining a standard process and a standard structure for a jurisdiction. The goal of this standard process and structure is to facilitate the collaboration between different administrative areas through a standard model of disaster management while allowing a plan design specific to each of them and the severity of a disaster. This type of plan is defined by [Federal Emergency Management Agency, 2010] as following: "Strategic plans describe how a jurisdiction wants to meet its emergency management or homeland security responsibilities over the long-term. These plans are driven by policy from senior officials and establish planning priorities."
A strategic plan defines the general tasks to prepare and achieve in case of disaster, the roles, and the responsibilities associated with these tasks. Besides, it provides an organizational structure by defining command centers and a hierarchy of managers. The role of a commander is generally at the charge of an administrative responsible or an executive officer of an organization playing a crucial role in disaster management. For example, a mayor can be defined as the director of rescue operations, and the chief of fire brigade can be defined as the rescue commander for a disaster at the level of a municipality.

Operational plan The operational plans are plans prepared by each administrative area to address specific needs. This planning depends on means of the administrative area and must respect the strategic plans of its jurisdiction. This plan aims at organizing the collaborative work between the different stakeholders of disaster management. This type of plan defines people and organizations associated with a role, details tasks, and actions to perform by a function to address risks of the locality. They also contain an estimation of the resources required for their achievement. The locality inventory of resources that are available in case of disaster accompanies the operational plans.
An operational plan is defined by [Federal Emergency Management Agency, 2010] as following: "Operational plans provide a description of roles and responsibilities, tasks, integration, and actions required of jurisdiction or its departments and agencies during emer-

gencies. Jurisdictions use plans to provide the goals, roles, and responsibilities that a jurisdiction's departments and agencies are assigned, and to focus on coordinating and integrating the activities of the many responses and support organizations within a domain. They also consider private sector planning efforts as an integral part of community-based planning, and to ensure efficient allocation of resources. Department and agency plans do the same thing for the internal elements of those organizations. Operational plans tend to focus more on the broader physical, spatial, and time-related dimensions of operation; thus, they tend to be more complex and comprehensive, yet less defined, than tactical plans."
The operational plans depend on risk estimation. They coordinate actions and tasks according to specific events (e.g., a certain level of water during a flood) or a particular warning risk level. These tasks can be described by other plans designed by an organization in charge of the task. Moreover, stakeholders (people and organizations), resources, tasks, and actions are located.

Tactical plan Finally, the tactical plans are the planning done during the response, corresponding to the resources management and detailed planning based on operational plans activation to respond accurately to the situation. A tactical plan corresponds to the decisions made to manage a disaster scenario based on the preparation. These plans aim at defining precisely who intervene, what, where, and how to respond to the disaster situation. Their goal is to respond accurately to the situation needs according to the responsibilities of the stakeholders. They define a protocol of actions and manage resources (e.g., humans, vehicles, equipment) to achieve a task according to the situation conditions (e.g., events, weather, location, available resources).

A tactical plan is defined by [Federal Emergency Management Agency, 2010] as following: "Tactical plans focus on managing personnel, equipment, and resources that play a direct role in incident response. Pre-incident tactical planning, based upon existing operational plans, provides the opportunity to pre-identify personnel, equipment, exercise, and training requirements. These gaps can then be filled through various means (e.g., mutual aid, technical assistance, updates to the policy, procurement, contingency leasing)."

Discussion These three levels of plans are clearly defined by the Federal Emergency Management Agency (FEMA) of the United States in [Federal Emergency Management Agency, 2010], but they are not specific to the United States. Indeed, the authors of [Labba et al., 2017] study the organizational structure of disaster management in the United States, the United Kingdom, and France. They observe these three levels in common to each of these countries. The European project EPISECC also highlights these three levels of management in [Information, 2014]. However, in Europe, there are variations in the definition of operational and tactical levels. For example, in France, definitions of operational and tactical levels are inverted compared to

the previously presented description. In France, they use the term tactical level to describe administrative staff and commander that make decisions, but who are not on the ground. They use the term operational level for the management on the ground. This difference is due to the meaning of operations, which are actions directly performed on the ground in France. In contrast, in the United States, the tactical level is nearest to the ground view because it means how to achieve an operation technically. In Germany, operational and tactical levels work together to manage operations through command and control management thanks to the proximity between the command center and people on the ground (c.f. Appendix A.3).
Despite definition variations and structural organization variations, the three aspects are present with a first standard level that provides organizational structure and guidelines to plan disaster management (corresponding to strategic aspect); and a second level with operational and tactical elements, generally leads by a responsible of jurisdiction.

3.1.2.3 Plan experiment

An efficient preparedness depends on the cycle which aims at (1) planning and organizing disaster response, (2) experimenting the plans through training and exercise, and (3) Assessment of the planning based on the experiments. Therefore, plan assessment is an essential step of the preparedness to identify potential problems and errors of planning. However, the plan assessment requires a plan experiment step. There are two ways for experimenting plans: exercises and computer simulations.

Exercises The goal of exercises is to test and improve the collaboration between the different stakeholders. It aims at simulating the real condition of a disaster according to a scenario. The exercises aim at gathering all stakeholders which intervene in disaster response and have a high cost to organize them, which makes them occasional. The high cost of exercises limits their number and the possibility to test scenarios essential in the assessment process of the response plan.

Computer simulation Computer simulations aim at testing designed plans or different strategies of actions. Compared to exercises, the main advantage of computer simulation is to allow a high number of experiments to assess a model or a strategy. Computer simulation is a low-cost and effective method for testing multiple inputs and assessing different outputs to observe a system [Brown and Robinson, 2005].

3.1.2.4 Plan effectiveness assessment

The evaluation of a plan's effectiveness should allow determining the conditions under which a plan can be applied, and its success rate under these conditions. To this end, it is necessary to define the effectiveness of a plan quantitatively using metrics. However, the effectiveness of a plan depends on its objective's success, which can affect various areas. The "Sphere project" [The Sphere Project, 2011] has identified the following areas: health, shelter, food and nutrition, water and sanitation. It is, therefore, necessary to define metrics to evaluate plans in these different areas. These metrics can then be used to group situations for which a plan has similar effectiveness to extract the common characteristics of these situations. These characteristics allow the identification of the applicability conditions of a plan. Thus, plan assessment requires clustering on simulation data to group simulations for which the plan has similar effectiveness. This is a data clustering problem.

Metric assessment Several metrics have been developed to support plan assessment. In the work in reference [Larsson, 2008], the authors identify three main criteria for assessing plans, including the effectiveness criterion. The authors propose to evaluate the effectiveness of a plan according to the following parameters:

1. Victims found
2. Victims whose condition worsens
3. Identified property domage
4. Property sustained further domage
5. Infrastructure operating

These parameters allow the evaluation of the plans' effectiveness in the field of health and shelter. In the work in reference [Bayram et al., 2012], the authors identify 12 quantitative parameters that can be used to make an assessment based on health, shelter, food and nutrition, water and sanitation. These 12 quantitative parameters can be used as a basis for making an assessment of the plans in terms of the areas they address:

1. Number of Excess Deaths
2. Number of Under-5 Excess Deaths
3. Number of Cases with Acute Communicable Diseases
4. Number of Cases with Traumatic/Chemical/Radiological Injuries
5. Level of Health Care Services
6. Number of Young Children (6-59 months) with Acute Malnutrition
7. Number of Displaced Persons (internally displaced or refugees)
8. Number of Persons with Inadequate Living Space

9. Water Quantity

10. Water Quality

11. Level of Sanitation Facilities

12. Gender-based violence

The metric used to compute the effectiveness of a plan must be adapted to its goal. The plan's goal is linked to one or several of the presented quantitative parameters. Therefore, the effectiveness metric must be computed with the parameters linked to the plan's goal. These parameters are observed variables during a simulation. In addition to computing the effectiveness metric, it is essential to define its associated applicability context. Such a definition requires gathering simulation with similar effectiveness and identifying the criteria that characterize the effectiveness' applicability context. When assessing plans, clustering is carried out on various dimensions, where each dimension represents a criterion. However, criteria impact a plan's effectiveness differently, and some criteria may not impact the plan's effectiveness. Unsupervised clustering by considering criteria that do not impact the plan's effectiveness leads to over-segmentation of groups. Therefore, it is necessary to identify the criteria that do not have a significant impact on the plans.

Clustering approaches by unsupervised learning The review [Saxena et al., 2017] presents most relevant approaches for Unsupervised Data Clustering. This review splits the different approaches in two main families of clustering approaches: Hierarchical clustering and Partitional clustering. Hierarchical clustering can be agglomerative or divisive and use single, complete or average-linkage. The studied approaches of this family are the approaches BIRCH [Zhang et al., 1996], CURE [Guha et al., 2001], ROCK [Guha et al., 2000], and CHAMELEON [Karypis et al., 1999]. Partitional clustering are divided between distance-based, Model-based, and density-based approaches. Distance-based approaches uses Error Square, whereas model-based and density-based approaches use probabilistic. The studied approaches for the Partitional clustering family are the approaches K-means [MacQueen et al., 1967], CLARANS [Ng and Han, 2002], FCM [Dunn, 1973].

Among the different approaches, the hierarchical clustering CURE is the most relevant approach for its low complexity (i.e., $O(n^2 log(n))$ as worst-case time complexity), its high scalability, and its suitability for large data and low sensibility to outliers. Although BIRCH approach has a better time complexity than CURE approach, CURE approach has a better quality than BIRCH approach. Moreover, BIRCH is not suitable for high dimensional data. Therefore, the CURE approach appears as the best compromise between time complexity and clustering quality.

3.1.2.5 Discussion

The preparedness is the essential step of disaster management to guaranty an effective response. It aims at preparing plans, assessing them, preparing resources, and stakeholders by training. The assessment of plans plays a crucial role in guarantee good preparedness and effective response. Indeed, being prepared and trained to apply an ineffective plan is useless; that is why the assessment of plans is essential to know its effectiveness. Therefore, the plan's effectiveness must be measured to support their improvement. Although it exists some metrics proposed in the literature and presented previously, there is no standardized system of measurement necessary for comparison [Guha-Sapir and Below, 2002]. A system to measure the plan's effectiveness must identify objectively, deficiencies in the application of plans. It thus requires to experiment plans to assess their global and their case-specific applicability through low-cost simulation. The assessment requires thus clustering approach to gather plans according to their common specificities that make them effective. Low-cost simulations must experiment plans to allow such an analysis. Therefore, computer simulations are the most suited to provide a vast diversity of experiments with the lowest cost. The next section presents the simulation approaches for disaster management.

3.1.3 Simulation approaches supporting preparedness

Simulation techniques are often used in the disaster community to support them in decision-making. They are mainly used to assess risk through the simulation of disaster or to assess response strategies during the preparation or in real-time during the Response. The review of [Mishra et al., 2019] on disaster management simulation modeling research highlights the use of four main techniques: System dynamics, Monte Carlo simulation, discrete-event simulation, and agent-based simulation. They analyze the repartition of the different techniques according to disaster management topics and disaster management steps. The system dynamics are the most present (42%), followed by Monte Carlo simulation (25%), followed by agent-based simulation (22%), and finally, discrete-event simulations (11%). System dynamics are mainly used during Mitigation for risk assessment/identification (21%) and vulnerability assessment (4%). It is also used during preparedness for prevention and recovery schemes (15%). Monte Carlo simulations are mainly used during Mitigation for risk modeling (16%). Furthermore, they are sometimes used during Response for solving disaster relief (2%). Agent-based simulations are mainly used during Preparedness and Response for simulating rescue and evacuation strategies (13%), disaster management (4%), and modeling healthcare (2%). Finally, discrete-event simulations are mainly used during preparedness and response to model large-scale (8%). Among the different simulation techniques used for disaster management, the agent-based simulation, also called multi-agent simulation, is the most used during the preparedness to assess strategies and plans of disaster response [Mishra et al., 2019]. The granularity of multi-agent simulations allows the representation of disaster management stakeholders, their organizational structure, their interactions, and their actions that come from their preparation and knowledge according to the situation, but also population behavior according to their specificities. This technique allows the simulation of complex systems without modeling it directly but by obtaining it by the emergence of its model components and their interactions. That is why the simulation technique based on a multi-agent system is the most suited simulation technique to address the plan experiments.

Multi-agent simulation approaches for preparedness The survey [Hawe et al., 2012] presents a classification of agent-based simulation for large-scale emergency response according to four different usages. The first category of usage (called U4 in [Hawe et al., 2012]) corresponds to simulations for real-time Response. The three other categories belong to simulations for preparedness. The second category (called U3 in [Hawe et al., 2012]) corresponds to preparedness simulation with agent behavior using new algorithms. The two last categories of preparedness simulations use agent behavior based on reality. The third category (called U2 in [Hawe et al., 2012]) corresponds to preparedness simulation that searches for a specific optimal response. The fourth category (called U1 in [Hawe et al., 2012]) corresponds to preparedness simulation that searches for a generalized optimal response. The usage of ABS during the real-time Response is minor compared to the usage during the preparedness. Among the usage during preparedness, the majority of simulation approaches focuses on determining optimal solution to a problem using an agent behavior based on reality rather than focuses on the elaboration of agent behaviors using new algorithms. The agent behavior based on reality corresponds to the assessment of existing strategies to identify the best one when there are several possibilities or the best parameter for applying an approach. The majority of simulation approaches searching for optimized solutions are designed for a specific problem [Hawe et al., 2012]. The most addressed specific problem is the allocation of resources both in terms of human rescuers [Praiwattana and El Rhalibi, 2016] or robotic rescuers [Blatt et al., 2016], and of resource means [Hawe et al., 2015, Marecki et al., 2005]. For the optimization of allocated resources and the planning of response action, the simulation results are assessed in terms of time and success quantity (e.g., ratio rescued people) according to the purpose of the strategy assessed by the simulation or the use case addressed by the simulation. This type of simulation addresses mainly the rescue strategies. Another set of simulation focuses on evacuation simulations [Christensen and Sasaki, 2008, D'Orazio et al., 2014, Zhou et al., 2012, Mas et al., 2015, Nagarajan et al., 2012] by taking into account some aspects as the behavior of the population, the traffic, communications, and prepared plans. Other approaches that focus on the generalized optimal solution as [Saoud et al., 2006] experiments and compare results obtained by different strategies. The combination of different action strategies corresponds to different plans. Therefore, such approaches of general optimization assess according to specific criteria a plan to define the optimal one according to some situation's characteristics. However, all of these simulation approaches for optimal solutions are specialized for a problem, which is more or less specific. The specificity of their design can go from (1) identifying the optimal parameters of a plan (meaning configuring a plan optimally) for a specific situation corresponding to an optimal solution for a specific problem to (2) identifying the best plan (meaning defining the best combination of action strategies of a plan) for a specific scenario. The specificity of the simulation model, according to a situation, is essential to simulate and assess a plan accordingly. However, the limit of the existing simulation approaches is their design and experiment process. Indeed, their simulation model and their implementation are limited to a use case and a predefined set of variations in a plan. This case-dependent design process limits the adaptation and the extension of these approaches to the diversity of disaster situations and the variety of response plans. For example, the approach of [Saoud et al., 2006], which is "generic per accident and location, " assesses the impact of some variations in the NOVI plan, which aims to rescue a large number of victims. This approach could be extended to allow the study of further variations of this plan, which is based on a specific organizational structure (c.f. appendix A.3). However, the assessment of another rescue plan, based on another organizational structure would require a new design process of simulation to define a new model and new implementation. Plan assessment through simulations requires modeling and implementing several types of agents and several organizations corresponding to groups of agents to represent all stakeholders, their different behaviors, and their different organizational structures. Therefore, the first limit of existing approaches for the evaluation of disaster management is the adaptability of simulation models and experiments to the diversity of disaster management plans. The process of modeling simulation and experiments is subjective. Thus, the first limit implies a risk of including different biases during the modeling of simulations and experiments for different plans. These different biases can compromise their evaluation and comparison. This thesis's objective to face this first limit is the flexibility increase to allow the assessment of the different plans of disaster management through their simulation. Such research objective belongs to the domain of multi-agent simulation engineering, whose primary investigations to achieve it are presented in section 3.2.1.

Approaches of multi-agent simulation engineering presented in section 3.2.1 propose metamodels and ontologies combined with specific system architecture allowing generative programming. They aim to solve the lack of flexibility and reusability of simulation models and their implementation in the same application domain. The study of these approaches shows the benefits of using ontologies to represent a simulation metamodel and a system architecture based on an extendable and adaptable agent's behavior to provide flexibility and reusability in simulation modeling and experiments. The combination of modeling with a set of implemented multi-agent system components as made in [Poveda et al., 2015, Boufedji et al., 2018] allows the extension and reusing components of the implementation model to simulate a variety of contexts.

The implemented components of multi-agent system are dependant of the used simulation platform (e.g. AGLOBE, Repast, Gama, etc c.f. section 3.2.3). Such engineering approaches provide flexibility and reusability of multi-agent components for model variations. This study shows that the adaptation of simulation modeling and experiments requires a flexible implementation that can be based on the combination of a set of multi-agent system components. However, existing approaches have some lacks to be reused for disaster management plans assessment through simulation experiments (further detailed in section 3.2.1). Therefore, the second limit linked to the first one is the adaptation and reusability of implemented multi-agent simulation components to allow a diversity of implementation of a simulation model for plan assessment. The objective of this thesis to face this second limit is to provide and use a set of implemented components that can be extended and reused to allow processing the diversity of plans. Such an objective requires to review the components of existing multi-agent simulations for disaster management, presented in section 3.2.2 to identify generic and specific components of the different approaches. It also requires to identify the suitable simulation platform allowing such implementation and extensibility. The study of the suitable platform is presented in section 3.2.3.

Moreover, these approaches highlight the third limit of existing approaches for plan assessment, which is a limit of expressivity. Indeed, multi-agent approaches for modeling disaster management simulation gathers knowledge about plans and disaster scenarios from experts ad hoc to model them in the multi-agent paradigm. Thus, existing approaches are addressed for the computer expert community to model disaster management simulation in multi-agent paradigm in cooperation with experts of the domain, but are not addressed to a disaster management community that search at assessing different plans to prepare them to face disaster. Therefore, the objective of this thesis to face the third limit is to increase the expressivity and the interoperability of plan representation for disaster management community to gather their knowledge and allow them to assess plans according to their definition of plans and knowledge. Such objective relates to the domain of knowledge engineering, whose related work is presented in Section 3.3.

Table 3.1 summarizes the identified limits of approaches for plan assessment and objectives to face them linked to the next sections of this chapter.

Limits	Objectives	Study
(1) Adaptability of simulation models and experiments to the diversity of plans representation	Increase Flexibility to allow the assessment of the diverse plans of disaster management through their simulation	Domain of multi-agent simulation engineering section 3.2.1
(2) Adaptability and Reusability of MAS components for disaster management	Increase Extensibility by allowing to process the diversity of plans	Components of existing multi-agent simulations in section 3.2.2 and simulation platform allowing such extensibility in section 3.2.3
(3) Plan and scenario are expressed in MAS paradigm	Increase Expressivity and Interoperability to gather DM knowledge and allow DM community to assess plans according to their plan definition	Domain of knowledge engineering in section 3.3

Table 3.1: Limits of existing multi-agent simulations for plan assessment

3.2 Multi-agent simulation

The previous section has presented the benefits of simulation to experiment plans and assess them. It has shown that the multi-agent-based technique is the most suited for plan experiments through simulation. It has finally highlighted the limits of existing work in the adaptability and reusability of both simulation models and components. Therefore, this section reviews in subsection 3.2.1, the existing work to adapt the simulation model and facilitate the simulation development to the diversity of disaster management plans. This review identifies some common points to facilitate the development of simulation from a diversity of models. However, existing approaches are not flexible enough to be used for the plan assessment. The second subsection 3.2.2 presents, thus, a review of components used in the multi-agent simulation for preparedness to identify the most suited components of a simulation model for preparedness. Finally, it reviews simulation platforms in subsection 3.2.3 to identify the most adapted one to fit the requirements of simulation components and flexibility.

3.2.1 Multi-agent simulation engineering

As part of disaster management preparedness, computer simulations are often used to optimize various response plans and determine the tasks, actions, or resources best suited to a situation. Among existing approaches, those based on multi-agent systems are the most widely used in this context to simulate the behaviors and interactions of stakeholders [Mishra et al., 2019]. It should be noted that disaster management stakeholders and their behavior varies from one locality to another. They also vary within the same locality depending on the disaster situation and needs. Defining a simulation model that can be adapted to different organizations and plans implies variability in entities, their behaviors and interactions. However, multi-agent simulation approaches are usually created with a specific objective in mind, which limits their reuse and explains the large number of models in this field. As pointed out by the authors of [Poveda et al., 2015], the lack of interest in sharing or connecting the work done condemns researchers in this field to reinvent the wheel by creating new simulation models and new implementations of these models in order to run the simulations. However, there is some work dealing with the adaptation of multi-agent simulations for disaster management.

Among these approaches, the approach presented by the authors of [Poveda et al., 2015] presents a general model for the design of emergency management services in indoor environments. This model is composed of three layers: a semantic layer, a simulation layer, and a layer containing the components of the emergency service. The latter layer is based on the simulation layer, which has to be adapted to the context. To solve the problem of reusing and adapting various contexts, the authors of [Poveda et al., 2015] used a semantic layer composed of the Einsim ontology that links external data and guides agent-based simulation. The simulation process is realized through a simulation control system that includes semantic representations in a model repository of emergency service components and an adaptation model that maps the semantic emergency metadata to the agent code. Representing the simulation process of emergency situations through an ontology (c.f. appendix A.1.1.1) allows simulations to be modeled independently of their programmed model. This advantage provides flexibility in extending and reusing the programmed model to simulate a variety of contexts. However, being specialized for social simulation in an indoor environment, this ontology is not general enough to simulate disaster management plans that are not limited to an indoor environment. Compared to emergency planning in an indoor environment, the simulation of disaster management plans requires a more complex representation with the organizational structure of actors, resource management, and a wide variety of plans.

Another interesting modeling approach for disaster emergency preparedness is presented in [Kruchten et al., 2007]. The authors present a conceptual model (formalized by an ontology) of disasters affecting critical infrastructure (energy, transport, communication, etc.). A conceptual model is an abstraction of the essential characteristics of the studied system.

This conceptual model describes four main components and their interactions: disaster events, infrastructures, agents involved in disaster management, and the impacts of the events on the population. Catastrophe events affect the well-being of the population and the condition of infrastructure elements. The latter are the object of observation by the agents and are the target of their actions. This conceptual model has the advantage of representing at a high level of abstraction the issues and main components of disaster management plans using concepts of the agent paradigm. However, this model was created to provide a common language for communicating, analyzing and simulating the interdependencies of critical infrastructures. The simulation of these interdependencies is intended to detect potential problems in the plans. This objective explains the lack of description of the simulation model in the sense that it does not allow the definition of the simulation model's variables and objectives. Indeed, this model limits the use of simulation to the study of potential problems in plans.

Among the non-specific approaches to disaster management, the work of [Boufedji et al., 2018] presents the adoption of variability models to describe the generic and specific aspects of a multi-agent system model. The generic aspects of the model are represented according to the concepts of the agent paradigm such as agent, environment, interaction and organization, while the specific aspects are represented according to the specific application domain. These models are respectively related to the implementation of reusable, generic and specific multi-agent system components. Although this work does not address specific aspects of simulation modeling (e.g. model variables, experimentation), it provides a method for reusing multi-agent system components for various applications represented by various models.

Another interesting approach not specific to the disaster management field but specific to the field of multi-agent simulation is presented in [Christley et al., 2004]. This work presents an ontology for automating agent-based modeling and simulation tasks. This ontology describes the different models involved in simulation design (e.g., conceptual, experimental, programmed models), the basic concepts of the agent paradigm (e.g., agent, environment), as well as concepts linking with simulation experiments (e.g., simulation data), and the programming of the simulation required for its execution (e.g., software programming). This ontology is suitable for any multi-agent simulation modeling and design, whatever the field of application. It offers an ideal set of concepts for the representation of various simulation models, experiments and their development as executable programs.

The study of multi-agent simulation approaches for model adaptation has shown the advantages of ontologies for simulation modeling. Indeed, modeling through an ontology promotes the reusability of the simulation model as well as its interoperability. In most of the approaches presented, model elements are linked to simulation platform components to produce the simulations corresponding to the model. Among the ontologies studied, the ontology of [Christley et al., 2004] offers the highest and most appropriate level of multi-

Approaches	Model	Agent	Environment	Action	Total
[Christley et al., 2004]	X	X	X	X	4/4
[Kruchten et al., 2007]		X	X	X	3/4
[Poveda et al., 2015]	X	X	X	X	4/4
[Boufedji et al., 2018]		X	X	X	3/4

Table 3.2: Overview of existing ontologies in relation to the main multi-agent simulation concepts

agent simulation model abstraction for our approach. Although these approaches facilitate the development of simulations through an ontology that accommodates a diversity of simulation models, they do not address the problem of adapting the simulation model to disaster management knowledge. However, the work of [Kruchten et al., 2007] provides some clues on the relationships between disaster management knowledge and simulation modeling. Indeed, infrastructures and the population, generally considered as the elements at risk in case of disaster, are the main targets of the actions carried out by the agents. The next section presents the components of a multi-agent simulation model for disaster management.

3.2.2 Multi-agent simulation model components

A multi-agent simulation is based on a multi-agent system. An environment, objects located in the environment, agents that are active objects, and interactions between these different components characterize such a system [Ferber, 1997]. Objects being passive components contained in the environment, their modeling is generally a part of the environment modeling. Therefore, there are two main components to model in the multi-agent system: the environment (including objects inside) and agents. Moreover, simulation aims at experimenting, that is why, its modeling is composed of parameters to configure it and observed variables to assess the experiments. This section presents, thus, an overview of (1) Agent modeling, (2) Observed variables and assessed criteria, and finally, (3) Environment modeling.

3.2.2.1 Agent modeling for disaster management simulation

The concept of Agent represents active entities that can interact between them, with objects and with their environment. In the context of disaster management, the Agent concept represents mainly three categories: the affected population, disaster manager who make-decision, and responders who respond to a disaster on the ground. Among the responders, both responders (e.g., doctors, nurse, fireman) and responder engines (e.g., ambulance) can be represented by an agent as in [Saoud et al., 2006].

According to Wooldridge [Wooldridge, 2002], it exists two main types of Agents: reactive and cognitive.
The reactive agents have been presented by [Brooks, 1991, Brooks, 1991] as an alternative to artificial intelligence. Inspired by the biological systems, he has defined an intelligent behavior for an agent without explicit representation of the world, and without explicit abstract reasoning. An intelligent system has been defined as an emergent property of a complex system. It means that the Agent's behavior results from its interaction with the environment. The reactive Agent is generally used to represent the victims during a simulation of disaster. The stochastic behavior is the most used approach to represent the affected populations as in Plan-C [Narzisi et al., 2007], or as in Simgenis [Saoud et al., 2006] which uses a Markov chain. The cellular automaton is also a technique for a reactive agent, but it is more used to represent the behavior of the affected population in the context of evacuation [Arai et al., 2011, Guo and Huang, 2008]. The reactive Agent is a simple agent that allows the creation of complex systems based on simple entities' interaction. This is advantageous for large-scale simulations, but the behavior simplicity without world representation and reasoning is also a disadvantage to represent more complex behavior as human decision-making.

The cognitive Agent is the most classical agent type, which has a goal-directed behavior. The approaches used to create a cognitive agent are close to artificial intelligence approaches. The most basic cognitive Agent is the deduction reasoning agent. This Agent has a knowledge base corresponding to its view and beliefs about the environment, a function to see the environment which brings new information and updates the knowledge base [Wooldridge, 2002]. Its decision-process is based on a set of deduction rules to infer from the knowledge base, the actions to do. This action selection is reduced to a proof problem. This logic-based approach has the advantage of having a precise, logical semantics, which promotes its long-lived. However, the method of proof problems can have a high complexity according to the type of the problem. This high complexity implies an important time-consuming, which is a problem in a simulation with time-constrained. A long time of computation for the decision-process is also a rational problem if the environment has significantly changed between the time of information-making and the action execution.
Another cognitive agent is the procedural reasoning agent. This Agent has a set of plans with a goal corresponding to a postcondition and a context corresponding to a precondition of a plan. These plans correspond to a sequence of actions. According to its environment representation and its goal, the Agent selects a set of plans, whose preconditions are satisfied and allows the accomplishment of its purpose. It selects then one of these plans to execute it. An example of this process is presented by [Poveda et al., 2015], where the main goal is the evacuation in indoor environments: a set of "egress" strategies has been implemented, and their selection depends on the situation context. A famous agent model belonging to this approach of procedural reasoning is the Belief-Desire-Intention (BDI) model, developed by [Bratman, 1987]. According to its beliefs (its beliefs and view of the

environment) and its desires (what it would like to accomplish), the BDI agent chooses a top-level goal to pursue, which determines the agent intentions corresponding to an action plan which it has decided to do. An extended version of this model has been implemented in D-AESOP [Buford et al., 2006] for considering the situation awareness in the context of disaster situation management. The principal limitation of this approach resides in this finite set of plans, which does not allow flexibility in the combination of actions. Some multi-agent systems require agents solve problems according to their capacities of action. For this requirement, a practical reasoning agent is suitable rather than a procedural reasoning agent. Practical reasoning is composed of two main steps: (i) determining a goal to achieve according to the environment's state, (ii) determining how to accomplish the goal in detail. In a simple manner (in the sense of a goal is a simple action and not a plan). The ambulance agent in the AROUND project [Chu et al., 2009] uses a decision tree to decide the action to perform, then it uses a utility function to determine the details of this action. For example, if the selected action is "go to the hospital", the utility function will decide which hospital the Agent must go to. However, this approach is mainly used for more complex behavior design. The general operation of this approach corresponds to (i) observe the environment and update its beliefs, (ii) determine the available options and filter them to choose a goal to achieve, (iii) use means-end reasoning corresponding to artificial intelligence approaches of planning [Wooldridge, 2002] to determine how to achieve the goal, (iv) and finally execute it.
In the paper of [Praiwattana and El Rhalibi, 2016], the information about the environment (agent beliefs) and the available actions are stored in a knowledge base. The process of plan design takes this knowledge base and the agent goal as inputs of a planner using forward chaining state-space search with heuristic function. [Wickler et al., 2006] presents an artificial intelligence planning approach to coordinate activities in an Emergency scenario. This approach uses an ontology called INCA to build a constraint model. A hierarchical task network planner uses this model to determine the courses of action resolving a problem. These authors explain the BDI model, and more precisely, the process of intentions can be extended by using their planning approach to give more flexibility in the action combination. The main difficulty resides in the deliberative step (ii), which manages the Agent goal. The two extremes situations for this step are (i) to do not modify the target until it is achieved that can create a blocked situation if the goal cannot be achieved, and (ii) change this goal anytime, leading to a no achievement of purposes due to their abandonment. It is necessary to find a good trade-off between these two extreme situations for the strategy of deliberation.

In disaster simulations, stakeholders can be numerous. More large is the simulation, more there are agents, more simple agents must be. That is why, in evacuation simulation, the most used agents are reactive agents. On the contrary, in strategy planning simulation, agents that make a decision of actions and design the planning are practical agents to allow more complex cognitive behavior. In the context of simulating global plans application, the number of agents can become very large. That is why the simulation model must have a maximum of reactive agents and a minimum of cognitive agents to limit the complexity of the simulation. The next paragraph presents models to represent responder agents to go deeper in the agent modeling.

According to the structural comparison of disaster management among different countries, the authors of [Labba et al., 2017] have highlighted a hierarchical repartition of responders in three levels: strategic, tactical, and operational level. This is a vertical hierarchy (Top-down), with at the top the strategic level, and at the bottom the operational level. These authors have thus proposed an agent-based meta-model for response organization structures using three different types of agents: one for each level of the hierarchy. Despite different denominations, the study of existing multi-agent systems for disaster response highlights a recurrence of multi-agent models based on three different agents [Hooshangi and Alesheikh, 2017, Praiwattana and El Rhalibi, 2016, Siebra and Tate, 2003]. These three types of agents aim at organizing or planning the response at different levels of granularity. Table 3.3 presents the denomination of each type of agent for each level of granularity.

References/Structure	Top	Middle	Bottom
[Labba et al., 2017]	Strategic level	Tactical level	Operational level
[Hooshangi and Alesheikh, 2017]	Central agent	Coordinator	Rescue
[Praiwattana and El Rhalibi, 2016]	Decision-making agent (DMA)	Control agent (CA)	Field agent (FA)
[Siebra and Tate, 2003]	Strategic agent	Operational agent	Tactical agent

Table 3.3: Platforms comparison according to application domains related to the disaster management domain

This structure of three different agents being representative of the reality and having proved its usability is suitable to represent the different levels of responders. The techniques to represent them are generally specific to a task and cannot be reused for other purposes. The agent model presented by [Praiwattana and El Rhalibi, 2016] is composed of a knowledge base representing the possible actions that an agent can do, and their condition of application. The decision about the sequence of actions to do is made thanks to a planner. In the case of the preparedness, plans are sequences of actions, that is why a planner is not required.

The models for responder agents shows three types of responders:

- Central agent, representing the highest level of coordination between the stakeholder managers,

- Manager agent, generally representing the coordinator of people on the ground, and
- Actor agent, representing responders on the ground.

These three types of agents are mainly practical agents in strategy planning simulation to provide plans at different levels of detail. However, in the context of plan assessment, complex cognitive behavior is not required since plans are existing. Among the different types of responders, the central and manager agents make decisions about plans to apply according to the situation. Therefore, in the context of this thesis, these two agent types require a cognitive behavior to choose the plan among a set, which is adapted to the situation. That is why procedural agents are the most adapted to model central and manager agents in this context. Concerning actor agents (responders on the ground), they learn and train to apply procedures and adapt them to the situation. These agents must react according to orders coming from managers and environment situations. Therefore, reactive agents are adapted to model-actor agents and allow larger-scale simulation than a representation through cognitive agents. Moreover, reactive agents are also the most suitable to model the population, the victims, as the majority of the related works.

In a simulation model, other essential components are the observed variables and assessment criteria, which express the simulation results and allow their assessment. The next subsection review these components in works related to disaster management simulations.

3.2.2.2 Assessment criteria and observed variables in disaster management simulation

The review of observed variables and criteria for disaster management simulation has been applied on ten approaches having different application scenarii: emergency and crisis response [Hawe et al., 2015, Praiwattana and El Rhalibi, 2016, Walter et al., 2016], rescue strategy [Saoud et al., 2006, Takahashi, 2003, Siddhartha et al., 2009, Blatt et al., 2016, Marecki et al., 2005], and evacuation [Christensen and Sasaki, 2008, Balasubramanian et al., 2006].
These approaches have also different goals: resource allocation [Hawe et al., 2015], assessing or comparing strategies [Saoud et al., 2006, Blatt et al., 2016, Balasubramanian et al., 2006, Marecki et al., 2005] or built-environment [Christensen and Sasaki, 2008], planning response [Praiwattana and El Rhalibi, 2016, Walter et al., 2016], optimizing response [Takahashi, 2003, Siddhartha et al., 2009].

References	Protection of elements at risk					Activity performances		Resources quantity	
		Civil Protection		Goods Protection					
	IP	NbC	NbD	IB	IA	TP	SQ	HM	RM
[Saoud et al., 2006]		X	X			X	X	X	
[Takahashi, 2003]		X	X	X			X		
[Siddhartha et al., 2009]		X	X	X		X	X		
[Hawe et al., 2015]		X				X			X
[Praiwattana and El Rhalibi, 2016]						X		X	
[Walter et al., 2016]	X				X				
[Blatt et al., 2016]	X					X			X
[Marecki et al., 2005]							X		X
[Balasubramanian et al., 2006]		X				X			
[Christensen and Sasaki, 2008]						X			

Table 3.4: Observed variables in disaster management simulation (IP: Impacted population, NbC: Number of Casualty, NbD: Number of Dead, IB: Impacted building, IA: Impacted area, TP: Time performance, SQ: Success quantity, HM: Human means, RM: Resource means)

Table 3.4 provides an overview of the variables observed by the previously stated approach. 70% of these approaches observe the impact on the population (impacted population, number of casualties, or dead), and 30% of them observe both the impact on the population and the impact on goods (building or area). Population and goods are generally elements at risk, which are served by a plan or a service in disaster management. Therefore, elements at risk that are served by the assessed plans can be considered as observed variables during disaster management simulation.
The majority of these approaches (70%) assess the time performance of activities, 40% of approaches assess the success of activities, and 20% of them assess both time and success of activities. This observation shows that the time performance of an activity is an essential criterion of efficiency in disaster management.
Half of these approaches also observed the quantity of human or resource means. In a resource allocation problem as in [Hawe et al., 2015], resource quantity is an observed variable. In the assessment of rescue or response strategies, the activities performances are often studied according to human means quantity [Saoud et al., 2006, Praiwattana and El Rhalibi, 2016] or the quantity of other active resources such as robots in [Blatt et al., 2016] or fire engines in [Marecki et al., 2005]. This observation highlights that the quantity of resources has an impact on activity performance. Therefore, the quantity of resources is an essential observed variable to correlate with activity performance.

Observed variables are essential components of a simulation model to assess the results of simulation experiments. The study of observed variables in different simulations for disaster management allows the identification of their type and their correspondence into a disaster management model. This study summarized in table 3.4, highlights three links between observed variables and disaster management model:

1. a link to elements at risk addressed by the disaster management plan,

2. a relation to actions and tasks performed to achieve a disaster management plan, and

3. a link to resource quantity to achieve a disaster management plan.

This information is essential to design the simulation model from the disaster management model. It is used to elaborate on the transformation of the disaster management model into the simulation model in the proposed solution of this thesis. The elements at risk addressed by plans are observed variables of the simulation, whose final quantity must be minimized. The duration and the success of each action of a plan must be observed and recorded during the simulation to provide performance metrics. Finally, resource quantity is generally a source of decision-making during disaster management. Therefore, they are input parameters of the simulation, for which the simulation aims at optimizing them according to the minimization of elements at risk and the maximization of acting performances.

3.2.2.3 Environment modeling for disaster management simulation

An environment can be represented as continuous through vector GIS files (c.f. appendix A.2), as discontinuous through a grid or as a network through a graph. Grid representation is the most straightforward representation of an environment [Hawe et al., 2015], where objects and agents are located through a coordinate of the grid. Moreover, it facilitates the disaster situation representation, the agents' interactions with their environment, and environment effects on agents and objects by categorizing the grid cells. For example, in the simulation model of [Saoud et al., 2006], there are three types of cells: obstacle, danger, and normal cells. These different categories impact the evolution of victims' health state and authorized agent actions according to their location cell and the surrounding cells. However, it has the disadvantage of restricting agent movement [Hawe et al., 2015]. Indeed, a continuous environment represented through vector GIS files is much more precise. It allows using real maps, with the real geometries of environment components (e.g., polygon for buildings, points for agents, lines for roads). It has the advantage of building a graph for a road network, allowing realistic movement of agents. In disaster management simulation, realism and application to real use cases are essential. Therefore, creating an environment based on vector GIS files is more suited to be more realistic for simulating real accidents on real maps [Saoud et al., 2006]. That is why they are often used to create an environment for evacuation simulation as a base for creating the graph of the road network.

Therefore, the simulation platform used for disaster management must allow using vector GIS files to create a realistic environment. The next section presents a review of the multi-agent simulation platform.

3.2.3 Multi-agent simulation platforms

As shown by the literature review (Section 3.1.3), agent-based simulation is the most suitable simulation model for representing the different stakeholders acting and interacting to respond to disaster management according to their plans. Therefore, the chosen platform must be a multi-agent platform.
The simulation's goal in this system is to make a scientific simulation based on experiments to conclude on disaster management plans' effectiveness. To provide a scientific level of such simulation, the realism of the simulation plays a crucial role. Without appropriate realism, the drawn conclusions cannot be used to support decision-making. Therefore, the real-world aspect of geospatial data interpretation into the simulation is a primary requirement for disaster management simulation.
The platform must have the capabilities to simulate the diversity of disaster management applications such as the application of plans for city evacuation in the context of a flood. Such type of application requires two capabilities: large scale simulation and taking into account natural resources and environment. The large-scale ability aims at allowing the

simulation of a vast number of agents, which is necessary for a scenario of a city evacuation containing more than one million inhabitants. Natural resources and environment have mainly a role in the case of natural disaster management as floods or bushfires.

In conclusion, the chosen platform must be a multi-agent platform, ideally designed for scientific multi-agent simulation. The platform must allow the use of geospatial data and large-scale simulations. It must allow scheduling and planning applications through the implementation of Belief-desire-intention agents. Finally, it must allow natural resources and the environment to be taken into account.

The choice of the simulation platform depends on the requirements previously stated for multi-agent simulation of disaster management and the platform's reliability.
The authors of [Kravari and Bassiliades, 2015] have made an up-to-date comparative review of agent simulation platforms to support readers in their choice of a platform. This review presents and compares 24 platforms according to a set of 23 universal criteria gathered into five categories: Platform properties, Usability, Operating ability, Pragmatics, Security management. They also collect platforms according to domains of application.

As presented previously, the prime pre-requisite of a platform for disaster management simulation is to allow real-world simulation from geospatial data. Therefore, among the 24 presented platforms, the first selection of seven platforms has been made according to their specialization or common use in the real-world and GIS aspects. These platforms are AGLOBE [Šišlák et al., 2006], Cougaar [Helsinger and Wright, 2005], Repast [Kravari and Bassiliades, 2015], CybelePro, SeSAm [Klügl, 2009], AnyLogic [Borshchev, 2013], and GAMA [Grignard et al., 2013]. The more platform is designed for an application, and more, this platform is efficient for this application. Thus, these seven platforms have been compared according to the other requirements and application domains of disaster management. Table 3.5 shows the comparison of these seven platforms according to six application domains related to disaster management (made according to Table 9 in [Kravari and Bassiliades, 2015]).

	AGLOBE	Cougaar	Repast	CybelePro	SeSAm	AnyLogic	GAMA
Real-world and GIS	X	X	X	X	X	X	X
Large scale simulations		X		X			X
Scientific simulations			X	X	X	X	X
General purpose agent-based simulations					X	X	X
Scheduling and planning				X	X	X	X
Natural resources and environment					X	X	X
Total:	1/6	2/6	2/6	4/6	5/6	5/6	6/6

Table 3.5: Platforms comparison according to application domains related to the disaster management domain

This comparison shows the GAMA platform as the most suitable platform for application domains related to disaster management. The platforms CybelePro, SeSAm, and AnyLogic also provide capabilities for the majority of application domains related to disaster management.
The main advantage of the GAMA platform compared to SeSAm and AnyLogic platforms are to allow large-scale simulations. This capability plays a crucial role to simulate evacuation plans of a big city.
The main advantage of GAMA compared to CybelePro is to be specialized in agent-based simulations and to have applications in natural resources and the environment.

These four platforms (SeSAm, CybelePro, AnyLogic, and GAMA) are further analyzed using the universal criteria of [Kravari and Bassiliades, 2015]. Concerning the pragmatics category, they have all at least excellent user support and are still in active development. Concerning the categories of operating ability, platform properties, and security management, the platforms can be gathered into two categories: one category gathering AnyLogic and CybelePro, and another one gathering SeSAm and GAMA. The first group has the advantage of having a high operating ability globally and at least a good platform security with fairness. In contrast, the second group has an excellent running ability and average security without fairness. However, although the first group has better security management and operating ability, it has a significant limit in the platform properties category. The first group is a commercial, whereas the second group is free. Moreover, the level of operating ability is good enough for prototyping. Finally, concerning the usability category, the first group has an average simplicity, whereas the second one is simple. In terms of usability, GAMA also has the advantage of being compatible with ACL and FIPA standards, which is not the case for the three other platforms.

Although the platforms as AnyLogic or CybelePro have higher security and operating ability, GAMA is still the most suitable platform for the context of plan experimentation for disaster management preparedness due to its license and the capabilities that it provides. GAMA allows "complete modeling and simulation" of large-scale simulations. It combines explicit multi-agent simulations with GIS data management, multi-level modeling, and the capability to implement BDI and reactive agents. Thus it is powerful for prototyping through its agent-oriented language GAML.

Besides, GAMA allows an efficient extension of the agents' behavior using "skills" defined as additional plugins. GAMA provides a library of "skills" that can be assigned to an agent for diverse domains such as moving, communication, graphics. Therefore, these skills need to be extended to the disaster management domain to allow plan simulation.

3.2.4 Discussion

This section has highlighted approaches to provide flexibility in the simulation development by proposing (1) a model or a meta-model most often specified through an ontology and (2) an architecture based on a set of implemented components used by a simulation platform.

It has been necessary to study primary components of such simulation to design such architecture for plan assessment through disaster management simulation. This study has been made in section 3.2.2 by presenting (1) the most suited agent types for the modeling of the different categories of agent belonging to this application domain, (2) links between observed simulation variables and the application domain, and (3) the most suited environment modeling for the application domain. Based on this study and some other identified requirements, section 3.2.3 has presented the review of multi-agent simulation platforms to conclude that the GAMA platform is the most suited for this application domain.

Concerning the simulation modeling, meta-models specified through an ontology have shown great flexibility to represent a diversity of simulation models and facilitate their development. Indeed, the authors of [Durak and Oren, 2016], claim that simulation engineering through the use of ontologies is the evolution of the simulation domain. The use of ontologies for simulation modeling provides three main advantages:

1. **Facilitating the simulation development.** As presented in section 3.2.1, approaches as [Poveda et al., 2015, Kruchten et al., 2007, Christley et al., 2004] use ontologies to provide flexibility in the simulation modeling and allow the reusability of implemented simulation components. Ontologies are also used to facilitate simulation development for other application cases as distributed simulation applications [Benjamin et al., 2006] or other simulation techniques as discrete-event simulation with the ontology DEMO [Miller et al., 2004]. Another interesting approach for simulation development based on ontology has been proposed by authors of [McGinnis et al., 2011]. These authors use an ontology to create a specific conceptual model for a problem in a domain. The user can create the ontology implementation referred to as a domain-specific language (DSL) for a class of simulation applications through the use of OMG SysML. Once the conceptual model created through DSL ontology, a model transformation is used to automate the translation to a computational simulation model.

2. **Promoting and facilitating the sharing and reusing of simulation data.** Broadly speaking, ontologies in the domain of the Semantic Web aim at describing resources to facilitate the research of relevant information. The simulations are composed of several models, are based on data, and produce data. The sharing of these elements can facilitate their reusability. The authors of [Lacy and Gerber, 2004] explain that XML has been long used to interchange simulation data thanks to its advantage of

solving interchange problems. However, they promote an upgrade from XML to OWL, which overcomes the weakness of XML-only approaches, provides an explicit semantic, and allows the inference (c.f. inference in appendix A.1.2.2). The creation of an ontology is also one approach to facilitate sharing of knowledge in the domain of modeling and simulation as illustrated by the ontology of discrete-event simulation and modeling called DEMO [Miller et al., 2004] and the ontology to describe the simulation systems engineering process [Durak and Oren, 2016] according to the IEEE standard for Distributed Simulation Engineering and Execution Process (DSEEP) [IEEE, 2011].

3. **Facilitating the integrated simulation with a real system into a unique system.** The common vocabulary provided by an ontology facilitates the exchange of information between two systems: a real system can base its operation on information representing through an ontology, which is enriched by the integrated simulation. The authors of [Poveda et al., 2015] have designed an ontology-based simulation, which aims at being reused by other intelligent systems. The authors of [De La Asunción et al., 2005a] present the SIADEX planning framework composed of an integrated simulation and a real system. This SIADEX framework aims at planning firefighting according to the situation representation. It uses an ontology called BACAREX to represent the information required for the planning process. The integrated simulation aims at enriching the BACAREX ontology by representing the future state of the fire situation that constitutes an input parameter of the planning system. The ontologies with the techniques from the Semantic Web also allow logical reasoning operation called inference. Such a process based on ontology supports the operation of a real system. In [Han et al., 2010], the authors use the integrated system to obtain a representation of the fire situation evolution through an ontology and use the inference process to determine hazards from the situation representation.

From these advantages, the use of ontology for simulation modeling brings benefits both for the sharing in the simulation community and for the flexibility of simulation modeling and development required for plan assessment. Among existing ontologies studied in section 3.2.1, the ontology proposed by [Christley et al., 2004] is the most suited for multi-agent simulation of disaster management. However, it provides only high-level concepts that must be specified for the application domain. Moreover, although approaches presented in section 3.2.1 provides flexibility for the diversity of simulation modeling, they do not propose a method to design the conceptual simulation model according to disaster management knowledge.

Although no studied approaches transform disaster management model into a multi-agent simulation model, some simulation engineering approaches exist in other application domains that use ontology in the process of model transformation from a domain ontology to an ontological simulation model. The method presented by the authors of

[Silver et al., 2007] uses the knowledge encoded in ontologies to facilitate simulation modeling. The suggested technique establishes relationships between domain ontologies and a modeling ontology. It then uses the relationships to instantiate a simulation model as ontology instances. An application of this suggested technique has been presented by authors of [Silver et al., 2009] for simulations based on discrete-event modeling technique. These authors use alignment and mapping information between the domain ontologies and the Discrete-event Modeling Ontology (DeMO) to create DeMO instances. These DeMO instances can then be used by a code generator to produce an executable simulation model. Such an approach based on alignment and mapping information between a domain ontology and a simulation modeling ontology brings flexibility to adapt simulation modeling according to a domain ontology. It is necessary to design a domain ontology of disaster management to apply such a method adapted to the application case of disaster management model and multi-agent simulation modeling. Such research belongs to the domain of knowledge engineering presented in the next section.

3.3 Knowledge engineering

In the knowledge management literature, the knowledge is generally defined according to its hierarchy based on information and data, called the Data-Information-Knowledge (DIK) pyramid. Rowley has reviewed the most influential literature to compare the description of the different levels of this pyramid [Rowley, 2007]. Although the definitions of data, information, and knowledge are various, there is an agreement that each level of the pyramid can be obtained from lower levels and that the highest level is the most valuable. As said Ackoff: *"An ounce of information is worth a pound of data. An ounce of knowledge is worth a pound of information," [Ackoff, 1989].*

Data At the base of this pyramid states data. [Ackoff, 1989] presents data as *"symbols that represent the properties of objects and events"*. In [Rowley, 2007], Rowley gathers data definitions that explain *"data are discrete, objective facts or observations"* based on [Chaffey and White, 2010, Awad and Ghaziri, 2004] and *"Data items are an elementary and recorded description of things, events, activities and transactions"* based on [Boddy et al., 2005]. Moreover, this author highlights that the data definitions contain mainly *"what data lacks"* and thus *"lay the foundations for defining information in terms of data"* [Rowley, 2007]: *"Data has no meaning or value because it is without context and interpretation"* [Jessup and Valacich, 2003] and they are *"unorganized and unprocessed"* [Awad and Ghaziri, 2004, Chaffey and White, 2010]. Thus, in the domain of disaster management, data can, for example, be "Köln", "830", "13:30", "23.08.2018". This example shows that data items are raw numbers or raw character chains that are useless due to the lack of meaning and context of these values. These data are processed to become meaningful and useful, but once processed, it is no more data but information.

Information Indeed, information is defined in terms of data. Information is: *"formatted data"* [Jessup and Valacich, 2003], *"data that have been organized so that they have meaning and value to the recipient"* [Rainer and Turban, 2008], *"data that have been shaped into a form that is meaningful and useful to human beings"* [Laudon and Laudon, 2006], *"aggregation of data that makes decision making easier"* [Awad and Ghaziri, 2004]. These definitions show information brings an added value to data based on their process. They illustrate different methods to transform data into information such as organizing, shaping, or aggregating. The authors of [Davenport et al., 1998] have presented five methods to transform data into information by adding values: (i) *"Contextualized: we know for what purpose the data was gathered"*, (ii) *"Categorized: we know the units of analysis or critical components of the data"*, (iii) *"Calculated: the data may have been analyzed mathematically or statistically"*, (iv) *"Corrected: errors have been removed from the data"*, (v) *"Condensed: the data may have been summarized in a more concise form"*. The added values of information compared to data are the meaning, relevance, the purpose, and the usefulness as presented by Jashapara's definition *"Information is data that is endowed with meaning, relevance and purpose"* [Jashapara, 2004]. Information is, therefore, interpreted data. It corresponds to the setting in the context of the raw values contained in the data. Thus, it is necessary to interpret data to create information. In the domain of disaster management, the interpretation of data: "Köln", "830", "13:30", "23.08.2018" means the water level in Cologne (Germany) is 830 cm at 1:30 pm on 23rd August 2018. This is information.

Knowledge Rowley in [Rowley, 2007] shows that knowledge is defined in term of data, information, and added values compared to information:

"Knowledge is the combination of data and information, to which is added expert opinion, skills, and experience, to result in a valuable asset which can be used to aid decision making" [Chaffey and White, 2010],

"Knowledge is data and/or information that has been organized and processed to convey understanding, experience, accumulated learning, and expertise as they apply to a current problem or activity" [Turban et al., 2005],

"Knowledge builds on information that is extracted from data [...] While data is a property of things, knowledge is a property of people that predisposes them to act in a particular way" [Boddy et al., 2005].

These definitions highlight the added values of knowledge, which are the experience, skills/accumulated learning, and the expertise, as well as the goal of the knowledge, which is to make a decision, act and solve problems. Some authors, as Jashapara [Jashapara, 2004], explain the semantic importance of the information to create knowledge. Indeed, the same information can have different implications and refers to different knowledge for different persons according to their experience, skills, and expertise. Let us continue the example of disaster management based on the information on the water level. A water level of 830 cm in Cologne (Germany) implies the building of a protection wall against floods and the application of the procedure to build a protection wall. The implication of information and how it is realized constitute knowledge. However, in this case, different knowledge comes into play according to the expertise of the various stakeholders. For example, this information, which implies the building of protection walls, implies for police to block some roads. In contrast, it implies bringing materials and build walls for the organization managing the protection walls. This example shows the role of different expertise in knowledge from the same information. The authors of [Davenport et al., 1998] have presented four methods to transform information into knowledge: (i)*"Comparison: how does information about this situation compare to other situations, we have known?"* (ii)*"Consequences: what implications does the information have for decisions and actions?"* (iii)*"Connections: how does this bit of knowledge relate to others?"* (iv)*"Conversation: what do other people think about this information?".* These methods show the role of experience, expertise, and information to create knowledge.

Some authors differentiate two types of knowledge, which are implicit and explicit knowledge: *"Tacit knowledge refers to personal knowledge embedded in individual experience and involves intangible factors such as personal belief, perspective, and values [...]. Explicit knowledge refers to tacit knowledge that has been documented [...]" [Laudon and Laudon, 2006],*

"Tacit knowledge is knowledge embedded in the human mind through the experience and jobs [...]. Explicit knowledge is knowledge codified and digitized in books, documents, reports, white papers, spreadsheets, memos, training courses, and the like" [Awad and Ghaziri, 2004].

Explicit knowledge in computer science generally relates to semantic knowledge representation often used for intelligent systems (e.g., for object detection [Ponciano et al., 2019a, Ponciano et al., 2019b, Karmacharya et al., 2015], for simulation [Poveda et al., 2015, Durif, 2014, Miller et al., 2004]) or for information systems supporting decision-making (e.g. for disaster management [Shafiq et al., 2012, Babitski et al., 2011, Han et al., 2010, GeoPii, Integrasys, 2014, Fan and Zlatanova, 2010, Beneito-Montagut et al., 2013]).

Knowledge defines what to do and how to act, decide, or solve a problem. Knowledge engineering consists of representing knowledge explicitly. According to Guarino in [Guarino, 1995], it exists different levels of knowledge representation (further detailed in appendix A.1). Among these different levels, the ontological level is used to represent the knowledge explicitly through its semantic. It aims at creating a knowledge base (c.f. appendix A.1.2) that defines concepts with meaning understandable both by humans and machines. The ontological level is chosen for its right balance between the humans' linguistic level and the logical level of computers. The ontological representation of knowledge is done through an ontology (c.f. appendix A.1.1.1).

According to the ontology design methodology presented in appendix A.1.1.2, the first step to build a knowledge base is to identify the scope and the goal of the knowledge

base. The scope of this thesis is to support the preparedness of disaster management. Its goal is to gather knowledge about disaster management plans and their assessment to improve preparedness. It is then essential to study existing ontologies that address this knowledge domain according to the determined scope and goal. Therefore, the first subsection presents existing ontologies related to disaster management and identifies the one that could be reused to represent disaster management plans. Then, the second subsection presents approaches allowing the knowledge extraction from data and their integration into a knowledge model. Indeed, the disaster management preparedness requires gathering all contextual information on administrative areas to allow a preparation adapted to the context. The context is composed of the geographic situation and the resource situation.
The geographic situation is described through information about infrastructures, whose critical infrastructures (e.g., hospitals, schools), the global population repartition, risks, as well as a vulnerable population and critical infrastructures related to risks.
The resource situation corresponds to information about available material (e.g., vehicles, power generator), human resources (e.g., organizations, number of firefighter or doctor), and potential shelters (e.g., gymnasium, celebration room). Resources are mainly named, located, quantified and have a capacity (primarily for shelters and some materials) or responsibilities (for human resources).
The majority of this information answers to who, what, and where are retrieved from geospatial data. It exists two models of geographic data (vector data and raster data), which are presented in appendix A.2.

From the information presented previously, the disaster preparedness knowledge is mainly composed of the organizational structure of responders, which explains how responders are organized and how they communicate between them; and the response plan, containing tasks, actions, responders roles that explains how to respond to a disaster situation.
This knowledge comes directly from experts and explicit knowledge, which is documented knowledge (as explained previously). The documented knowledge often corresponds to organizational charts and paper plans.
In addition to information and knowledge provided by the disaster preparedness, some supplementary information is required to assess plans through their application. The plan application depends on the scope of plans, which depends on the disaster location, type, and impact level, or disaster strength. Information about a disaster aims at determining the administrative area impacted and risks that have been realized to trigger the adapted plans. These elements have been identified through the comparison of plans for France and Germany, made in appendix A.3, to analyze common points and differences in disaster preparedness.

3.3.1 Knowledge representation for disaster management

This section presents the existing ontologies for disaster management. In 2013, the authors of [Liu et al., 2013] reviewed ontologies used during crisis management. This section presents seven ontologies, also presented by [Liu et al., 2013], and eight ontologies, more recent. Among these ontologies, there are two main goals at the design of these ontologies. They are either designed for information exchange to facilitate collaboration or for planning the response to support decision-making. The next subsections present the existing ontologies according to these two categories of goals. They present their advantages and disadvantages in terms of reusability.

3.3.1.1 Ontologies for information exchange to facilitate collaboration

Emergel Emergel ontology [Casado et al., 2015] has been created for a European project called Disaster, which aims at providing an international common knowledge structure allowing exchange information about performed tasks between the different countries of the European Union during Emergency. It covers domains of disaster (e.g., fire, transport accident), time (e.g.timeslice), Resource (e.g. Vehicle, Equipment, Communication), Role (e.g. BrigadeLeader, Squad leader), Infrastructure (e.g. building, street), Organisation (e.g. FireFighting, Police, AmbulanceService), damage (e.g. 25PercentOutage), Task (e.g. Extinguish, Transport, FloodProtection), and Movement. It is linked to FOAF and WAI vocabulary to describe responders and their roles, as well as NeoGeo vocabulary [2] for spatial objects. Its main advantage is the completeness of the vocabulary to describe actions and tasks made on the ground. Its disadvantage is the lack of high-level concepts for preparedness as plans description.

MOAC The MOAC ontology has been created for the management of a crisis vocabulary. MOAC ontology [3] aims at defining vocabulary for the management of crisis and is expressed through RDF. It is focused on the description of an event (such as natural hazard, emergency) and humanitarian activities through "Who, What, Where". The prime advantage of this ontology is the richness for the description of the event of a disaster, their material aftermath, and the hazards which can result from a disaster because many classes compose it for the infrastructure damage, problem of health, menace, the operation for a vital resource. The principal disadvantage of this ontology is the lack of concepts for the description of different responders in disaster management. Moreover, it focuses on humanitarian activities.

INSPIRE directive INSPIRE directive [Seifert, 2008] has been initiated by the European Union to create an infrastructure for spatial information. The aim is to support Commu-

[2]NeoGeo Vocabulary Specification: `http://geovocab.org/doc/neogeo/`, visited on 2020-09-22

[3]Management of a Crisis (MOAC) Vocabulary Specification: `http://www.observedchange.com/moac/ns/`, visited on 2020-09-22

nity environmental policies, and policies or activities which may have an impact on the environment. A natural or another disaster often affects the environment; that is why it is possible to find a vocabulary for a specific domain of disaster management in the INSPIRE directive. The INSPIRE directive is ordered in 34 themes, whose "Human health and Safety" and "Natural risk zones". Initiatives exist to create an INSPIRE ontology [4] such as the ontology[5] created in the context of heterogeneous geospatial data integration [Homburg et al., 2016]. INSPIRE covers damages (e.g., EventConsequence, environmental damage) and affected population (e.g., injured, evacuated, isolated) from the theme safety; the domains of risk and hazard from the theme of natural risk zones; operations and actors from the theme of utility and governmental services; the infrastructure domain from themes as buildings, production, and industrial facilities, agricultural and aquacultural facilities, and transport network; population repartition from the theme of population distribution and demography; meteorology domain from the theme of meteorological geographical features. The advantage of INSPIRE is that it offers a vocabulary for describing information related to disaster preparedness such as infrastructure, governmental service, risk and population, and disaster description such as damage and meteorology. However, INSPIRE has a lack of vocabulary to describe resources and disaster management activities. INSPIRE is not specific to disaster management, but it is accurate in the environment. That is why it allows gathering much information but not the representation of disaster management.

Dires In [Beneito-Montagut et al., 2013], the platform web "Disaster 2.0" allows actors to deposit or research information, make requests of needs, and get answers for the application of requirements. The information and requests added on this platform are stored in an ontology. This ontology is called Dires and describes seven main domains: damage (an element which has been affected by a disaster), disasters (e.g. technological, natural, conflict), geo-location (e.g., location of the incident command post, heliport, stagging area), operations to respond to a disaster, organizations (e.g., police, fire brigade, Business entity), responder roles (e.g., chief, commander, fireman, policeman, ambulance man), and resources (e.g., food, clothes, vehicle, power generator). Dires is an ontology specific to the disaster response. The advantage of this ontology is the definition of the response domain through Ressources, Operation, Casualty, Actor, Disaster, and Infrastructure. The lack of this ontology appears at the level of risk, hazard, and plan description.

DoRES DoRES ontology [Burel et al., 2017] aims at representing information sources, reports, as well as the events and situations that occur in emergency crises. It covers domains of event, situation, geolocation, document, report, task, role, organization, and actors. This ontology gathers concepts from ontologies SIOC, FOAF, Geoname, WGS84, and Dublin Core.

[4]INSPIRE ontology from European Commission: `https://inspire.ec.europa.eu/glossary/Ontology`, visited on 2020-09-22

[5]INSPIRE ontology from SemGIS: `https://github.com/i3mainz/SemGISOntologies`, visited on 2020-09-22

SIOC[6] (Semantically-Interlinked Online Communities) describes the information of online community sites as their structure and contents to find related information and new connections between content items and other community objects.
FOAF[7] (Friend of a friend) Core describes characteristics of people and social groups that are independent of time and technology. In addition to terms of FOAF Core, there are terms to describe Social Web as internet accounts, addressbooks, and other Web-based activities.
GeoNames[8] ontology allows the addition of geospatial semantic information to the Word Wide Web. It provides URI to represent a large number of geographic names and locations.
WGS84[9] is a vocabulary for representing latitude, longitude and other information about spatially-located things, using WGS84 as a reference datum.
Dublin Core[10] is a vocabulary used to describe information about digital resources (e.g., video, images, web pages) and physical resources (e.g., books, CD). This information is related to content (e.g., title, subject, source, description, etc.), intellectual property (e.g., creator, contributor, editor,etc.), and instantiation (e.g., date, type, format, etc.).
Although the DoRES ontology describes some concepts of disaster management, there is a lack of vocabulary to describe disaster preparedness due to their goal, which is the management of information sources.

EDXL-RESCUER EDXL-RESCUER ontology [Barros et al., 2015b, Barros et al., 2015a] has been developped to coordinate and exchange information between rescuers and is based on EDXL (Emergency Data Exchange Language), developped by the Organization for the Advancement of Structured Information Standards (OASIS[11]) [Jones, 2013]. EDXL provides a set of XML-based messaging standards to improve information sharing during an emergency like a natural disaster, for example. EDXL is based on the National Information Exchange Model (NIEM) which is an XML-based information exchange framework from the United States. It exists different types of message for the diverse needs: EDXL-RM (ressource message), EDXL-DE (Distribution Element), EDXL-SitRep (Situation Reporting), EDXL-TEP (Tracking of Emergency Patients), EDXL-CAP (Common Alerting Protocol). The ontology part corresponding to EDXL-CAP covers two main descriptions: message description (e.g., Message type, Alert, Info) and information about the incident (e.g., category of event, resource, response type). This ontology is limited to the description of an incident.

[6]SIOC Core Ontology Specification: `https://www.w3.org/Submission/sioc-spec/`, visited on 2020-09-22

[7]FOAF Vocabulary Specification (0.99): `http://xmlns.com/foaf/spec/`, visited on 2020-09-22

[8]GeoNames Ontology: `http://www.geonames.org/ontology/documentation.html`, visited on 2020-09-22

[9]WGS84 Vocabulary: `https://www.w3.org/2003/01/geo/`, visited on 2020-09-22

[10]Dublin Core Specification: `https://www.dublincore.org/specifications/dublin-core/`, visited on 2020-09-22

[11]OASIS: `https://www.oasis-open.org/`, visited on 2020-09-22

EPISECC EPISECC ontology [Pan and Space, 2016] has been developed in OWL to improve information sharing through a Common Information Space. It is a Spatio-temporal ontology for modeling a common operational picture for the first responders. It addresses the interoperability during the response. It covers five main domains: disaster (e.g., earthquake, urban flood, flash flood), process (e.g., resource management, physical response as decontamination, search for people), resource (e.g., financial, human, institutional), organization (e.g., Governmental, Non-governmental, Private), common operational picture, whose static and dynamic data (e.g.situational data: weather forecast, affected people; operational data: resource, process). It references to GeoSPARQL, W3C Time and DOLCE-Lite ontologies.

GeoSPARQL[12] is a Geographic Query Language for RDF Data. It is an extension of SPARQL for processing geospatial data, which "*supports representing and querying geospatial data on the semantic web. [It] defines a vocabulary for representing geospatial data in RDF. Also, GeoSPARQL is designed to accommodate systems based on qualitative spatial reasoning and systems based on quantitative spatial computations.*" [Perry and Herring, 2012]. GeoSPARQL has the same use as SPARQL, but the difference is when it recovers a triple in a database, its declaration, serialization has not done by RDF/XML but rather than RDF/GML. GML is a Geo Markup Language, which allows describing geospatial data.

W3C Time [13] provides the vocabulary to describe topological temporal relations, temporal reference system (e.g. clock, calendar) time position and duration.

DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [Gangemi et al., 2002] is a foundational ontology. This ontology has a clear cognitive bias, in the sense that it aims at capturing the ontological categories underlying natural language and human common sense. It is an ontology of particulars[14], divided into two categories: the enduring and perduring entities. Endurance is continuants, which are wholly present at any time at which they exist and can change in time (e.g., physical objects). Perdurants or occurrents are extended in time and only partially present at any time at which they exist (e.g., events and processes).

DOLCE-Lite[15] is a lite version of DOLCE to simplify translations of DOLCE into various logical language.

HXL ontology Humanitarian eXchange Language (HXL) ontology [Keßler and Hendrix, 2015] covers the domains of disaster (e.g., Incident), geography (e.g., Administrative Unit), damage (e.g., affected population), organization, and humanitarian response (e.g., displaced population). HXL is linked to vocabularies from FOAF and GeoSPARQL. It allows mainly for describing the population impacted by an incident (e.g.,

[12]GeoSPARQL (OGC Standard): `http://www.ogc.org/standards/geosparql`, visited on 2020-09-22

[13]W3C Time Ontology specification: `https://www.w3.org/TR/owl-time/`, visited on 2020-09-22

[14]Particulars are entities which have no instances [Gangemi et al., 2002]

[15]DOLCE-Lite Ontology: `http://www.ontologydesignpatterns.org/ont/dul/DLP_397.owl`, visited on 2020-09-22

death, injured, displaced). It is limited to humanitarian activities.

Discussion on ontologies for information exchange The ontologies for information exchange to facilitate collaboration have the main advantage to provide a large vocabulary to describe a situation of crisis and response activities. However, they are generally application ontology, which limits them to: (i) a part of response activities as humanitarian (e.g. MOAC, HXL) activities; (ii) a situation and activities description related to information content (e.g.EDXL-RESCUER, DoRES). The ontologies covering larger the domain (e.g. Emergel, INSPIRE, Dires, and EPISECC) do not allow the description of Preparedness' elements related to stakeholders' organizational structure and the plans to gather a set of tasks. The next section presents ontologies for planning response and aims at searching ontologies able to represent preparedness components.

3.3.1.2 Ontologies for planning response

AktiveSa AktiveSa [Smart et al., 2007a, Smart et al., 2007b] aims at representing humanitarian and disaster relief operations for military agencies. It has been used by the system UICDS [Shafiq et al., 2012] to store information for planning and rule-based reasoning to organize planning elements according to the constraints. Even though AktiveSA was designed for military agencies, it is based on humanitarian and disaster relief operations. In this disaster context, military agencies needs to exchange information with humanitarian agencies about: the security situation in the operations area, the locations of humanitarian staff and facilities, the activities planned by humanitarian actors, mine action activities, significant movements of civilians, activities of relief planned by military agencies, strike locations and explosive munitions used during military campaigns, communication infrastructure as the best location for radio repeaters. The knowledge areas of this model are: Geography, Meteorology, Activity, Humanitarian aid, Military, Equipment, Organizations, Weapons. This model is expressed through OWL. The advantage of AktiveSA is a high level of vocabulary description by representing a vast diversity for each knowledge area. However, it does not provide a transparent model with defined relationships between the different concepts.

e-response E-response ontology aims at describing an emergency and the response to that emergency. This ontology is derived from the AktiveSa ontology and contains some concepts derived from DOLCE ontology (e.g.Endurant, Perdurant). E-Response being based on AktiveSa, it covers similar areas of knowledge. Therefore, it has a similar advantage and limit. It is used for the creation and use of a virtual organization to respond to highly dynamic events, such as emergencies in the project Aktive response [16].

INCA The I-N-C-A (Issues – Nodes – Constraints –Annotations) ontology [Tate, 2003] is a constraint model used as a shared representation of intentions for emergency response in

[16]e.Response: `http://www.aiai.ed.ac.uk/project/ix/e-response/index2.html`, visited on 2020-09-22

rescue simulations [Wickler et al., 2006] or to represent rescue actions with constrained in I-RESCUE simulation [Siebra and Tate, 2003]. INCA provides a shared representation of the agent's intention to coordinate activities in an emergency response scenario. The I-N-C-A ontology is also used as a base for an intelligent command and control system in the FIREGRID system presented in [Rein et al., 2007], to represent tasks with constraints. This ontology is interesting to solve problems planning in the context of emergency response to support decision-making but is too limited to represent the preparedness components.

IsyCri IsyCri ontology [Benaben et al., 2008, Lauras et al., 2015] is a meta-model for crisis management represented through OWL-DL. It has been designed to gather information and knowledge about crisis management into a crisis response coordination system. This ontology aims at characterizing a crisis to coordinate the response process between the heterogeneous partners. It represents concepts through three categories:

- Crisis characterization through concepts as Crisis, Factors, Effect, and Trigger;
- Studied system through concepts as Risk, Danger, Event, and Study System Components (i.e., Good, Civilian society, People, and Natural site);
- Treatment system through concepts as Actor, Procedure, Resource, Service, Task, Collaborative process.

The meta-model Isycri has the advantage of a high-level description to cover the whole set of required concepts. However, a meta-model can only be a base for further development and specification of use cases. Thus, a meta-model provides a base for the description of a diversity of use cases and for generic reasoning on high concepts to apply knowledge on a crisis situation.

Bacarex The SIADEX framework is decision support in forest fire fighting [De La Asunción et al., 2005a]. It has a monitoring algorithm, which (1) tracks in real-time changes according to the execution of the current plan, (2) updates the ontology, and (3) checks if the execution is like the prediction. This check aims to detect a problem as failure execution or unexpected delay to give the solution by re-planning according to the problem's circumstance. The ontology used by this algorithm is called BACAREX. The BACAREX ontology is an ontology of planning objects and activities related to forest fire fighting plan. Objects are called resources and are represented in two main classes: Material and Human. Among material, there are facilities and vehicles. Facilities have a GIS point as a fixed position, whereas cars and humans have a current position due to their dynamic of the move. Information stored for every object is operational (usage of coordinates by reasoning process of planning) and informational (information required for by the technical staff during an episode). The BACAREX ontology contains a fire scenario concept that forecast weather and has a physical deployment in GIS locations specific to fire and a concept of shifts with duration and linked to Human. This ontology is also composed of constraints to check the consistency dynamically and early detect the case of inconsistency.

These constraints are temporal constraints, mainly related to actions and necessary for planning, or restrictions on operating procedures to specify resource use conditions. This ontology's limit for this thesis is its focus on fire fighting domain that limits the vocabulary (for a plan, resource, situation description) to this domain.

EMPATHI The EMPATHI ontology [Gaur et al., 2019] has been created for emergency management and planning about hazard Crisis. Super-concepts of this ontology can be gathered into three categories of description:

- Data and information management through concepts as Modality of data, Report and Surveillance information;
- Situation description through concepts as Place, Event, Impact, Age Group,Hazard type and phase;
- Response Activity through concepts as Involved actors, Service, Status, Facility.

EMPATHI ontology integrates vocabulary from Friend Of A Friend (FOAF) [17], GeoNames [18], Linked Open Descriptions of Events (LODE) [Shaw et al., 2009], Simple Knowledge Organization System (SKOS) [19], Semantically-Interlinked Online Communities (SIOC) [20], Federal Emergency Management Agency (FEMA) [21], Emergency Disasters Database (EM-DAT) [22], MA-Ont [23], iContact [24].
LODE is an ontology for publishing descriptions of historical events as Linked Data, and for mapping between other event-related vocabularies and ontologies.
SKOS is a W3C recommendation to represent knowledge organization systems using the Resource Description Framework (RDF). It provides a standard data model for sharing and linking knowledge organization systems via the Web. It captures the similarity of structure and applications of knowledge organization systems and makes it explicit, to enable data and technology sharing across diverse applications.
According to [Gaur et al., 2019], the FEMA provides a glossary of terms related to disaster preparation and management [Anderson, 1999].
EM-DAT is a database gathering disaster events. According to [Gaur et al., 2019], it provides precise definitions of concepts and a categorization of disturbance-related events

[17] FOAF Vocabulary Specification 0.99: `http://xmlns.com/foaf/spec/`, visited on 2020-09-22
[18] GeoNames Ontology: `http://www.geonames.org/ontology/documentation.html`, visited on 2020-09-22
[19] Simple Knowledge Organization System (SKOS): `https://www.w3.org/2004/02/skos/`, visited on 2020-09-22
[20] SIOC Core Ontology Specification: `http://rdfs.org/sioc/spec/`, visited on 2020-09-22
[21] Federal Emergency Management Agency (FEMA): `https://www.fema.gov/`, visited on 2020-09-22
[22] Emergency Disasters Database (EM-DAT): `https://www.emdat.be/index.php`, visited on 2020-09-22
[23] Ontology for Media Resources 1.0: `https://www.w3.org/ns/ma-ont`, visited on 2020-09-22
[24] International Contact Ontology: `http://ontology.eil.utoronto.ca/icontact.html`, visited on 2020-09-22

[Jonkman, 2005].
MA-Ont is a W3C Recommendation that describes a core vocabulary of properties and a set of mappings between different metadata formats of media resources published on the Web. These mappings aim at providing metadata representations that describe the characteristics and behavior of media resources in an interoperable manner, and facilitate the sharing and reusing the metadata.
iContact is an ontology that provides basic classes and more specific properties for representing international street addresses, phone numbers, and emails. Its benefit compared to other ontologies as FOAF is that it considers details of international addresses, phone numbers, and emails.
The EMPATHI ontology has the advantage of providing a diversity of vocabulary to describe disaster situation and response activities, but not to describe plans and organizational structure elaborated during the preparedness.

Ontology-based Representation of Crisis Management Procedures for Climate Events The Ontology-based Representation of Crisis Management Procedures for Climate Events presented by [Kontopoulos et al., 2018] has been developed for decision support systems for crisis management. It covers the representation of a crisis, climate parameters that may cause climate crises, sensor analysis, first responder unit allocations, crisis incidents, and related impacts. This ontology is limited to disasters related to climate.

Discussion on ontologies for planning response The ontologies for planning address the response step and provide mainly a large vocabulary to describe response activities. However, as their goal is to results in planning, they do not give vocabulary to define plans related to response activities. Only IsyCri ontology has concepts related to plans as a procedure and provides a global view of prepared response activities. The particularity of IsyCri is to be a meta-model. A meta-model has the advantage of being composed of super-concepts linked between them by super-properties. Super-concepts and super-properties allow then to gather a vast diversity of specifications. Such a high level of description provides a base, a structure to define different models of disaster management. In addition to IsyCri ontology, it also exists some other meta-models for disaster management as a global disaster management meta-model presented in [Othman and Beydoun, 2013] and a meta-model for each phase of disaster management presented by [Othman et al., 2014] and used by [Othman and Beydoun, 2016]. Among these different meta-models, the meta-model of [Othman and Beydoun, 2013] provides the most adapted concepts related to preparedness components and their application.

3.3.1.3 Discussion

From the study of disaster management in different countries presented in appendix A.3, the main terms required for the ontology have been identified and represented in Table A.5 of the appendix A.3.4. Reusable ontologies have been identified from the analysis of the main required terms and the study of these ontologies. Table 3.6 and 3.7 summarizes the analysis of existing ontologies according to the main terms required.

On the one hand, the meta-model presented by [Othman and Beydoun, 2013] has been identified as the most suitable for the main terms representation at a high level of description. On the other hand, the Emergel, Dires and EPISECC ontologies are very complete in terms of concepts for the description of the elements contained in a plan such as the tasks and resources that can intervene in the application of a plan. Among these three ontologies, Emergel ontology is the most interesting to complement the high-level concepts of [Othman and Beydoun, 2013] because of its design related to tactical symbols. This advantage facilitates the extraction of knowledge from tactical plans corresponding to maps containing tactical symbols. Thus, the use of the high-level concepts of [Othman and Beydoun, 2013] allows the representation of a wide variety of plans based on a wide variety of tasks, roles, resources whose concepts are described by the Emergel ontology.

The knowledge base using these ontologies can then be fulfilled by knowledge extracted from data. Therefore, the next section presents the existing methods for knowledge extraction and integration.

Ontologies	Concepts					
	Resources	Task	Role	Actors	Organization	Infrastructure
Ontologies for information exchange to facilitate collaboration						
Emergel	✓	✓	✓	X	✓	✓
MOAC	✓	✓	X	✓	X	✓
INSPIRE	X	X	X	X	X	✓
Dires	✓	✓	✓	✓	✓	X
DoRES	X	✓	✓	✓	✓	X
EDXL-RESCUER	X	X	X	X	X	X
EPISECC	✓	✓	X	X	✓	X
HXL	X	✓	X	X	✓	X
Ontologies for planning response						
AktiveSa	✓	✓	X	X	✓	✓
e-response	✓	✓	X	X	✓	✓
INCA	✓	✓	X	X	X	X
IsyCri	✓	✓	X	✓	X	X
Bacarex	✓	✓	X	X	X	X
EMPATHI	X	X	X	✓	X	✓
Meta-model						
[Othman and Beydoun, 2013]	X	✓	✓	✓	✓	X

Table 3.6: Existing ontologies according to required terms (1)

Ontologies	Concepts					
	Plan	Service	Risk	Population	Disaster	Damage
Ontologies for information exchange to facilitate collaboration						
Emergel	×	×	×	×	✓	✓
MOAC	×	×	✓	×	✓	✓
INSPIRE	×	✓	✓	✓	✓	✓
Dires	×	×	×	×	✓	✓
DoRES	×	×	×	×	✓	×
EDXL-RESCUER	×	×	×	✓	✓	×
EPISECC	×	×	✓	✓	✓	
HXL	×	×	×	✓	✓	✓
Ontologies for planning response						
AktiveSa	×	×	×	✓	✓	×
e-response	×	×	×	✓	✓	×
INCA	×	×	×	×	×	×
IsyCri	✓	✓	✓	✓	✓	✓
Bacarex	✓	×	×	×	✓	✓
EMPATHI	×	✓	✓	✓	✓	✓
Meta-model						
[Othman and Beydoun, 2013]	✓	✓	✓	×	✓	✓

Table 3.7: Existing ontologies according to required terms (2)

3.3.2 Knowledge extraction and integration

The conventional approach for integrating data into a system (e.g., data warehouse, information system, knowledge base) is the Extract-Transform-Load (ETL) approach. This approach has been promoted in the 1970s for managing data warehouses and is always often used nowadays in the semantic domain as shown by works of [Bansal and Kagemann, 2015] in the context of big data integration. The extract step consists of extracting the raw values of the data. The transformation consists of structuring data into a targeted schema. The transformation step also cleans data by removing redundant or useless information, groups information, and checks information. Finally, the load step consists in the insertion of data into the targeted system.

In the context of heterogeneous data integration into an ontology, each data source is transformed into a local ontology gathering the extracted raw values. The local ontologies can either be linked to a global ontology and thus be directly loaded into the global ontology or require a new transformation based on a mapping step with the global ontology to be then loaded [Cruz and Xiao, 2005, Hacherouf et al., 2015]. This step of transformation by ontology mapping is also a process used to integrate knowledge and information represented into another ontology model.

Such a process is also used in the Semantic Web to enrich linked data, as shown by works of [Debruyne et al., 2017]. This work presents a methodology of processes for enriching linked data with a geospatial dimension, from CSV files [Repici, 2006]. The first process of this methodology that they call uplift transforms the geospatial data into RDF triples. The second one enriches data and consists of link discovery and their incorporation into RDF data. In addition to the ETL approach principle, this methodology proposes the last step to produce new datasets. This last step is called downlift and transforms the newly enriched RDF data into an enriched CSV file.

Many works in the integration of heterogeneous data concern geospatial data due to the diversity of data formats and the heterogeneity of information that they can contain. Geospatial data are also the primary data containing information related to disaster management preparedness. Thus, this section presents mainly approaches related to geospatial data integration.

In the domain of semantic and geospatial integration, the GeoKnow project [Grange et al., 2014], which aims at geographically enriching data with linked data web, gathers several tools to apply the methodology presented by [Debruyne et al., 2017]. This project uses TripleGeo [Patroumpas et al., 2014] and Sparqlify [Stadler et al., 2013] as tools to do the uplift process. The enrichment is done using LIMES [Ngomo and Auer, 2011] for link discovery and Geolift for enriching and data cleaning. The end-user of GeoKnow aims at managing linked data on the web, so no downlift step is included. Instead, the authors visualized the data in a user interface.

These examples from the Semantic Web illustrate the two main steps of integration, which are the uplift process to transform data content into RDF triples and the ontology mapping process used for link discovery in the Semantic Web, for transforming a local ontology into a global ontology in the ETL process and for knowledge integration from another ontology model.

3.3.2.1 Uplift process

The most common uplift processes are semi-automatic approaches based on a schema matching to transform data into RDF triples directly loadable into the global ontology. Such semi-automatic approaches are used in projects as Karma [Knoblock et al., 2012], which can process heterogeneous data and Silk [Volz et al., 2009] specialized for data reading of RDF, CSV and XML files to convert data into the RDF form. However, the schema matching approaches are generally specific to a data type.

Semi-automatic approaches for database Many techniques and standards are created for information integration from a database to an ontology. The W3C has developed the R2RML standard, a language to express relational databases into RDF triples [Das et al., 2012]. An RDF graph represents RDF mapping. Thus, R2RML expresses a customized mapping from relational databases to RDF data sets. R2RML can also be used to express a mapping from a CSV file as presented in the paper [Debruyne et al., 2017]. Another similar approach is DB2OWL, process presented in [Ghawi and Cullot, 2007] which aims at generating an ontology mapping from a relational database. Some tools like BOOTOX [Jimènez-Ruiz et al., 2015] have been developed to facilitate the mapping from given relational databases to extract a corresponding ontology from the database schema. The tool Sparqlify included in the GeoKnow project [Grange et al., 2014] provides an RDF view through a SPARQL query, using SPARQL to SQL translation mechanisms. Similarly, [Rodríguez-Muro and Rezk, 2015] presents an approach to access the data in a database. This approach uses R2RML, not to convert or translate the content of the database in an ontology but to translate a SPARQL request (used to request an ontology) into a SQL request. Concerning approaches used in disaster management system, SOKNOS in [Paulheim et al., 2009] presents a semi-automatic approach, which is a conversion of an interactive mapping by the user between data from database and resource ontology into F-logic rules. The COBACORE data framework [GeoPii, Integrasys, 2014] is composed of a service provider that uses a domain ontology to transform a relational database into RDF triples.

Semi-automatic approaches for geospatial data Several approaches have been developed to convert the content of geospatial data into an RDF model. Some approaches are specialized for one type of storage, like the approach GML2RDF presented in [Casado et al., 2015], which is specialized for the translation of GML format. [Bizid et al., 2014] presents another approach for GML files. This approach converts GML data sets to local ontologies using GML schemas and provides automated interlinking strategies for similarly structured database resources.

Automatic approaches It exists also some automatic uplift approaches. Some approaches are specialized for relational databases as the direct mapping presented by the W3C [Arenas et al., 2011]. Some others are able to process heterogeneous formats (e.g. databases, CSV [Repici, 2006], GML [Burggraf, 2006], shapefile [Environmental Systems Research Institute (ESRI), 1998]) as the Datalift project [Scharffe et al., 2012]. The uplift process of this project converts the input format into RDF triples (subject-predicate-object). The subject corresponds to an element of a row. The predicate is based on the column name. The object is the content of a cell corresponding to the intersection between the row of the subject and the column of the predicate. Then, it converts RDF triples into a "well-formed RDF" according to chosen vocabularies by using SPARQL CONSTRUCT queries. [Pinkel et al., 2017] presents a schema matching based on intermediate graphs obtained by transforming the two inputs corresponding to a relational database and an ontology. These two intermediate graphs are then matched. Their approach uses two matchers based on the graph structure and a lexical one. First, it creates a matching using a pairwise connectivity graph to gather pair by pair of potential nodes. It then applies a Jaccard, similarity matcher [Niwattanakul et al., 2013] and finally, applies a structure matcher using an adaptation of similarity flooding algorithms [Melnik et al., 2002]. Other approaches of automatic schema matching are presented in the survey of [Rahm and Bernstein, 2001, Rahm, 2011].

3.3.2.2 Ontology mapping process

The authors of [Choi et al., 2006] make the distinction between the three following ontology mapping:

- Ontology mapping between an integrated global ontology and local ontologies "is used to map a concept found in one ontology into a view, or a query over other ontologies (e.g., over the global ontology in the local-centric approach, or over the local ontologies in the global-centric approach)."
- Ontology mapping between local ontologies "is the process that transforms the source ontology entities into the target ontology entities based on semantic relation. The source and target are semantically related at a conceptual level."
- Ontology mapping in ontology merge and alignment "establishes correspondence among source (local) ontologies to be merged or aligned, and determines the set of overlapping concepts, synonyms, or unique concepts to those sources. This mapping

identifies similarities and conflicts between the various source (local) ontologies to be merged or aligned."

The first type of ontology mapping (between an integrated global ontology and local ontologies) can be used to access local ontologies based on the vocabulary of the knowledge base and resulting from the uplift process. However, these local ontologies are generally directly integrated into the knowledge base through, for example SPARQL Update query.

The second type of ontology mapping (between local ontologies) requires a semantic definition of relationships between concepts of the source and target ontologies. This semantic definition can be expressed in OWL or through semantic rules.

The third type of ontology mapping (in ontology merge and alignment) requires to establish correspondences between ontologies. The process to establish correspondences between ontologies is called ontology matching and aims at solving link discovery problems. Different techniques of ontology matching are reviewed in [Otero-Cerdeira et al., 2015]. Their classification is divided between the element-level and structure-level, but also between semantic and syntactic techniques. Another method of classification is to use the kind of input rather than the granularity interpretation. In this case, the classification of techniques is divided between context-based techniques, which can be semantic or syntactic, and content-based techniques, which can be terminological, structural, extensional or semantic.
[Nentwig et al., 2017] surveyed current link discovery frameworks and discussed eleven frameworks by highlighting their specificities. All of the presented frameworks support the relation *owl:sameAs*, but only the Silk Framework [Volz et al., 2009] and LIMES [Hillner and Ngomo, 2011] allow the user to specify other relations. The majority of them require manual configuration. However, four of them have a semi-automatic and adaptive linking specification based on the data set analysis and the identification of the most discriminative properties. Among the four learning-based frameworks, three use supervised learning, whereas only two utilize unsupervised learning. Concerning similarity measures, the eleven frameworks all utilize them, but only five have a structure matcher. Two of them are interesting when it comes to the geospatial domain: Zhishi.links [Niu et al., 2011] and LIMES (GeoKnow project), because they use geographical coordinates as a similarity measure. These similarity measures intervene in the primary step of link discovery, which is the ontology matching.
As shown by the study of the link discovery frameworks, the matching techniques are generally combined to obtain better results. The work of [Do and Rahm, 2002] aims at comparing the efficiency of the different types of matchers and assesses their combinations. Their benchmark highlights the efficiency of a combination of matchers and the reusability of matcher results to simplify future mapping.
This third type of ontology mapping aims at merging ontologies, as the merge of local ontologies into a global ontology to constitute the knowledge base or at aligning ontologies to

access various ontologies as it is done in the architecture federated of [Farias et al., 2015], which uses SWRL-Rule selection for ontology interoperability [de Farias et al., 2016].

3.3.2.3 Discussion

A knowledge base can be fulfilled by the integration of knowledge extracted from data. An integration process is composed of uplift and ontology mapping processes.
An end-to-end plan assessment system requires an automatic process of uplift to process heterogeneous data without requiring supervision. This requirement aims to integrate knowledge extracted from all stakeholders' data without requiring, for example, a new mapping specific to the database of stakeholders. Among the automatic uplift processes, the majority of them are specialized in one type of data [Arenas et al., 2011]. Only a few approaches allow the processing of heterogeneous data. Datalift, which is an automatic approach for heterogeneous data, has been tested to assess its usability for the proposed solution. However, the RDF graph resulting from its uplift contains only annotation properties; it has not produced an RDF graph composed of individuals linked by properties. This quality of uplift is not enough to extract disaster management knowledge. Therefore, there is a lack of the automatic integration approaches of knowledge extracted from heterogeneous data.

3.4 Discussion

The disaster management consists of four steps: mitigation, preparedness, response, and recovery. Among these steps, the response step is the most critical since human lives depend on this phase's effectiveness. The response effectiveness depends on plan effectiveness defined in upstream during the preparedness step. Therefore the preparedness step and more precisely, the assessment of plan effectiveness during the preparedness phase is an important research area to avoid problems during the response step. The plan effectiveness assessment required (1) the identification of their application conditions, (2) some metrics to quantify their effectiveness, and (3) their experimentation through exercises or computer simulation.

The identification of plan application conditions required to cluster situations for which a plan has similar effectiveness. The common features between such situations form the conditions impacting the plan's effectiveness. In the case of a large-scale plan application or extensive testing of such plans, the clustering and analysis of characteristics should be unsupervised to avoid human analysis prone to error when the number of situations and characteristics is significant. Among unsupervised clustering, the cure approach [Guha et al., 2001] appears to have the best compromise between computational complexity and clustering quality.

The works [Bayram et al., 2012] and [Larsson, 2008] define different metrics that can be used to assess the plans' effectiveness according to the four main areas defined by the Sphere Project [The Sphere Project, 2011] (health, housing, food and nutrition, water and sanitation). However, these metrics depend on the plan's purpose and should be defined within the plan's scope. It means that they must be observed during the plan's experimentation phase. The different situations characteristics of plan experimentation must be put in correlation. A common characteristics analysis can achieve such correlation in order to determine the characteristics that impact the plans. The plan's experimentation phase is carried out through exercises or computer simulations. Computer simulations have an advantage over exercises, allowing a high number of experiments to assess plans. Among computer simulations, multi-agent simulations are the most suitable for plan experiments [Mishra et al., 2019]. However, most multi-agent simulation approaches for disaster management [Christensen and Sasaki, 2008, D'Orazio et al., 2014, Zhou et al., 2012, Mas et al., 2015, Nagarajan et al., 2012, Mishra et al., 2019, Hawe et al., 2012, Saoud et al., 2006, Poveda et al., 2015] are limited. Indeed, their case-dependent simulation modeling produces a lack of adaptability from one situation to another. Yet, the testing of disaster management plans requires adaptation of simulations to address the diversity and complexity of disaster management scenarios. Therefore some approaches [Poveda et al., 2015, Kruchten et al., 2007, Christley et al., 2004] allow simulation adaptation to a diversity of action strategies. Such approaches use ontologies to model the simulation. These ontologies represent the concepts of the simulation domain at a high level of abstraction in order to allow the modeling of a wide variety of strategies. They are used to automate simulation development and have the advantage of facilitating interoperability with other systems. However, these approaches do not allow simulations to be modeled based on disaster management knowledge. Nevertheless, the use of ontologies both to represent the simulation modeling and the disaster management knowledge allows taking advantage of Semantic Web technologies (c.f. appendix A.1.2) for the adaptation of simulation modeling to disaster management knowledge as well as for the enrichment of disaster management knowledge based on simulation results. Indeed, the Semantic Web technologies can allow defining mapping and alignment between the two ontologies to design simulation modeling instances according to disaster management instances.

On the one hand, the study of disaster management ontologies shows the benefits of using high-level concepts of the meta-model presented by [Othman et al., 2014] to allow the definition of a wide variety of plans, but also its need for specification. On the other hand, it shows the advantages of the ontology Emergel [Casado et al., 2015] for its comprehensiveness in the description of concepts at a low level. These ontologies provide the terminology to describe disaster management knowledge. However, the knowledge is represented through assertions (c.f. appendix A.1.2). Some parts of knowledge are stored through data. Therefore, data of disaster management can be used to extract knowledge by creating assertions that are then added to the ontology. The study of existing knowledge extraction approaches from data has highlighted an interesting project, which is Datalift [Scharffe et al., 2012]. However, Datalift creates assertions of annotation, which is not the most adapted RDF representation to integrate knowledge.

This simulation modeling process requires the explicit representation of disaster management knowledge as well as a high-level representation of simulation modeling concepts capable of accommodating a wide variety of simulation models.

The study of approaches to facilitate simulation adaptation has identified the ontology presented by [Christley et al., 2004] as the most relevant ontology for simulation modeling. However, this ontology must be completed to specify the high-level simulation modeling concepts according to the modeling needs for disaster management simulation.

Such simulations must be composed of at least three responder agents' type (central, manager, and actor), GIS environment with information such as the demography, the critical infrastructure, area of risk. They should allow observing the 12 types of variables defined by [Bayram et al., 2012] for which plan are assessed according to their purpose.

Among different multi-agent simulation platforms (AGLOBE, Cougaar, Repast, CybelePro, SeSAm, AnyLogic, and GAMA) the GAMA platform allows real-word and GIS representation, large scale simulations, scientific simulation, general-purpose agent-based simulation scheduling and planning, natural resources and environment and thus, appears to be the most suitable platform for the simulation of plan compared with the other studied approaches. The GAMA platform does not have an agent's behavior set specific to the disaster management domain. Therefore, it is necessary to extend the agent behaviors by various functionalities typical of the disaster management domain. Such an extension can be carried out by the addition of external plugin call "skills".

Table 3.8 summarizes (1) the primary limits of preparedness and existing multi-agent simulations for the plan assessment resulting from the section 3.1, (2) approaches tackling these limits, (3) the advantages of these approaches, and (4) their limits, resulting from the reviews made in sections 3.2 and 3.3.

Limits observed in the assessment of disaster response plans	Approaches tackling the limits	Advantages of approaches	Specific limits of approaches
No evaluation of changes in the description of plans	Hierarchical clustering CURE [Guha et al., 2001]	Low complexity $O(n^2 log(n))$, high scability, suitability for large data and low sensibility to outlier	Over-segmentation of groups by considering criteria that do not impact the plan's effectiveness
Adaptability to simulate and assess the diversity of plans representation	Multi-agent simulation ontology [Christley et al., 2004]	Contain high level concepts for MAS allowing the representation of an application case diversity	Does not contain specific concepts for DM simulation
Adaptability and Reusability of MAS components for disaster management	Gama Platform [Taillandier et al., 2019]	Extensibility, Geospatial data management	Lack of agent's skills for disaster management
Plan and scenario are expressed in MAS paradigm	Disaster management meta-model [Othman et al., 2014]	Provide high concepts for representing DM plan diversity	Not in OWL, does not contain specific concepts
	Ontology Emergel [Casado et al., 2015]	Provide a complete vocabulary for plan contents	Does not allow representation of plans
	Datalift project [Scharffe et al., 2012]	Automatic	Limited to RDF annotations

Table 3.8: Relevant approaches for a knowledge-driven plan assessment through simulation and limits to overcome

4 Conceptual approach

Contents

4.1 Requirements overview . . . 102

4.1.1 Lack of change evaluation in the DM plan description . . . 104

4.1.2 Lack of adaptability to simulate and assess the diversity of plans representation . . . 104

4.1.3 Lack of adaptability and reusability of MAS components for disaster management . . . 106

4.1.4 Lack of expressive plans and scenarios . . . 106

4.2 Proposed approaches to meet requirements . . . 107

4.2.1 Assessing a plan's effectiveness and its applicability . . . 109

4.2.2 Adapting the design of simulation experiments . . . 110

4.2.3 Adaptability and reusability of MAS components for disaster management . . . 117

4.2.4 Disaster management knowledge modeling for plan assessment . . . 117

4.3 Approach for assessing the effectiveness of a response plan . . . 124

4.3.1 Modeling of the studied system . . . 126

4.3.2 Simulation modeling . . . 127

4.3.3 Programming simulations and executing the related experiments . . . 129

4.3.4 Assessing the response plan . . . 129

4.4 Summary . . . 130

Related work limits The literature review in chapter 3 shows that the assessment of disaster management plans has four main limits. The studied approaches do not assess the effectiveness and applicability context of different plans (**L1**). The most relevant approaches to identify the plan's effectiveness with their associated applicability context consider criteria that do not impact the plan's effectiveness. The non-differentiation of criteria impacting the effectiveness of a plan leads to over-segmentation. Multi-agent simulations (MAS) have been highlighted as the most suitable for experimenting with plans but lack adaptability to simulate and evaluate plan diversity (**L2**). The most appropriate adaptation approaches do not consider the specific aspects of DM plan simulation. Furthermore, the adaptability and reuse of MAS components for disaster management (**L3**) are limited by a lack of capacity of specific agents for DM simulation. Existing approaches do not allow the design of the conceptual model according to the DM knowledge domain (**L4**).

Based on the above-listed limits, this chapter outlines the requirements needed to overcome them (c.f. section 4.1), then specifies approaches meetings those requirements (section 4.2) and finally specifies a combination of these approaches for assessing disaster response plans (section 4.3).

4.1 Requirements overview

Seven main requirements have been identified to solve related works limits. The lack of changes evaluation in the DM plan description requires the identification of relevant criteria impacting plan effectiveness (R1) to overcome the issue of over-segmentation. The lack of adaptability to simulate and assess the diversity of plan representation requires the specification for DM simulation (R2) to overcome the lack of such concepts in the ontology of [Christley et al., 2004], the generation of conceptual simulation model according to disaster management plans (R3), and the generation of the programmed simulation model (R4). The lack of adaptability and reusability of MAS components for disaster management requires the extension of an agent's behaviors for DM actions (R5) to overcome the lack of such behaviors of the GAMA platform. The lack of plan and scenario expressivity requires a disaster management knowledge model combining high-level and low-level concepts (R6) to overcome this lack of approaches [Othman and Beydoun, 2013] and [Casado et al., 2015] and transforming heterogeneous data into knowledge (R7) to overcome the issue of annotations produced by the project Datalift. Table 4.1 summarizes these requirements according to the limits and approaches, to which they are related. The relation between the identified limits and requirements is further commented in the next sub-sections.

Limits observed in the assessment of disaster response plans	Approaches tackling the limits	Specific limits of approaches	Requirements
No evaluation of changes in the description of plans (L1)	Hierarchical clustering CURE [Guha et al., 2001]	Over-segmentation of groups by considering criteria that do not impact the plan effectiveness	Identification of relevant criteria impacting the plan's effectiveness assessment (R1)
Adaptability to simulate and assess the diversity of plans representation (L2)	Multi-agent simulation ontology [Christley et al., 2004]	Does not contain specific concepts for DM simulation	Specification for DM simulation (R2), generation of conceptual simulation model according to disaster management plans (R3), and generation of programmed simulation model (R4)
Adaptability and Reusability of MAS components for disaster management (L3)	GAMA Platform [Taillandier et al., 2019]	Lack of agent's skills for disaster management	Extension of agent's behaviors for DM actions (R5)
Plan and scenario are expressed in MAS paradigm (L4)	Disaster management meta-model [Othman and Beydoun, 2013]	Not in OWL, does not contain specific concepts	a disaster management knowledge model combining high-level and low-level concepts for plan assessment (R6)
	Ontology Emergel [Casado et al., 2015]	Does not allow representation of plans	
	Datalift project [Scharffe et al., 2012]	Limited to RDF annotations	Transforming heterogeneous data into knowledge through classes, properties, and individuals (R7)

Table 4.1: Overview of requirements to overcome the specific limits of approaches tackling the limits observed in plan assessment

4.1.1 Lack of change evaluation in the DM plan description

Experiments Clustering and Filtering Firstly, the analysis of design experiments requires assigning an effectiveness rate to each experiment, secondly, grouping the experiments according to their effectiveness, and thirdly identifying the common points between the different experiments to define application situations. Clustering and filtering experiments according to their effectiveness require comparing the related effectiveness values to determine whether they are similar or different. According to the study of clustering approaches by unsupervised learning, made in section 3.1.2.4 of the previous chapter, the hierarchical clustering approach CURE [Guha et al., 2001] has been identified as the most relevant to cluster plans. However, such clustering produces over-segmentation due to the consideration of criteria that do not impact the plan's effectiveness. Therefore it is necessary to identify what criteria impact the plan effectiveness for filtering them (**R1**).

4.1.2 Lack of adaptability to simulate and assess the diversity of plans representation

The computer paradigm best suited for disaster management simulation modeling is the multi-agent paradigm (c.f. section 3.1.3).

Modeling the simulation The discussion of subsection 3.2.4 has highlighted the benefits of ontologies for simulation modeling to provide flexibility and reusability in simulations. The review on ontologies for multi-agent simulation adaptability, presented in subsection 3.2.1 of the previous chapter, has shown the approach of [Christley et al., 2004] as the most adapted for modeling simulations of disaster management plans. However, this ontology only provides high-level concepts for simulation modeling. On the one hand, these high-level concepts are independent of any simulation domain, thus, it is flexible enough to be applied for disaster management. On the other hand, applying this approach to the Disaster Management domain requires a specification of the ontology for the DM simulations adapted to the plan's effectiveness assessment (**R2**).

Simulation design Simulation experiments are achieved through a simulation model executed by a simulation platform. This model is called a programmed model and is an abstraction platform-specific. It corresponds to a multi-agent abstraction but according to concepts specific to a chosen platform. There are, therefore, two possible options for simulation design:

- either, create the programmed model from the system description,

- either, create the programmed model from a conceptual model, platform-independent, created from the system description.

Among these two options, the second one is a longer approach in terms of design, but more flexible than the first one. The conceptual representation in the agent paradigm, independent of any platform, can be shared and exploited in other works that require using another platform than the one chosen for this thesis. Indeed, a platform-independent model allows a diversity of implementation and the reuse of modeling through different platforms. For this reason, the second option has been chosen to allow and facilitate the reuse of the work in the scientific community.

Simulation design requirements

Simulation design requires knowledge of computer experts. Besides, the effort to model disaster management information in the multi-agent paradigm can be considerable (several months of work for experts) to model various plans and scenarios. Similarly, the programming effort can also be very significant because of the structural changes that the modeling of plans and scenarios can imply. For example, one plan may consider physicians without dissociating their specialties (surgeons, general practitioner, dentist, anesthetist), but another plan may need to dissociate these specialties and thus require changes to the simulation programming structure. Moreover, the modeling and programming processes are subjective processes that depend on their designer. Therefore, this subjectivity can produce different biases in modeling disaster management information in the agent paradigm and during the simulation programming. Such biases can distort the plan assessment. These constraints and problems limit the plan's assessment, requiring a large number of test scenarios and adaptation of both modeling and programming.

An automatic design process based on multi-agent simulation knowledge would (1) avoid the requirements of MAS experts and efforts at each new plan assessment, and (2) provide a unique subjectivity for the overall processes guarantying at least a uniform bias for a uniform plan's assessment. Such a process requires, thus, firstly, automatic conceptual modeling of simulation from the system description of DM and based on MAS knowledge (**R3**). Secondly, it requires a programmed model specific to a simulation platform. According to the review on platform simulation made in section 3.2.3, the most suitable platforms is the GAMA platform. Therefore, the second requirement for an automatic simulation design is an automatic generation of the programmed simulation model for the GAMA platform according to the defined conceptual model (**R4**).

4.1.3 Lack of adaptability and reusability of MAS components for disaster management

The programmed model depends on the GAMA platform identified in section 3.2.3 as the most suitable for the thesis application purpose. Its automatic generation requires a set of implemented components that can be used by the platform, and that can be combined to program the model (c.f. section 3.2.4). Although GAMA provides a pre-implemented BDI agent model and a set of basic agent operations, it does not provide basic agent behaviors for disaster management. Therefore, it is necessary to extend agent skills to allow the simulation of disaster management actions (**R5**).

4.1.4 Lack of expressive plans and scenarios

Knowledge formalization As presented previously in subsection 4.1.2, the simulation design consists of abstracting a system description into the multi-agent paradigm and formalizing it to generate then, its programmed model. Therefore, this process requires a description of the system to simulate. In the context of this thesis, the system to simulate is disaster management. The DM experts elaborate on the plans and have information linking with these plans that characterize this system. The characterization of the system is, thus, based on gathering and representing the knowledge of the experts to describe the system. Such representation requires, first of all, the formalization of knowledge in the field of disaster management (**R6**). This formalization must allow:

1. the representation of the diversity of disaster management plans (**R6C1**) and
2. the representation of different tasks, actions, and other essential elements of disaster management (**R6C2**).

According to the study made in section 3.3.1, two approaches are interesting to fulfill these constraints. On the one hand, the approach of [Othman and Beydoun, 2013] satisfies the constraint **R6C1** but not the constraint **R6C2** On the other hand, the ontology EMERGEL [Casado et al., 2015] satisfies the constraint **R6C2** but not the constraint **R6C1**. Therefore, concepts of these two approaches can be reused to formalize the disaster management into an ontology, but it requires to align the different concepts and extend them for geospatial knowledge.

Knowledge integration Then, it requires to gather knowledge and represent them through the defined model automatically. Heterogeneous geospatial data containing such a piece of knowledge, it is necessary to integrate knowledge extracted from these data. From the review made in section 3.3.2, Datalift is a project providing knowledge extraction and the possibility to link it to a chosen vocabulary. However, the knowledge generated by this approach is limited to RDF annotations. Therefore, the last requirement for the plan assessment is an automatic integration of knowledge extracted from heterogeneous geospatial data (**R7**).

4.2 Proposed approaches to meet requirements

The previous section has exposed the requirements to overcome the limits of preparedness. This section proposes approaches that meet the exposed requirements and whose the combination presented in the next section aims at overcoming the preparedness limits to assess plans. Table 4.2 provides an overview of approaches proposed for each requirement identified in the previous section 4.1.

General limits of preparedness for plan assessment	Requirements	Proposed approaches
No evaluation of changes in the description of plans	Identification of relevant criteria impacting plan effectiveness (**R1**)	Analysis based on a clustering combination applied to the different simulation criteria (**A1**)
Adaptability to simulate and assess the diversity of plans representation	Specification for DM simulation (**R2**)	The SemMAS ontology (**A2**)
	Generation of conceptual simulation model according to disaster management plans (**R3**)	Knowledge-driven simulation modeling (**A3**)
	Generation of programmed simulation model (**R4**)	Knowledge-driven simulation programming for the GAMA platform (**A4**)
Adaptability and Reusability of MAS components for disaster management	Extension of agent's behaviors for DM actions (**R5**)	Agent's skills development for disaster management(**A5**)
Plan and scenario are expressed in MAS paradigm	a disaster management knowledge model combining high-level and low-level concepts for plan assessment (**R6**)	The SemDM ontology (**A6**)
	Transforming heterogeneous data into knowledge through classes, properties, and individuals (**R7**)	Automatic extraction of knowledge from heterogeneous geospatial data (**A7**)

Table 4.2: Overview of proposed approaches to solve requirements identified for automating the plan assessment

The assessment of the plan's effectiveness for different plan descriptions requires identifying relevant criteria impacting plan's effectiveness (**R1**). The proposed approach to meet this requirement is an analysis based on a clustering combination applied to the different simulation criteria (**A1**), presented in subsection 4.2.1. The adaptation of simulation experiments design for the plan's effectiveness assessment has three requirements. The first one is the specification of the ontology of [Christley et al., 2004] for disaster management simulation (**R2**). The proposed specification gives birth to a new ontology, called SemMAS (**A2**). The second and third requirements are the generation of conceptual simulation model according to disaster management plans (**R3**) and the generation of a programmed simulation model (**R4**), respectively. This thesis proposes to use knowledge to fulfill these requirements through a knowledge-driven simulation modeling approach (**A3**) and a knowledge-driven simulation programming approach for the GAMA platform (**A4**). These two knowledge-driven approaches are based on the SemMAS ontology. These three approaches (**A2, A3, A4**) are presented in the subsection 4.2.2. The adaptability and reusability of MAS components for disaster management require an extension of the agent's behaviors for DM actions and the GAMA platform (**R5**). The proposed approach consists of developing new agent's skills used by the platform for disaster management (**A5**). The subsection 4.2.3 presents this approach. The disaster management knowledge modeling for plan assessment used for the conceptual simulation modeling has two requirements. The first requirement is a disaster management knowledge model combining high-level and low-level concepts for plan assessment (**R6**). The proposed approach is to create a new ontology, called SemDM (**A6**), that uses concepts from the approach of [Othman and Beydoun, 2013] and the ontology Emergel [Casado et al., 2015]. The second and last requirement is the integration of knowledge extracted from the disaster management data (**R7**). The proposed approach is an automatic integration of knowledge extracted from heterogeneous geospatial data into SemDM (**A7**). Subsection 4.2.4 presents these two last approaches.

4.2.1 Assessing a plan's effectiveness and its applicability

Analysis based on a clustering combination applied to the different simulation criteria (A1) The criteria affecting the effectiveness of the plan depend on the variables observed during the simulation. These criteria can impact the plans in a general way (on the full effectiveness) or specific (for a specific effectiveness rate only). The proposed approach to identify the criteria impacting the effectiveness of plans is an analysis based on a clustering combination applied to the different simulation criteria (**A1**). This approach firstly applies a clustering on the effectiveness value, and secondly, a clustering on each simulation criterion observed individually. Then for each effectiveness cluster, a comparison between the clusters based on each criterion is achieved. The clustering based on effectiveness allows the gathering of simulation with similar effectiveness. The clustering based on the other criterion enables the identification of criteria impacting the effectiveness by determining similar values of criteria that produce similar effectiveness' values. The comparison aims thus to determine a redundancy between the criteria combination. This redundancy analysis is the base for deducing the criteria impacting the plan for a given effectiveness cluster. Section 6.4.2 of the next chapter explains this process in more detail.

4.2.2 Adapting the design of simulation experiments

The adaptation of simulation experiments design is based on explicit knowledge to automize the process and avoid experts and effort needs. As highlighted in the introduction of section 3.3 and in the discussion, section 3.2.4, ontologies are the most adapted to represent knowledge and to allow simulation modeling reusability and flexibility, respectively. The ontology of [Christley et al., 2004] is the most adapted to represent the multi-agent simulation domain for plan assessment. However, it must be specified according to the simulation domain, which is here, the disaster management. This ontology aims at representing knowledge related to simulation modeling and design according to disaster management. This knowledge guides the two processes of simulation modeling and design. Therefore, the knowledge required to achieve these processes plays a role in the specification of the ontology of [Christley et al., 2004]. This section presents the knowledge-driven simulation modeling and the knowledge-driven simulation programming for the GAMA platform. Finally, it presents the new SemMAS ontology, based on the ontology of [Christley et al., 2004] and specified for the domain of disaster management and the two processes of modeling and programming.

4.2.2.1 Knowledge-driven simulation modeling (A3)

The proposed approach according to related work According to the review of the literature made in section 3.2.1, the studied approaches do not generate a multi-agent conceptual simulation model automatically from a disaster management model. The standard approach of simulation modeling is an approach ad hoc that consists of gathering knowledge on the studied system from experts of its domain to allow multi-agent experts to model a conceptual model of disaster management. This thesis proposes using explicit knowledge of disaster management and multi-agent simulation to define the studied system's conceptual simulation model (**A3**). This knowledge corresponds to relations between multi-agent simulation concepts and disaster management concepts that interpret the studied system model into a conceptual simulation model. This knowledge has been acquired through the literature study presented in section 3.2.2 and discussions with experts in both domains.

The usage of ontologies for modeling the studied system in DM and the conceptual simulation in MAS allows the use of Semantic Web technologies to define their relationships and reason on it to produce instances of the conceptual simulation in MAS according to instances of the studied system model in DM. As further detailed in appendix A.1.2, reasoning can be based (1) on a formal axiomatic corresponding to the logic description and (2) on a rule-based system for deducing implicit knowledge from explicit knowledge. On the one hand, a rule-based system provides more functionalities of reasoning than only a formal axiomatic. The functionalities of a rule-based system are varied and can be extended through built-ins of rules. A rule-based system is thus the most suitable to generate instances from others. On the other hand, combining axiomatics and rules can lead to undecidability and inconsistency. It is thus, necessary to manage these risks when using a rule-based system.

A rule-based system depends on a rule language (e.g. SWRL [Horrocks et al., 2004], SHACL [Knublauch and Kontokostas, 2017]. SWRL rules may face some complex, expressive difficulties that can lead to an inappropriate reasoning [Cregan et al., 2005]. Besides, each SWRL rule execution requires a complete analysis of the ontology. Thus, in large ontologies, the rules-based on SPARQL such as SPIN [Knublauch et al., 2011], SHACL [Knublauch and Kontokostas, 2017] or ASHACL [Patel-Schneider, 2017], offer greater efficiency. Due to the size of ontologies, this thesis uses a rule-based system using SHACL-SPARQL rules. SHACL is a recommendation of the W3C and "one of the most promising schema languages" [Corman et al., 2018]. Besides, the rule-based system using SHACL allows the verification of the consistency and the activation or not of the entailment during reasoning. These two functionalities enable applying rule-based reasoning that manages and ensures the decidability of reasoning and consistency of ontology.

This approach produces a multi-agent conceptual model, independent of any platform. Nevertheless, the execution of the simulation requires a programmed model specific to a platform. The GAMA platform has been defined as the most suitable platform; therefore, the next section presents the process to obtain the programmed model for GAMA from the conceptual model.

4.2.2.2 Knowledge-driven simulation programming for the GAMA platform (A4)

The proposed approach according to related work According to the study made in section 3.2.1, approaches adapting the simulation according to a model, use the knowledge on the simulation platform and, more precisely, on a set of implemented components. This knowledge is used as a base for a process of automatic programming. Although the studied approaches do not allow the programming of disaster management simulation to be executed on the GAMA platform automatically, it is possible to create an automatic generative programming process adapted according to knowledge specific to this platform. Therefore, this thesis proposes a knowledge-driven simulation programming for the GAMA platform (**A4**). This proposed approach is composed of two steps. Firstly, it generates the representation of the programmed model into SemMAS according to the conceptual simulation model. Secondly, it uses this representation of the programmed model with a generative programming process to generate the program for the GAMA platform.

This approach uses knowledge specific to the GAMA platform corresponding to implemented multi-agent components available for this platform. It creates a representation of the programmed model inside the SemMAS ontology by creating instances of the programmed model according to instances of the defined conceptual model. This process uses a similar technique than the previous approach. It uses SHACL-rules for defining the relationship between concepts of a conceptual simulation model and concepts of a programmed simulation model for the GAMA platform. It thus produces a representation of the programmed model by rule-based reasoning. This representation of the programmed model is then used by an intelligent process to generate the program and execute it on the GAMA platform. This process uses the SPARQL query language [Prud'hommeaux and Seaborne, 2008] to retrieve the knowledge in the SemMAS ontology and design the simulation program. It would have been possible to design the intelligent process for generating the program directly from the conceptual simulation model, without requiring to represent the programmed model into the ontology. However, this option has not been chosen to allow the verification of explicit knowledge about the GAMA specific concepts and their relations in the multi-agent domain by GAMA experts.

4.2.2.3 The SemMAS ontology (A2)

Ontology's goal and scope The SemMAS ontology aims to gather knowledge on simulation design for plan assessment and represent the different models intervening in this process. This knowledge is used for (1) designing a multi-agent conceptual simulation model corresponding to a studied system of disaster management and (2) representing the programmed simulation model to generate it and execute it. Such knowledge gathers the following aspects:

1. a general representation of multi-agent simulation through concepts as a model, parameter, experiment, agent, or environment, such description corresponds to a metamodel, in which different models can be represented.

2. a representation of the chosen multi-agent model through concepts represent-

ing the agent models used (e.g., reactive, BDI agent, manager, or actor) and the environment model used to model the disaster management simulation. Such representation aims to define and instantiate the simulation model with the different agents (e.g., firefighter agent or physician agent).

3. a representation of concepts specific to the execution platform.

From the studied of related work (section 3.2.1), the ontology of [Christley et al., 2004] allows the definition of different simulation models (e.g., conceptual or programmed model) for various platforms that can be used to program and execute the model (e.g., ComputerSimulationToolkit or Software-Programming), and for model parameterization (e.g., Parameter, Distribution, Time). It is the only one that allows the representation of both a conceptual multi-agent simulation model and its programmed model. The approach of [Christley et al., 2004] proposes, thus, a metamodel for multi-agent simulation, so the terminological box describes high-level concepts for modeling multi-agent simulation. Figure 4.1 provides an overview of the ontology proposed by [Christley et al., 2004]. This terminological box gathers concepts of the first category exposed previously but does not cover concepts specific for the modeling of disaster management simulation (category 2) and those specific to the GAMA platform (category 3). For example, concerning disaster management specific modeling, this terminological box does not contain the specific types of agents at the disaster management level, which are a fundamental basis for disaster management simulation modeling. As far as the simulation platform is concerned, the TBOX provides a vocabulary base to define a specific programmed model, specific software tools but does not define a specific one. Therefore, the concepts proposed by [Christley et al., 2004] must be specified to represent multi-agent simulation for disaster management and specifically depending on the simulation platform used.

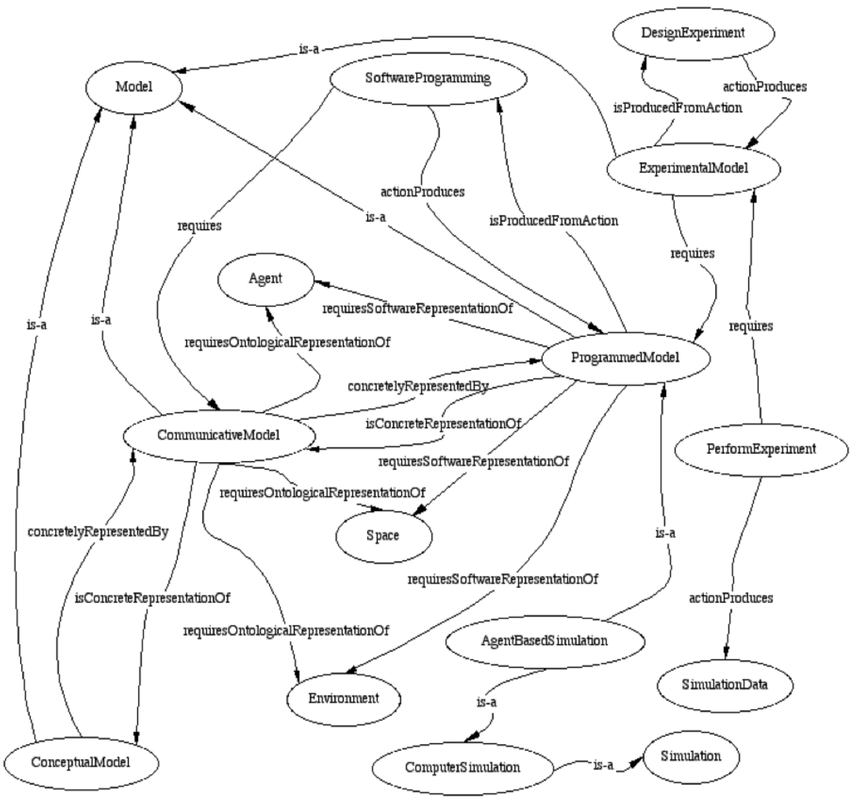


Figure 4.1: Ontology for multi-agent simulation proposed by [Christley et al., 2004]

MAS modeling for disaster management As highlighted in the related work section 3.2.2, the hierarchy of agent presented in figure 4.2, is similar to the most used hierarchy in the literature to represent the different levels of granularity for decision-making required by plans. According to the granularity level, the roles of agents are different, requiring a type of agent by levels. Although the three types of agents have different functions in the literature, they are sensibly similar, in the sense that they have practical reasoning. Similarly to the literature, three agent types are used in this thesis. However, these three types of agents follow different agent models than approaches to the literature, as explained in Section 3.2.2.1. The two first agent types that are central and manager agents have a procedural model, which is the Belief-Desire-Intention (BDI) model [Bratman, 1987]. In contrast, the third agent type (actor agent) is represented through reactive agents.

The central agent aims at organizing the collaboration and the communication between the different organizations, which intervene in the disaster response. Each of these organizations has a responsible and can, therefore, be seen as a manager. A manager decides what to do according to their role. The central agent uses its knowledge or beliefs to represent the different organizations, their role, and existing plans built in collaboration between different managers. Its role is the monitor-

ing of disaster evolution through the reception of information from other agents. It aims at deciding when to trigger a plan and what plan to trigger according to the situation. The trigger of a plan is done through an assignment of tasks to the organization responsible. It depends on the responsibilities defined by the plan.
The manager agent receives a task assignment, which is considered as a desire (a goal to achieve), with the information about the situation (defining what is the problem and where it is located). The manager uses its beliefs to determine the procedure which can be applied to attempt its desire according to available resources.
An actor agent can represent a rescuer (e.g., a firefighter) or a team of rescuers (e.g., an ambulance with three firefighters) in the real world. The actor reacts to the order received from its manager. First, it achieves the list of actions triggered by its order. Then, it gives feedback about the action execution and the situation to its manager.

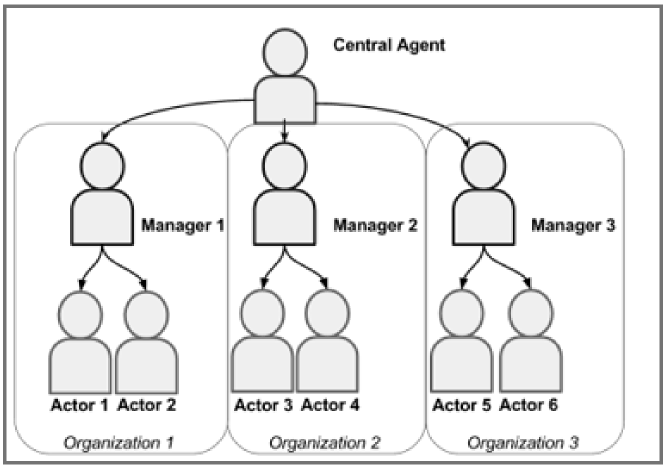


Figure 4.2: Hierarchy of agents

GAMA platform-specific knowledge The GAMA platform uses its own GAML language based on the Java programming language. An agent in the GAML language is specified as a "species", which has a set of attributes, actions, and reflexes ("Reflex") [Taillandier et al., 2019]. Any species may be nested within another species and may also inherit properties from another species. Similarly, any model is nested within an experiment. In the GAML language, model and experiment are categories of "specialized" species. The concepts of the agent paradigm are linked to the specific concepts of the simulation execution platform. Figure 4.3 shows an overview of GAML language concepts.

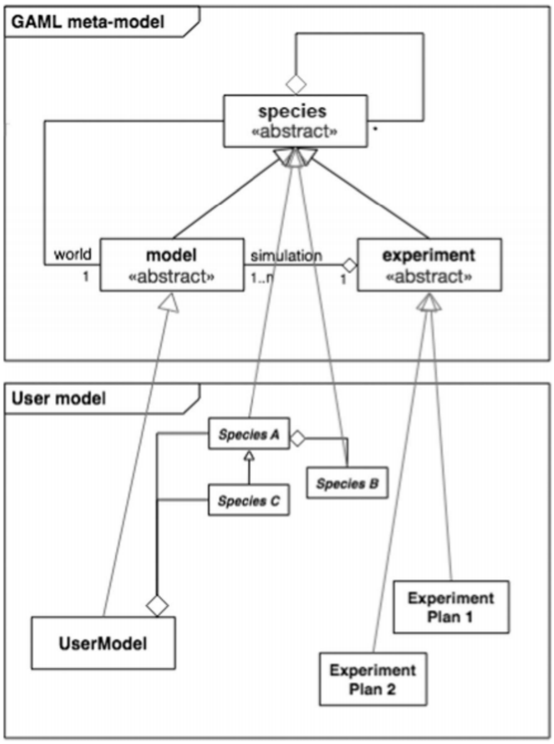


Figure 4.3: Main concepts from GAML used by the GAMA platform interlinked to concepts of multi-agent simulation [Taillandier et al., 2019]

Specification of the ontology of [Christley et al., 2004] for disaster management
From the study of the related work on disaster management simulation modeling and the GAMA platform, the SemMAS ontology (**A2**) reuses the majority of concepts from [Christley et al., 2004] and specifies further the different concepts to provide the knowledge required for the previously presented approach. The concepts integrated from [Christley et al., 2004] are concepts related to the different models (e.g., *Model*, *ProgrammedModel*, *ConceptualModel*) and those related to the agent modeling (i.e. *Agent*, *Environment*). The specifications provided by the SemMAS ontology concern mainly the following aspects:

- Concepts specific to disaster management: *Actor_agent* as a subclass of the *ReactiveAgent* concept, *Manager_Agent* and *Central_agent* as disjoint subclasses of *BDIAgent* also added as a subclass of the *CognitiveAgent* concept.

- Concepts specific to simulation platform: *GAML_model* as a subclass of the *Programmed_model* concept, *Species*, *Reflex*, *Skills*.

Besides, the Assertional Box (ABox) is enriched by the skills and reflexes implemented for the GAMA platform. Next chapter 5 presents the SemMAS ontology.

4.2.3 Adaptability and reusability of MAS components for disaster management

Agents' actions in the GAMA platform The GAMA platform provides a base of agent's actions through the use of skills methods. The platform allows the extension of agent's action through the definition of new "skills". A "skill" in GAML is composed of attributes and methods, allowing an agent to make an action or a decision. The GAMA platform contains a various set of implemented skills to defined common skills of agents (e.g.perceiving its environment, decision-making, communicating, moving, and acting). These skills can be combined to obtain complex behaviors of an agent.

Extension of agent's skills library for disaster management simulation on the GAMA platform This thesis proposes extending agents' skills library for disaster management simulation on the GAMA platform (**A5**). The goal is to allow common actions of an agent during Disaster Management. The implementation of these new skills is presented in section 5.2.2. The skills have been extended according to common actions for Disaster Management as defined in the SemDM ontology through the EMERGEL concepts. These new skills are a piece of knowledge for the simulation modeling and programming. This knowledge is thus, a part of the SemMAS ontology to define a simulation model based on implemented agent skills.

Goal of such an extension Such an extension aims to simulate various plans combining various disaster management actions through different agent's behaviors. A set of basic disaster management skills defining actions allows creating various behaviors representing various disaster management strategies. It becomes possible to simulate multiple strategies based on a set of actions that can be combined at will. It also has the advantage of being reused for other disaster management simulation works on the GAMA platform.

4.2.4 Disaster management knowledge modeling for plan assessment

4.2.4.1 The SemDM ontology (A6)

Ontology's goal and scope This thesis aims at allowing the assessment of disaster management plans. It must thus allow the disaster management community to define plans according to their own vocabulary. An ontology allows knowledge modeling, in a way that is understandable both by humans and machines. Therefore, an ontology, based on disaster management vocabulary is a solution to define plans to study. As presented previously, such an ontology requires (1) representing all types of disaster management plans (R6C1) and (2) representing all actions and other fundamental elements of disaster management (R6C2). Related work has allowed identifying the approach of [Othman and Beydoun, 2013] as appropriate for constraint R6C1 and the ontology Emergel [Casado et al., 2015] as appropriate for constraint R6C2. In addition, a disaster management plan is also related to geospatial information (e.g., the governmental echelon of its application or resources' location as the equipment's location). GeoSPARQL [Perry and Herring, 2012], an OGC (Open Geospatial Consortium) standard[1], defines a vocabulary for representing geospatial data in RDF along with an extension to the SPARQL query language for processing geospatial data. It thus allows representing geospatial data and manipulating it.

Representing all types of disaster management plans (R6C1) The metamodel proposed by [Othman and Beydoun, 2013] provides a conceptualization of disaster management. Figure 4.4 illustrates this metamodel, which contains concepts to represent procedures linked to a service and an organization. A service (i.e. the concept *DisasterActionService* in Figure 4.4) follows a procedure (i.e. the concept *DMProcedure* in Figure 4.4) and an organization (i.e. the concept *Organization* in Figure 4.4) is owner of procedures. A procedure is also indirectly linked to a role through the intermediate of a service, since a role (i.e. the concept *ActorRole* in Figure 4.4) provides a service (i.e. the concept *DisasterActionService* in Figure 4.4). This metamodel provides an interesting base for an ontology of disaster management for plan assessment. Contrary to works make later as [Othman et al., 2014, Othman and Beydoun, 2016], this metamodel is not specific to the preparedness phase but addresses all phases. In this thesis, this scope is adapted to the plan assessment that combines aspects of preparedness and response through plans experiment.

[1]GeoSPARQL - A Geographic Query Language for RDF Data (OGC) : https://www.ogc.org/standards/geosparql

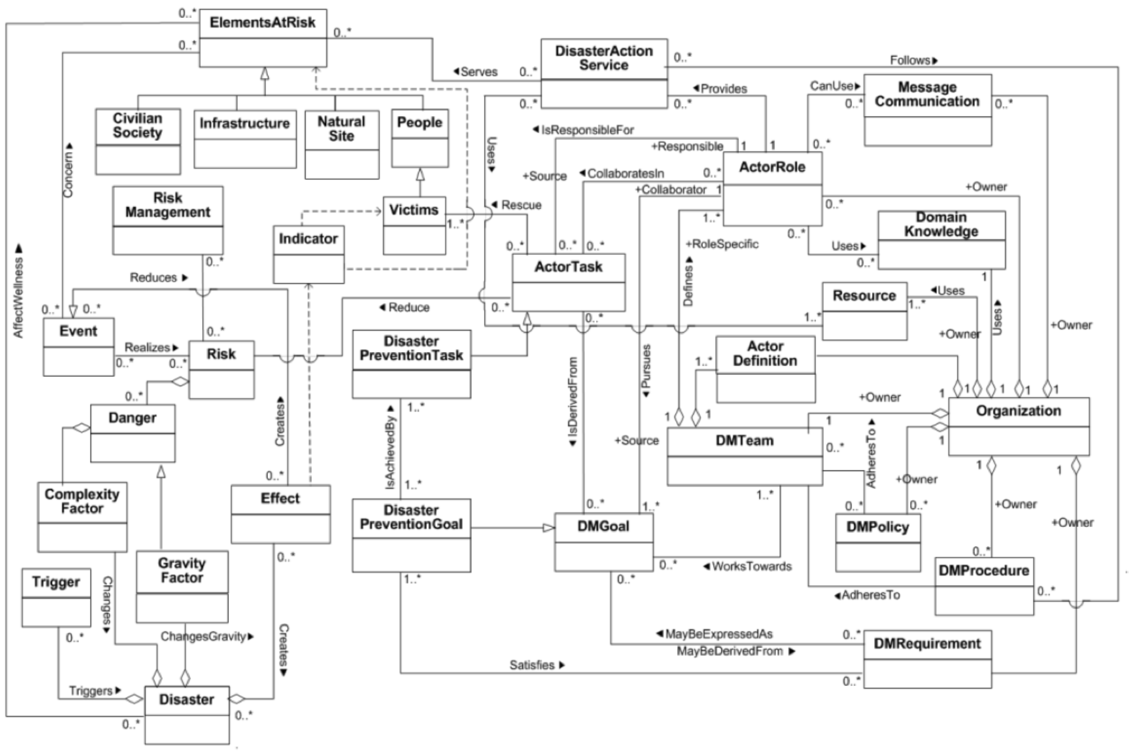


Figure 4.4: The DM Metamodel proposed by [Othman and Beydoun, 2013]

Representing all actions and other fundamental elements of disaster management (R6C2) The Emergel ontology represents a wide variety of concepts intervening in activities on the ground led by a plan. Figure 4.5 shows an overview of Emergel ontology with its highest concepts through WebVOWL[2]. This overview is composed of concepts as *Task* that is a subclass of *Activity*, *Equipment* and *Vehicle* that are resources in disaster management, *Spatial point* often defined in plans, *Infrastructure*, *Person*, and *Organisation*.

[2]WebVOWL website: `http://vowl.visualdataweb.org/webvowl.html`, visited on 2020-09-22

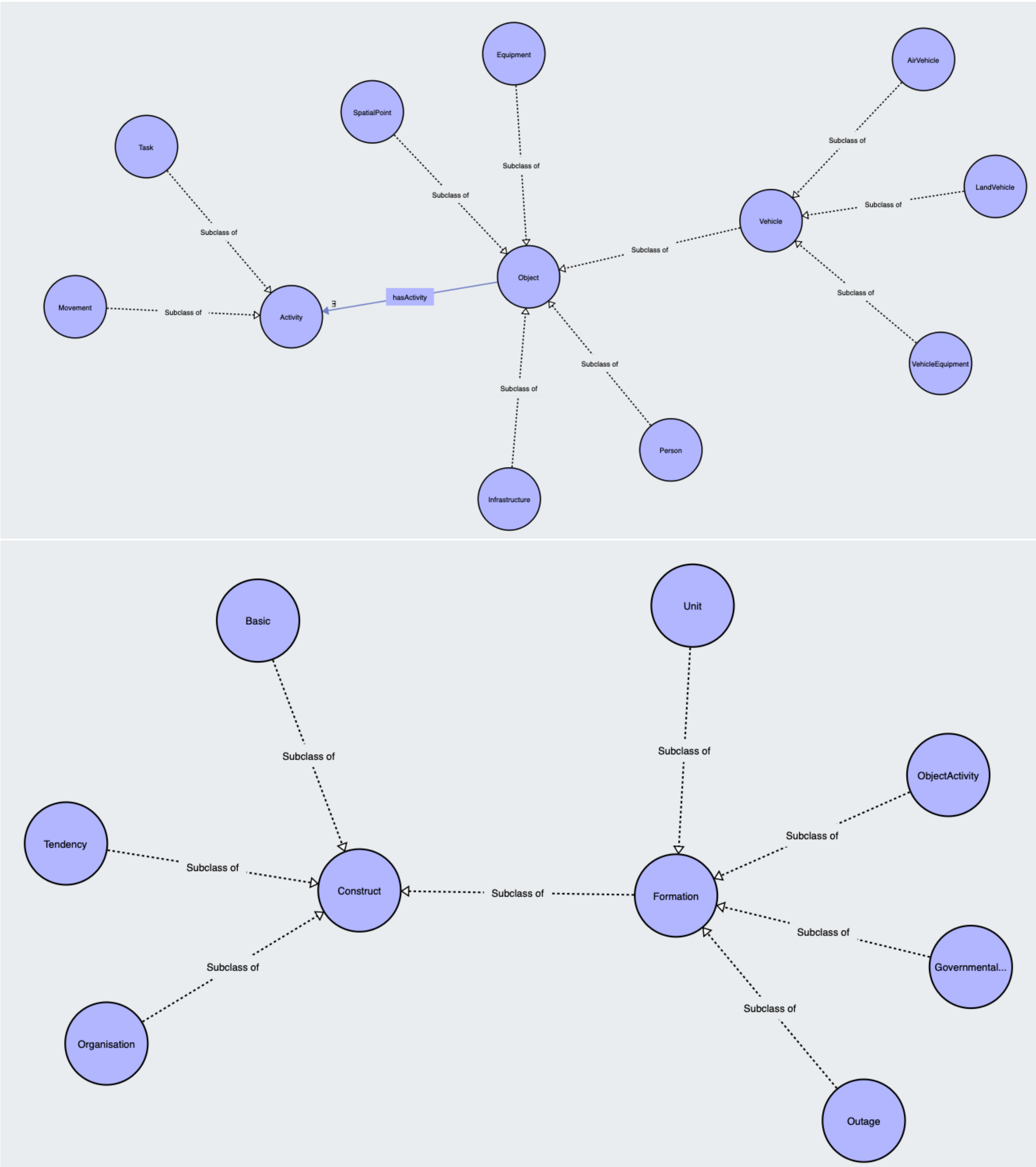


Figure 4.5: Overview of Emergel ontology through its highest concepts

These concepts are further specified through several subclasses. Figure 4.6 illustrates the subclasses of the concept *Task*. This example shows the diversity and the completeness of this vocabulary to describe activities on the ground that are described in plans.

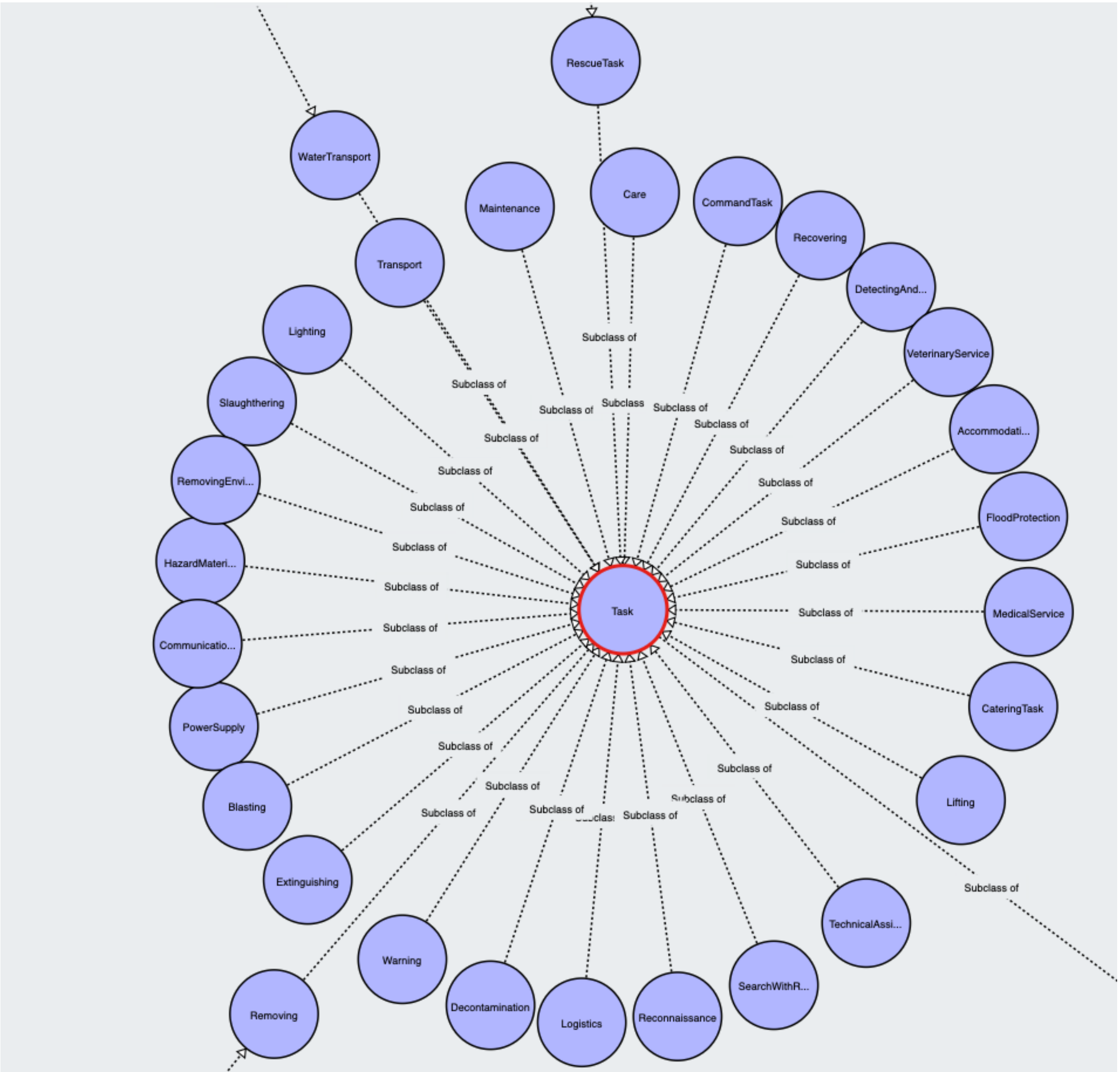


Figure 4.6: Tasks represented in Emergel ontology

GeoSPARQL vocabulary [Perry and Herring, 2012] The GeoSPARQL vocabulary is composed of three main concepts *geo:SpatialObject*, *geo:Feature*, and *geo:Geometry*. Figure 4.7 shows an overview of this vocabulary. The concept *geo:Geometry* has several subclasses allowing representing different geometries of a *geo:Feature* (e.g. *sf:Point*, *sf:LineString*, *sf:Polygon*). It has the advantage of allowing representing all kind of geospatial data and managing them through an extension of SPARQL.

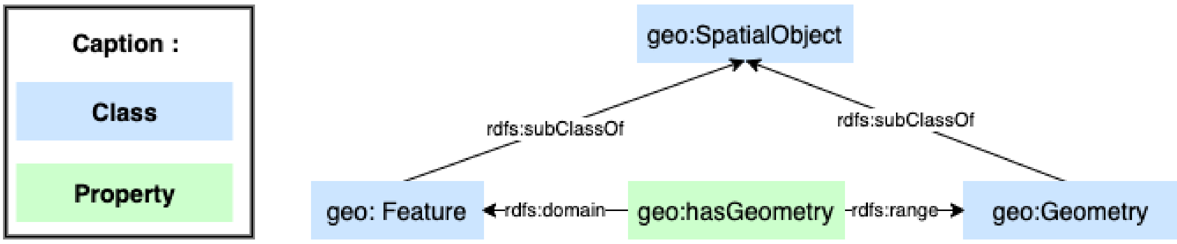


Figure 4.7: GeoSPARQL vocabulary

The SemDM ontology for plan assessment Each of these approaches for modeling knowledge related to disaster management provides interesting concepts for plan assessment. However, none can satisfy both the **R6C1** and **R6C2** constraints defined previously. The [Othman and Beydoun, 2013] approach satisfies **R6C1**, the Emergel ontology satisfies **R6C2** for the representation of emergency actions, and GeoSPARQL [Perry and Herring, 2012] provides vocabulary for geospatial data. Concepts from interesting ontologies and vocabularies can inspire a new conceptualization or be integrated and aligned. It has been chosen to create a new ontology, called SemDM (**A6**) inspired by concepts of [Othman and Beydoun, 2013] and Emergel ontology due to the fact that: (1) the metamodel proposed by [Othman and Beydoun, 2013] is not expressed through ontology, (2) the structures of the metamodel of [Othman and Beydoun, 2013] and the ontology Emergel do not fit together and need adjustments by means of intermediate concepts. Moreover, some concepts from these approaches are also unnecessary and are thus, not considered in the conceptualization of SemDM ontology. However, the vocabulary provided by GeoSPARQL has been used into the SemDM conceptualization, benefiting from its management through queries. Figure 4.8 illustrates the conceptualization of SemDM ontology through some example concepts. The detailed modeling of the SemDM ontology is presented in the next chapter 5.

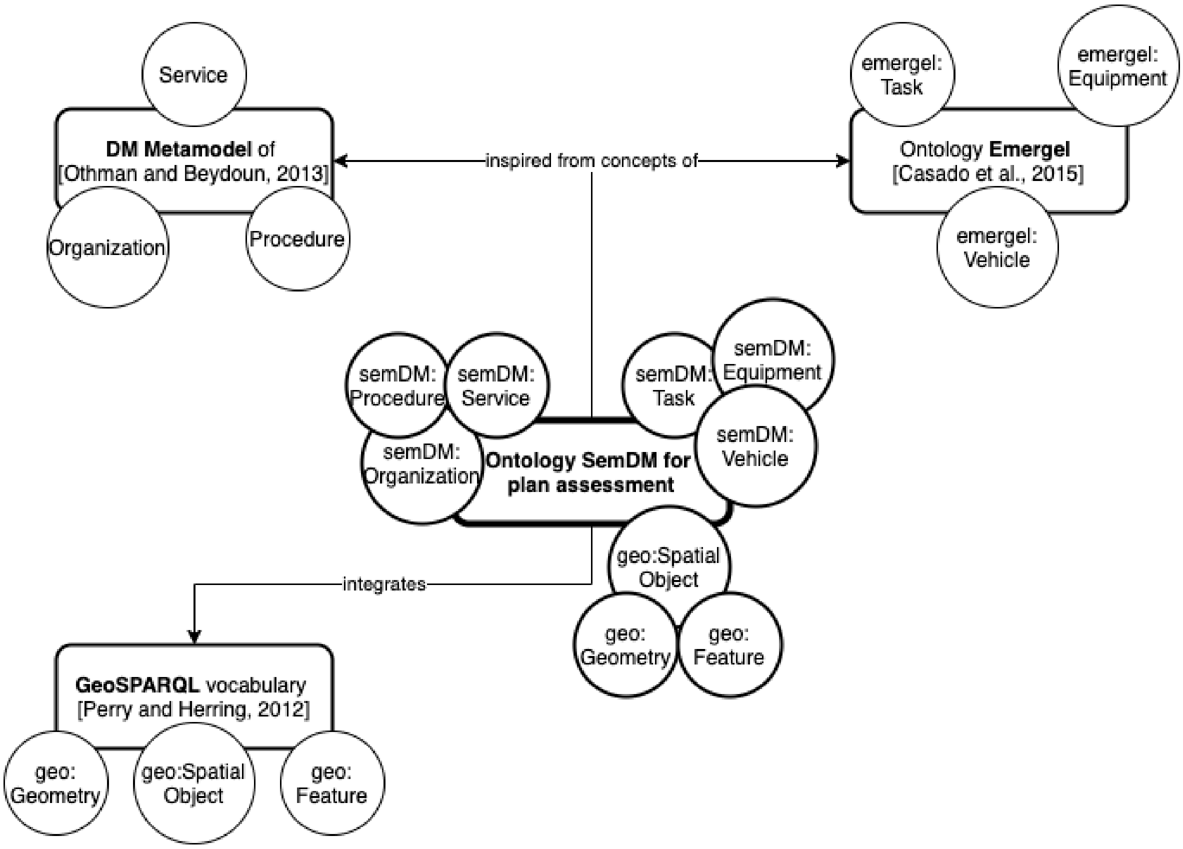


Figure 4.8: Conceptualization overview of SemDM ontology

4.2.4.2 Automatic integration of knowledge extracted from heterogeneous geospatial data into SemDM (A7)

The related work in section 3.3 has shown that knowledge can be obtained from information and data. The disaster management community has much data related to plans (e.g., resource location, a roadmap of the city, building map of a city). The knowledge related to such data plays a significant role in plan elaboration and assessment. Therefore, it is essential to automatically extract knowledge from such data to integrate it into the SemDM ontology. Data of the disaster management community are mainly geospatial data, using different formats. Inside the SemGIS project, in the context of which this thesis was done, several approaches have been developed to integrate geospatial data into the Semantic Web [Homburg et al., 2017]. Among the approaches developed in this project, a new automatic approach of integrating knowledge extracted from heterogeneous geospatial data has been developed (**A7**). This approach aims to overcome the limits of existing approaches that produce knowledge extraction results limited to RDF annotations (c.f. section 3.3.2). It has been developed to automatically gather disaster management geospatial data and benefit from Linked Open Data enrichment. This approach focuses on data with a table structure (e.g., shapefile, database), which are the most common data used in the disaster management community.

The approach (**A7**) combines Natural Language processing with geographic and semantic tools in order to extract knowledge of spatial data into a local RDF graph linked to concepts of Wikidata. The uplift process into an RDF graph of this approach is based on the table structure of data that provides structural information on data. The matching to Wikidata combines different matching techniques to obtain better matches (as exposed in section 3.3.2). It uses syntactic techniques through natural language processing and semantic matching by comparing geometries and other features to identify an individual. The implementation of this approach is presented in section 6.1 of the next chapter.

The combination of the different approaches presented in this section allows assessing disaster management plan's effectiveness and, thus, overcoming limits of preparedness. The method, combining these approaches to assess the plan's effectiveness, is presented in the next section 4.3.

4.3 Approach for assessing the effectiveness of a response plan

Previous sections have highlighted requirements to overcome the limits of preparedness and then presented a set of approaches allowing fulfilling the requirements. This section presents the method that uses the set of previously proposed approaches to assess the plan's effectiveness. The proposed method is an automated knowledge-based assessment of the plan's effectiveness using simulation to experiment the disaster management plans. Figure 4.9 provides an overview of the proposed method by highlighting the previously proposed approaches.

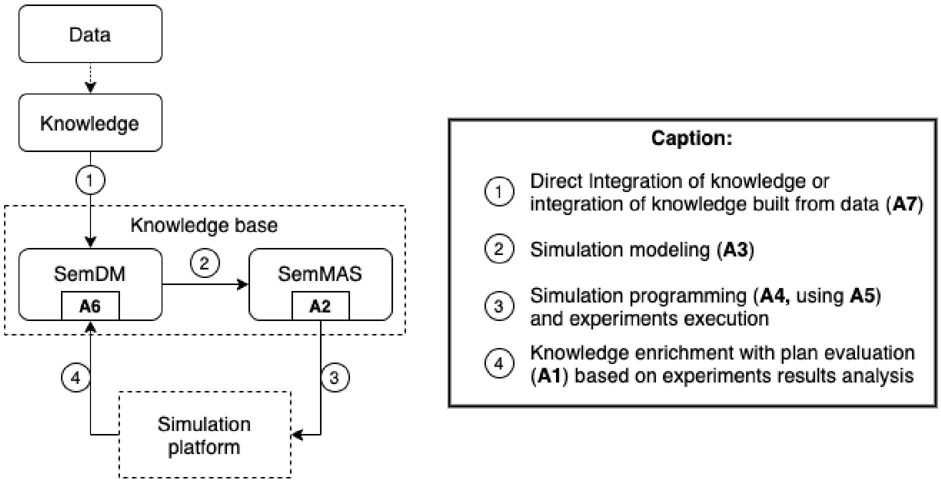


Figure 4.9: Method overview

This method uses firstly, a knowledge base containing the ontology for disaster management called SemDM (**A6**) and the ontology for multi-agent simulation (**A2**), and secondly, the GAMA platform to execute simulation experiments. It is composed of four main processes:

1. Modeling of the studied system in disaster management,
2. Simulation modeling into SemMAS,
3. Simulation programming and execution,
4. Plan assessment.

The studied system modeling of disaster management presented in subsection 4.3.1 is achieved through the integration of disaster management into the SemDM ontology, followed by reasoning using rules-based system. The simulation modeling presented in subsection 4.3.2 is composed of the conceptual simulation modeling and the programmed simulation model. The simulation programming and execution, presented in subsection 4.3.3, realizes the automatic programming for the GAMA platform and executes the simulation experiments. Finally, the plan assessment, presented in subsection 4.3.4, achieves the analysis based on the clustering combination on simulation experiment results to assess the plan's effectiveness. Enrichment of the SemDM ontology with the result of the plan's effectiveness assessment follows the analysis. Figure 4.10 shows the complete workflow of this method.

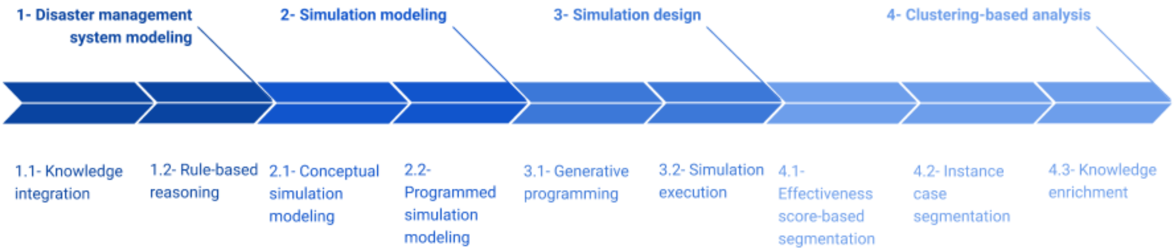


Figure 4.10: Processing workflow of the method

4.3.1 Modeling of the studied system

The studied system in disaster management contains plans that aim at being evaluated. The description of this studied system with events, in which it must be evaluated depends on the knowledge of disaster management. The presented method propose two steps to define the studied system into the SemDM ontology: (1) the integration of disaster management knowledge and (2) rule-based reasoning on the integrated knowledge to define the geospatial relations.

4.3.1.1 Knowledge integration

Disaster management knowledge comes from experts and is linked to geospatial knowledge. Geospatial knowledge, such as resource location, can be extracted from geospatial data that contain geospatial information related to disaster management. The automatic integration approach of knowledge extracted from heterogeneous geospatial data (**A7**) allows building knowledge from geospatial data based on the free knowledge base Wikidata [3]. This approach, integrating knowledge from data, implies an alignment of Wikidata and SemDM concepts to integrate knowledge built from data into SemDM. This alignment is illustrated in section 5.3 of the chapter 5 and its usage is presented in section 7.2.1.1 of the chapter 7.

4.3.1.2 Reasoning

The integration of such knowledge allows defining the studied system globally. However, the definition of the studied system according to different disaster events requires identifying plans that intervene to respond to a certain scenario of a disaster, in which plans must be evaluated. Such identification of plans according to disaster events is realized through reasoning on the SemDM ontology, using a

[3]Wikidata website: https://www.wikidata.org/wiki/Wikidata:Main_Page, visited on 2020-09-22

rule-based system. The rules are defined through SHACL (Shapes Constraint Language) [Knublauch and Kontokostas, 2017]. SHACL is a language for validating RDF graphs against a set of conditions. These conditions are provided as shapes and other constructs expressed in the form of an RDF graph. These shapes and constructs allow validating graph. In addition to graph validation, SHACL provides advanced features [4] that allow rule-based reasoning. In this advanced features, SHACL enables defining two types of rules: Triple rules (*sh:TripleRule*) and SPARQL rules (*sh:SPARQLRule*). The proposed approach uses the SHACL-SPARQL rules (*sh:SPARQLRule*, see explanation of this rule type in appendix B.2) and benefits from SPARQL advantages. Indeed, SPARQL is extendable by new SPARQL functions, which opens the possibilities of reasoning. The use of SHACL-SPARQL allows, thus, the benefit of new SPARQL functions. Section 6.1.3 of the chapter 6 presents some newly developed SPARQL functions and SHACL rules used to reason on SemDM and define the studied system. The resulting system is the subject of the simulation modeling process, presented in the next section.

4.3.2 Simulation modeling

The simulation modeling is achieved through the parsing function that uses the knowledge model SemDM to fulfill the knowledge model SemMAS. As explained previously, the knowledge model SemMAS aims at representing the domain of multi-agent simulation modeling and design. The simulation modeling process aims to create the model of multi-agent simulation corresponding to the disaster management model represented into SemDM. It first creates the conceptual model of multi-agent simulation, corresponding to the abstraction of the disaster management model into the paradigm agent. This model has the advantage of being independent of the simulation platform and can thus be implemented for different platforms. However, the simulation design requires a platform to execute the simulation model and a programmed model. The platform chosen in this thesis is GAMA. Therefore, the process creates then the programmed model using concepts of SemMAS specific to the GAMA platform according to the conceptual model previously created into SemMAS.

4.3.2.1 From the studied system model to a conceptual model of simulation

The creation of the conceptual model from the disaster management model is achieved through rule-based reasoning. Similarly to the previously explained rule-

[4]SHACL advanced features: `https://www.w3.org/TR/shacl-af/`, visited on 2020-09-22

based reasoning, it is based on SHACL [Knublauch and Kontokostas, 2017] rules and benefits from a consistency checking before updating the knowledge base. Rules used for this process, require creating new instances of the conceptual simulation model from disaster management instances. Such a creation of instances is based on URI generation. Therefore, section 6.2 presents the new rules built-in designed for allowing the URI generation and, thus, the instance creation. Figure 4.11 illustrates this process of model transformation based on an ontology-driven architecture.

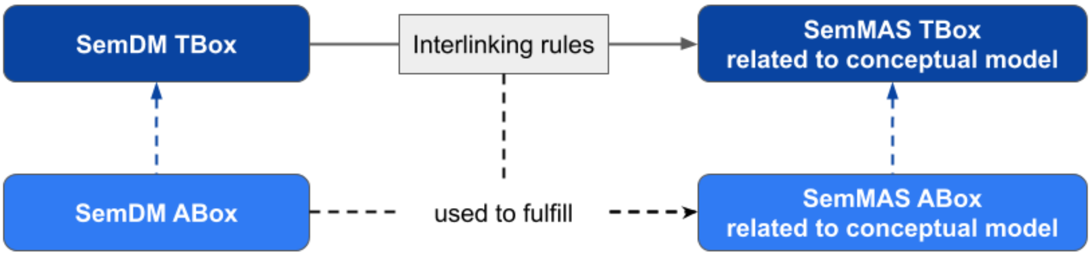


Figure 4.11: Overview of simulation explicit conceptualization process from disaster management model

4.3.2.2 From the conceptual model of simulation to its programmed model

The second process of the simulation modeling step is transforming the conceptual simulation model into the programmed simulation model. Figure 4.12 illustrates this process. The programmed model depends on the chosen simulation platform. In this thesis, the chosen platform is GAMA. Therefore, the programmed model corresponds to a GAML model. The transformation is achieved through rule-based reasoning. It uses SHACL rules and verifies the consistency of the inference before updating the knowledge base. This rule-based transformation is preceded by an inference using a reasoner (such as Pellet [Sirin et al., 2007], HermiT [Shearer et al., 2008], FaCT++ [Tsarkov and Horrocks, 2006] according to application requirements [Dentler et al., 2011]) on OWL2-EL Profile and RDF(S) to deduce the most straightforward relations between the instances of the conceptual model and the programmed model. The OWL2-El profile has been chosen because it provides the best compromise between expressivity and reasoning performance in applications using large ontologies [W3C OWL Working Group, 2012], such as those used in this thesis. This inference is a monotonic inference of open-world assumptions (c.f. appendix A.1.2).

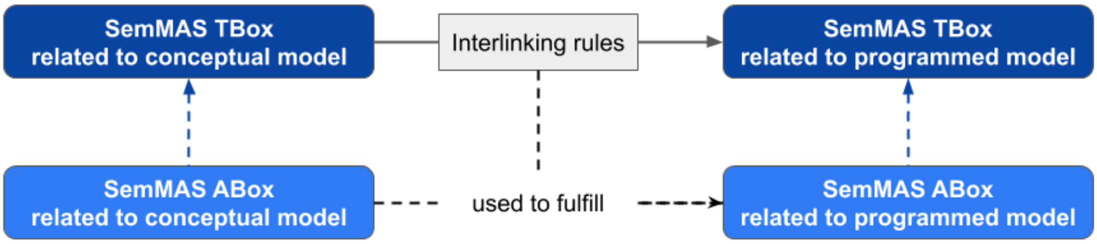


Figure 4.12: Overview of program explicit conceptualization process from the conceptual multi-agent simulation model

4.3.3 Programming simulations and executing the related experiments

4.3.3.1 Implementing the programmed model

Generative programming uses the programmed model's representation into SemMAS to generate the code of the simulation. In GAML, the code of simulation contained both the programmed model and the experimentation model. The simulation code generated is then integrated into a model library of the simulation platform. The simulation code is based on the pre-implemented skills in the library of actions, thanks to their knowledge representation in SemMAS and the dynamic simulation modeling and design.

4.3.3.2 Generating simulation experiments

The simulation execution is achieved by the GAMA platform. The execution uses the model, which has been automatically programmed and added to the library of models. Results obtained from the execution of simulation experimentation are stored into a directory for CSV files.

4.3.4 Assessing the response plan

The simulation results stored in the directory are then retrieved to be analyzed. The simulation results analysis uses an unsupervised learning approach to clustering experiments in categories according to their common points between the different simulations. These categories aim at identifying the conditions of application for a plan and computing a plan rating. The plan assessment consists of determining the scope of the plan (global or specific) and an effectiveness rating. The plan's effectiveness scope is assessed according to the effectiveness rating and the different experiment categories (corresponding to observable variables). This scope is defined as

- "global", if the effectiveness rating is similar among the different conditions of simulation experiments, or
- "specific", if the effectiveness rating changes according to conditions of simulation experiments.

In the case of a specific plan, the effectiveness rating is related to simulation experiments' conditions. Finally, the result of plan's assessment is integrated into the SemDM ontology to enrich it.

4.4 Summary

The literature review done in the previous chapter 3 has highlighted four main limits for the assessment of disaster management plans, which are **L1**: the lack of approaches to assess changes in the description of plans, **L2**: the lack of adaptability to simulate and evaluate plan diversity, **L3**: the lack of adaptability and reuse of MAS components for disaster management, and **L4**: the lack of approaches to allow designing the conceptual model according to disaster management plans. Based on these four limits, this chapter has highlighted seven main requirements to overcome the related work limits. The identification of relevant criteria impacting plan effectiveness (**R1**) is the first requirement face to the lack of changes evaluation in the DM plan description. The specification for DM simulation (**R2**), the generation of conceptual simulation model according to disaster management plans (**R3**), and the programmed simulation model's generation (**R4**) are the requirements to face the lack of adaptability to simulate and assess the diversity of plan representation. The extension of an agent's behaviors for DM actions (**R5**) aims at overcoming the lack of adaptability and reusability of MAS components for disaster management. A disaster management knowledge model combining high-level and low-level concepts for plan assessment (**R6**) and the integration of knowledge extracted from heterogeneous data (**R7**) aims at fulfilling the lack of plan and scenario expressivity.

The chapter has then proposed an approach to fulfill each identified requirement. An analysis based on a clustering combination applied to the different simulation criteria is proposed to identify relevant criteria impacting plan effectiveness. A new ontology, called SemMAS, is proposed to specify the ontology of [Christley et al., 2004] for disaster management simulation and, thus, provides an ontology for disaster management simulation modeling and design. Two knowledge-driven approaches, based on the SemMAS ontology generate a conceptual simulation model according to the disaster management ontology and the programmed simulation model corresponding to the conceptual model and specific to the GAMA platform. These approaches are a simulation modeling approach and, respectively, a simulation programming approach for the GAMA platform. The development of new agent's skills used by the GAMA platform for disaster management simulation is proposed to extend the agent's behaviors with DM actions for the GAMA platform, allowing the adaptability and reusability of MAS components. A new ontology, called SemDM and inspired from literature approaches [Othman and Beydoun, 2013, Casado et al., 2015], has been proposed to represent disaster management for the plan's effectiveness assessment. The last proposed approach is an automatic integration of knowledge extracted from heterogeneous geospatial data into SemDM ontology to gather disaster management knowledge.

Based on these approaches, it has finally presented the method of this thesis. This method combines the proposed approaches to provide an automatic knowledge-based plan's effectiveness assessment through simulation experiments. This method is based on four main steps, which are

1. modeling the studied system of disaster management through knowledge integration into SemDM ontology and rule-based reasoning,

2. modeling the simulation through the generation of the conceptual and programmed simulation model into the SemMAS ontology,

3. programming simulations and executing the related experiments through an automatic programming process, and

4. assessing the plan's effectiveness through a clustering-based analysis and an enrichment of SemDM ontology with the plan's effectiveness rating.

The next chapter presents the architecture of this method's implementation.

5 Architecture

Contents

5.1 Presentation of the architecture 134

5.2 Multi-agent simulation platform 137

5.2.1 GAMA platform and its programmed model 138

5.2.2 MAS components base 139

5.3 Knowledge base 141

5.3.1 SemDM ontology: disaster management plans' assessment 142

5.3.2 SemMAS ontology: multi-agent simulation 146

5.4 Client 153

5.4.1 Disaster management system's modeling 154

5.4.2 Event modeling 155

5.4.3 Geospatial modeling 155

5.5 Processing server 156

5.5.1 Disaster management system's modeling 157

5.5.2 Plan assessment 157

5.6 Summary 159

This chapter aims at explaining the architecture used for the implementation of the method presented in the previous chapter (c.f. section 4.3). The method has been elaborated from a set of approaches fulfilling the requirements to overcome the limits of preparedness for the plan's effectiveness assessment. In section 4.2, four approaches were proposed: clustering-based analysis (A1), knowledge-driven simulation modeling (A3), knowledge-driven simulation programming (A4), and automatic integration (A7); two knowledge models: ontologies SemMAS (A2) and SemDM (A6); and a library extension: Agent's skills (A5). Based on this method, the architecture has been designed to allow its implementation. The approaches A1, A3, A4, and A7 composing the method are processing approaches, which have been implemented on a processing server. These processes use ontologies (A2 and A6) and a simulation platform, including the library extension of Agent's skills (A5). Therefore, a knowledge base storing the two ontologies and the GAMA simulation platform have been installed on the processing server. Finally, a user interface has been implemented to allow a client to communicate with the processing server.

This chapter begins by explaining this architecture in section 5.1. It then presents its four components. Section 5.2 presents the Multi-agent Simulation Platform (C4) by explaining the specificities of the GAMA platform and the development of an agent's skills for disaster management (A5). The chapter describes the content of the knowledge base (C3) through the knowledge modeling of SemDM (A6) and SemMAS (A2) ontologies in section 5.3. It then presents in section 6.1.1, the client (C1) allowing knowledge integration and plan assessment by requesting the processing server. Finally, section 5.5 explains the role of the processing server (C2) through a sequence diagram.

5.1 Presentation of the architecture

The method presented in section 4.3 of the previous chapter is based on the processing approaches (A1, A3, A4, and A7) that use the SemDM and SemMAS ontologies and the extended agent's skills library. This method is composed of four main steps. The first step consists of modeling the studied system of disaster management through knowledge integration and rule-based reasoning. Then, the second step corresponds to the simulation modeling through a conceptual model and a representation of the GAMA platform's programmed model. The third step is the simulation design through an automatic process of generative programming and the simulation experiments' execution. Finally, the last step consists of enriching the SemDM ontology with the plan's effectiveness rating obtained from a clustering-based analysis of the simulation experiment results.

The architecture has been designed to allow the method achievement. The four method's steps are achieved through processing approaches. The architecture requires, thus, executing a set of processing. A server that aims at achieving the different processes intervening in the method is, therefore, the main component of the architecture. A server has been chosen to allow access to the system from different devices and places. A client component allows users to request the processing server through a user interface. The method's processes use the SemDM and SemMAS ontologies and a simulation platform, based on agents' skill extension for disaster management. Therefore, the processing server contains two sub-components. The first is a knowledge base to store and manipulate the ontologies, and the latter is the GAMA simulation platform to execute simulations. The architecture is thus based on four components: a client (C1), a processing server (C2), a knowledge base (C3), and the GAMA simulation platform (C4). Figure 5.1 illustrates these components and their interactions through a deployment diagram.

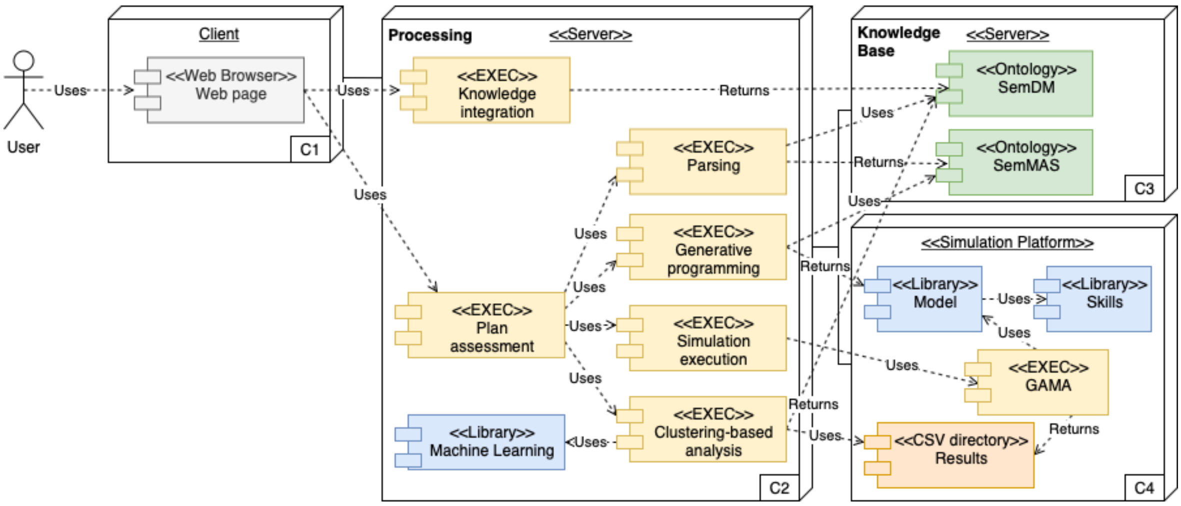


Figure 5.1: Deployment diagram of the proposed method

C1: The Client Users are experts in disaster management. They have thus implicit knowledge and data that they use to prepare and face a disaster. A user interface (described in section 6.1.1 of the next chapter) allows users to provide their data and represent their knowledge explicitly. This explicit knowledge is modeled through the vocabulary of the SemDM ontology, which is specific to the disaster management domain. Data and knowledge provided by a user are two different inputs for the server's knowledge integration process. Data are the inputs of the method's approach that extracts knowledge from heterogeneous data and integrate them into the SemDM ontology (A7). During user knowledge modeling, explicit knowledge is stored locally to achieve consistency checking before incorporating it to the knowledge base. The user's knowledge model is integrated into the SemDM ontology once the user validates its modeling, and the consistency has been validated. Only after this step, the user can request to the server to apply the plan assessment, which takes the fulfilled SemDM ontology as input. Finally, the user can visualize the SemDM ontology's content, whose the plan's effectiveness, once plan assessment has been achieved.

C2: The Processing server The processing server executes two main tasks, the knowledge integration and the plan assessment, from the user's request. These two main tasks aim at achieving the proposed method, as illustrated in Figure 5.2. The knowledge integration takes explicit knowledge and data of users as inputs and fulfilled the SemDM ontology in the knowledge base. Once a user requests the plan assessment, that means the knowledge integration is finished, and rule-based reasoning is applied to the SemDM ontology. This reasoning aims at completing geospatial relations between the disaster management modeled and geospatial knowledge of the SemDM ontology. It ends the first step of the method corresponding to the disaster management system modeling. Then, the plan assessment begins. It is composed of four primary processing: (1) a parsing process, (2) a generative programming process, (3) the simulation execution, and (4) the clustering-based analysis. The parsing process aims at modeling the simulation (second step of the method) in the SemMAS ontology. It is based on rule-based reasoning taking the SemDM ontology as an input to fulfill the SemMAS ontology. The outputs of this process are the instantiation and the modeling of the simulation conceptual and programmed models in the SemMAS ontology. The generative programming and the simulation execution processes aim at achieving the second step of the method, corresponding to the simulation design. The generative programming takes knowledge related to the programmed model's instance in the SemMAS ontology as input to produce the simulation code and the experimental plan as outputs. This code and this experimental plan are stored in a simulation model library. The simulation execution is achieved through a script provided by the GAMA simulation platform. It takes the experimental plan stored in the simulation model library and the output directory path as inputs. It provides a CSV file with simulation results as outputs. This file is stored in a results directory, whose path has been given as input. The clustering-based analysis process aims at achieving the last step of the method. It takes the result file produced by the simulation's execution as input and produces an RDF graph representing the plan's effectiveness as output. This output is integrated into the SemDM ontology to complete a plan knowledge with its effectiveness knowledge.

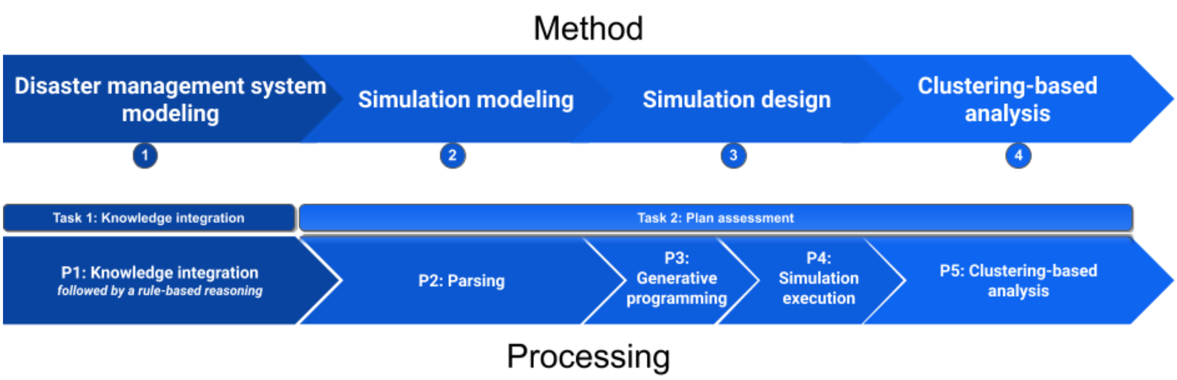


Figure 5.2: Overview of server's processes intervening in the method achievement

C3: The Knowledge base The knowledge base is installed on the server and contains the SemDM and SemMAS ontologies in the same repository. It is used to drive the simulation modeling and design from a user's inputs. The knowledge base and, more specifically, the knowledge of a plan in the SemDM ontology is enriched by the knowledge of its effectiveness resulting from the clustering-based analysis applied to simulation results. The user can visualize the content of this knowledge base through the client interface.

C4: The GAMA simulation platform The Gama platform is installed on the server. It uses the experimental plan and the simulation code contained in the simulation model library to execute the simulation experiments. Execution results are provided through a CSV file, stored in a CSV directory. This file can then be used by the clustering-based analysis to assess the plan's effectiveness and model it through an RDF graph.

The next sections detail each component with a no-linear order. They firstly present the components included in the processing server, C3 and C4. The simulation platform is presented before the knowledge base. This order aims at getting acquainted with the GAMA platform before detailing the modeling of its concepts into the knowledge base. Secondly, section 5.4 presents user inputs provided by the client. Finally, after having presented inputs and components used by the processing server, the different processes executed through it are introduced in section 5.5 and their implementation will be detailed in the next chapter.

5.2 Multi-agent simulation platform

This section aims at describing elements related to component 4 of the architecture, presented in red in Figure 5.3.

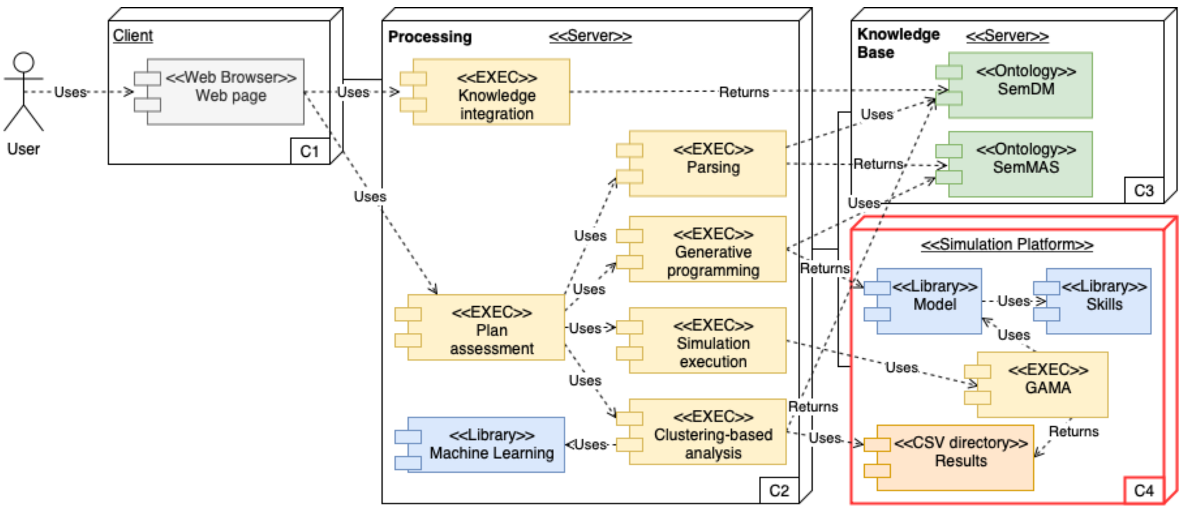


Figure 5.3: Component 4 of the architecture: the simulation platform

The multi-agent simulation aims at assessing prepared plans through their application in a simulated environment similar to the real world. The execution of simulation required a programmed model and experiment settings. The programmed model depends on the system to simulate. At the same time, the experiment settings depend on the simulation's goal and what must be observed. Both of them depend on the used simulation platform. Therefore, this section presents firstly the structure of the programmed model used by Gama. Secondly, it presents the agent's skills structure and explains the extension made.

5.2.1 GAMA platform and its programmed model

The GAMA platform executes simulation models defined through the GAML language. A programmed GAML model is organized through the four following parts, as explained in the wiki [1] of the GAMA platform:

- **model model_name**, which is the model header;
- **global {}**, which is a distinct species corresponding to the world species. It is unique and allows the definition of global attributes, actions, and behaviors;
- **species specie_name {}**, which is is a regular species. Several regular species can be defined through its attributes, actions, behaviors, and aspects. The definition of regular species can be done after the global definition and at the same level or incorporate inside the global part. In the implementation, we have preferred to split the global species from the other regular;
- **experiment experiment_name type:gui/batch/test/memorize {}**, which allows defining experiments with parameters and outputs of the simulation. The experiments are defined at the end of the file and can be multiple. There are four types of experiments: GUI for a graphical interface, BATCH to execute numerous successive simulation runs, TEST to write unit tests on a model, and MEMORIZE to store each step of the simulation in memory and to backtrack to previous steps.

The global species defines the specificities of the simulated environment. The other species allow defining disaster management stakeholders and their behavior intervening in a plan. Finally, the experiment allows designing the type of the experimental model, the evolution of input variables, also called parameters, and defining

[1] Basic skeleton of a model: `https://gama-platform.github.io/wiki/ModelOrganization#basic-skeleton-of_a_model`, visited on 2020-09-22

the observed variables. The method presented in the previous chapter 4 aims at defining such a model to assess plans through their application in different situations. It thus adapts such a model to the plan to assess. This requires adapting the agent's behaviors to fit the plan's actions as performed by stakeholders. An agent's behavior is composed of an agent's actions that correspond to an agent's skills in GAML. Therefore, such adaptation requires to have an implementation of the necessary actions made for disaster management. A certain amount of such actions are not available in GAML; therefore, this thesis proposes to extend the agent's skills allowing the necessary disaster management actions. The next subsection presents and explains this extension.

5.2.2 MAS components base

The GAMA platform allows the extension of agent action through the definition of new "skills". A "skill" in GAML is composed of attributes and methods, allowing an agent to make an action or a decision. The GAMA platform contains various implemented skills to define agents' necessary skills (e.g., perceiving its environment, decision-making, communicating, moving, and acting). These skills can be combined to obtain complex behaviors of an agent.

The agents' skills library of the architecture's component 4, is linked to the GAMA "Skill". The plugin system can easily extend these necessary skills. A skill plugin has been developed for disaster management to provide necessary skills to the agents for simulation in this domain. For example, the functionality of "transport" made by an agent, takes two parameters: (1) what is transported, (2) where it is transported. Such functionality can be created by combining the functionality of the moving skill "goTo" and "follow". The skills can then be combined to form complex behaviors. Let us take the example of the "rescue behavior". This behavior is composed of the functionality "goTo" of the existing skill "Move", followed by the functionalities "assess" and "transport" of the new skill "Acting on people". The functionality "transport" applied to this behavior is a rescuer's skill, which takes another agent as the first parameter and a position as the second parameter. In the case of the NOVI plan, which is the case study presented later in chapter 7, the rescuer agent has such rescue behavior with a casualty as the first parameter and the position of the advanced medical post as the second parameter. This usage of the implemented "rescue behavior" is illustrated in Figure 5.4. The rescue behavior allows a rescuer moves ("goTo") to a casualty, then assesses ("assess") its medical state, and transport it to the advanced medical post through its skill "goTo" and the skill "follow" of the casualty.

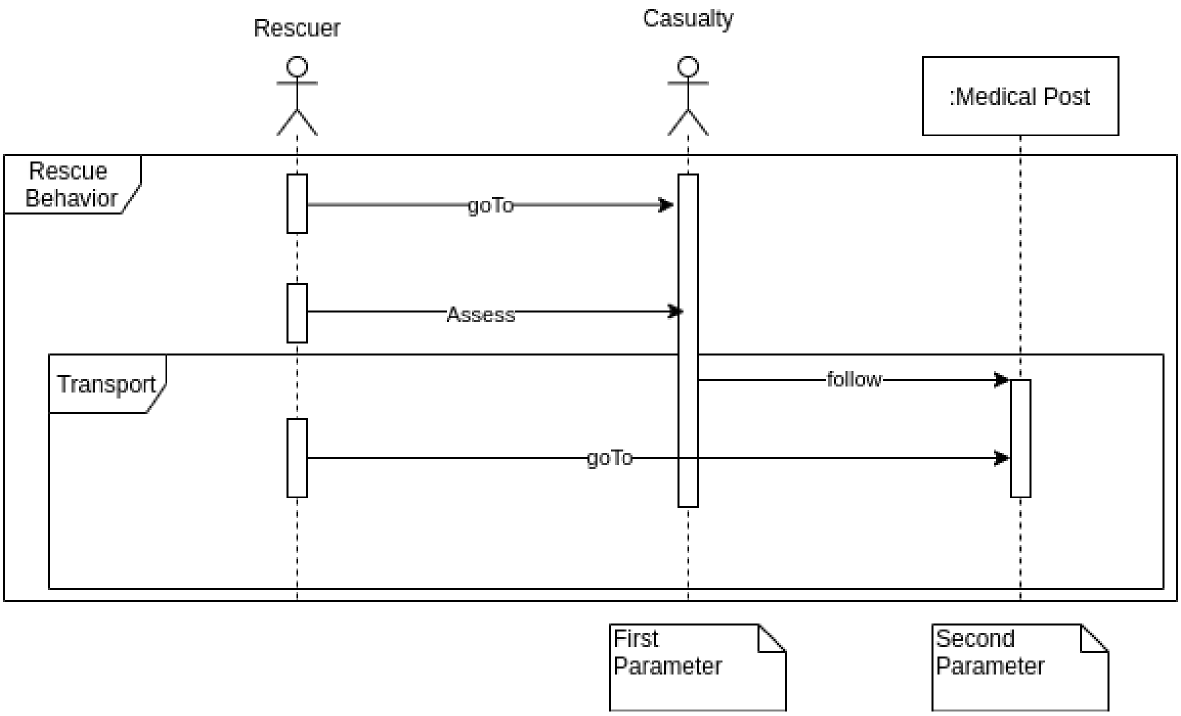


Figure 5.4: Usage example of the new implemented skill: rescue behavior

Such behavior represented in the SemMAS ontology is thus, based on skills that have been developed to extend the existing skills and provide a base of skills specific to the disaster management domain. Figure 5.5 illustrates the plugin for disaster management inside the GAMA structure of skills.

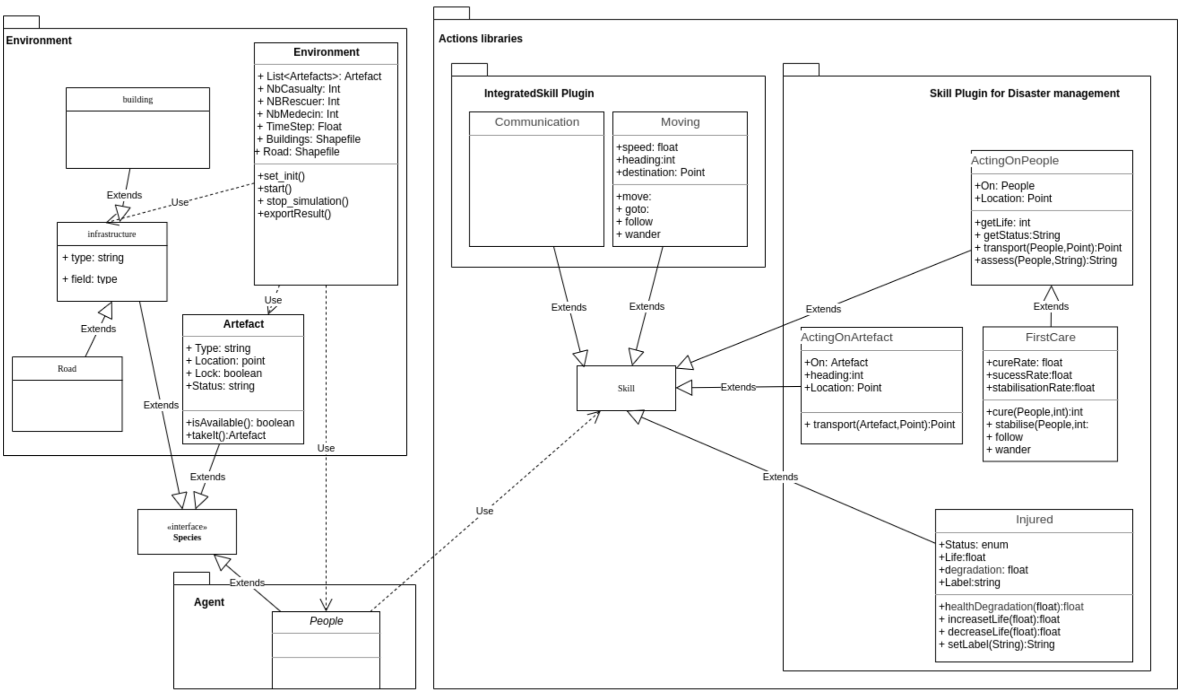


Figure 5.5: Class diagram of the root structure of the simulation

The skills have been extended according to standard actions of disaster management defined in the SemDM ontology through the EMERGEL concepts. These new skills have been represented as an agent's actions of the simulation model into SemMAS. Their representation in SemMAS has been interlinked with the EMERGEL concepts into SemDM. The representation of these skills linked to disaster management actions allows defining a simulation model based on implemented agent skills. The modeling of standard disaster management actions in SemDM and the modeling of agent's behaviors based on skills in SemMAS belong to the knowledge base, which is presented in the next section 5.3.

5.3 Knowledge base

This section presents the modeling of knowledge represented in the knowledge base corresponding to the red component 3 in Figure 5.6.

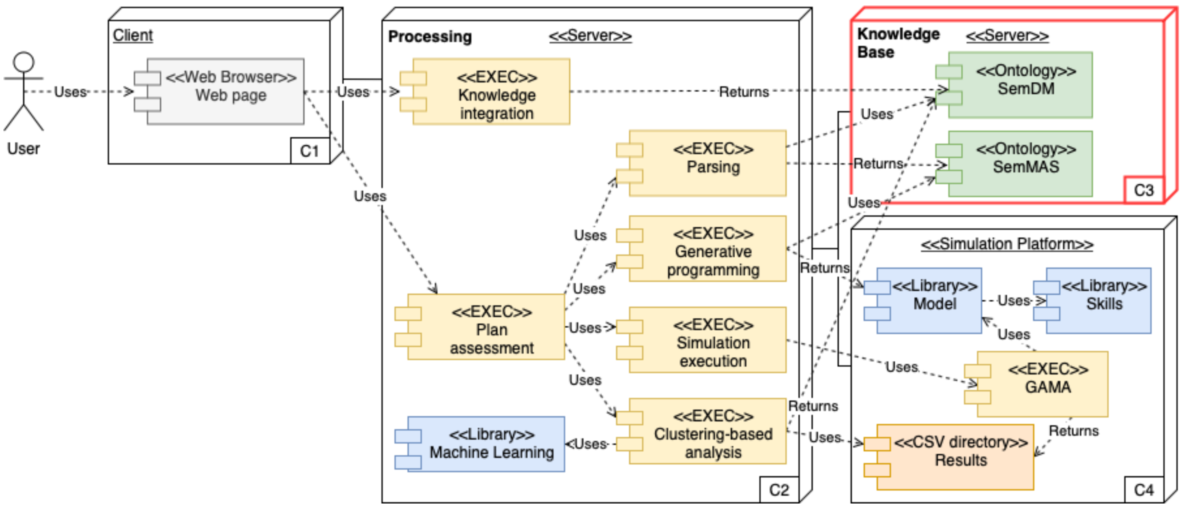


Figure 5.6: Component 3 of the architecture: the knowledge base

The knowledge modeling concerns two domains of knowledge: the disaster management domain and the multi-agent simulation domain. Therefore, the knowledge base contains an ontology for each domain, allowing sharing information from each domain. The SemDM ontology representing the disaster management domain and presented section 5.3.1 is linked to concepts from GeoSPARQL and Wikidata in the Semantic Web to facilitate data interpretation and integration. The SemMAS ontology representing the multi-agent simulation domain is presented in section 5.3.2. This ontology is composed of concepts defined according to concepts in the SemDM ontology. Such a concept definition allows the modeling of multi-agent simulation according to the disaster management model. Figure 5.7 shows an overview of ontologies represented in the knowledge base. Each ontology is expressed through a small set of its most representative concepts.

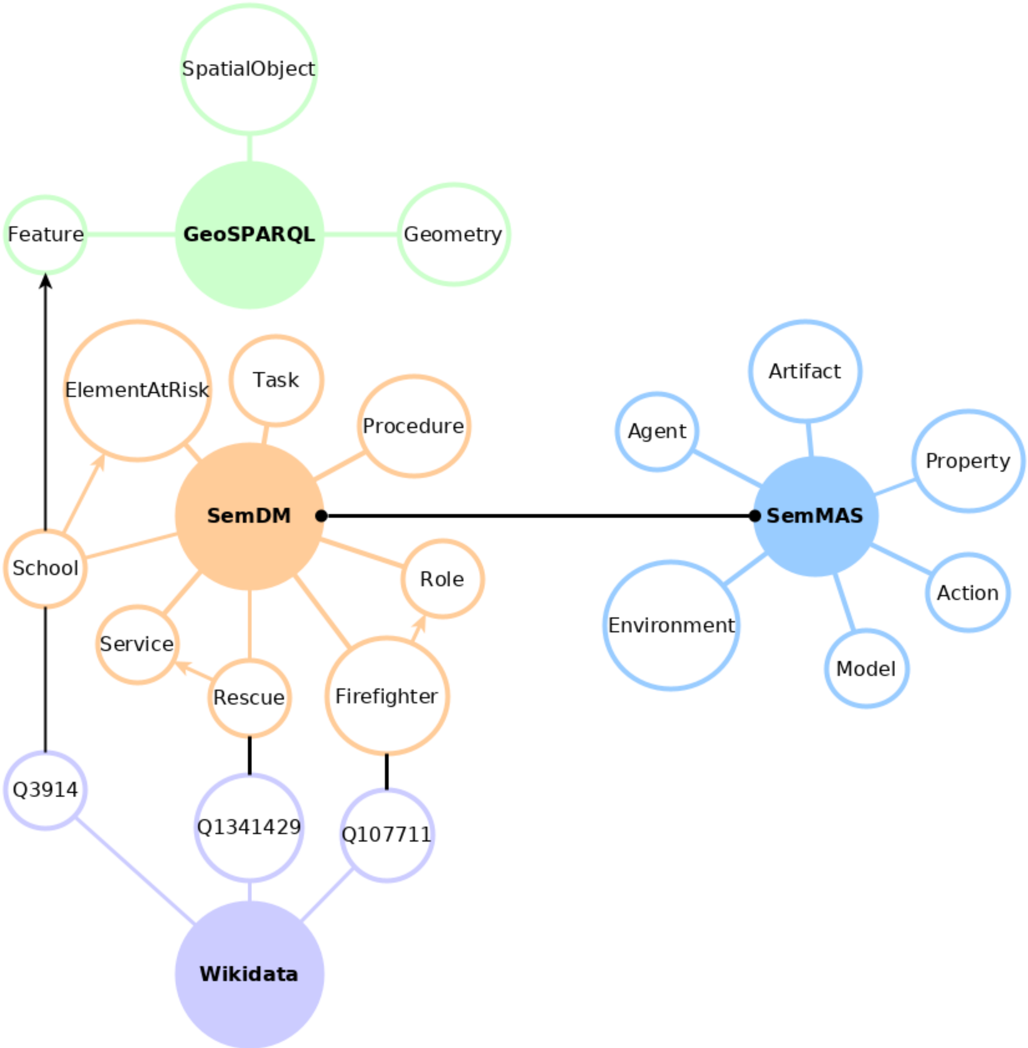


Figure 5.7: Overview of knowledge base

5.3.1 SemDM ontology: disaster management plans' assessment

Disaster management is based on expert knowledge, preparation of plans, and information on each stakeholder's resources and capacities. This thesis proposes an ontology representing the field of disaster management, called SemDM, to formalize all this knowledge. This ontology allows the definition of knowledge and plans prepared to manage a disaster. As these plans depend on events and the geospatial location, their representation in the SemDM ontology is essential.

Modeling related to the concept *semDM:Service* representing disaster action service The SemDM ontology aims to represent disaster management for plan assessment, which requires their representation and application. A plan is a procedure applied when a service is triggered to serve an element at risk. Therefore, the SemDM ontology is based on a concept *semDM:Service* that follows a *semDM:Procedure* and serves an *semDM:ElementAtRisk*. A *semDM:Service* is delivered by a *semDM:Role*. The stakeholders of disaster management are represented through the concepts *semDM:Organization* and *semDM:Person*. A *semDM:Person* belongs to an *semDM:Organization* and has one or several roles. Organizations also provide roles. Figure 5.8 illustrates the service modeling.

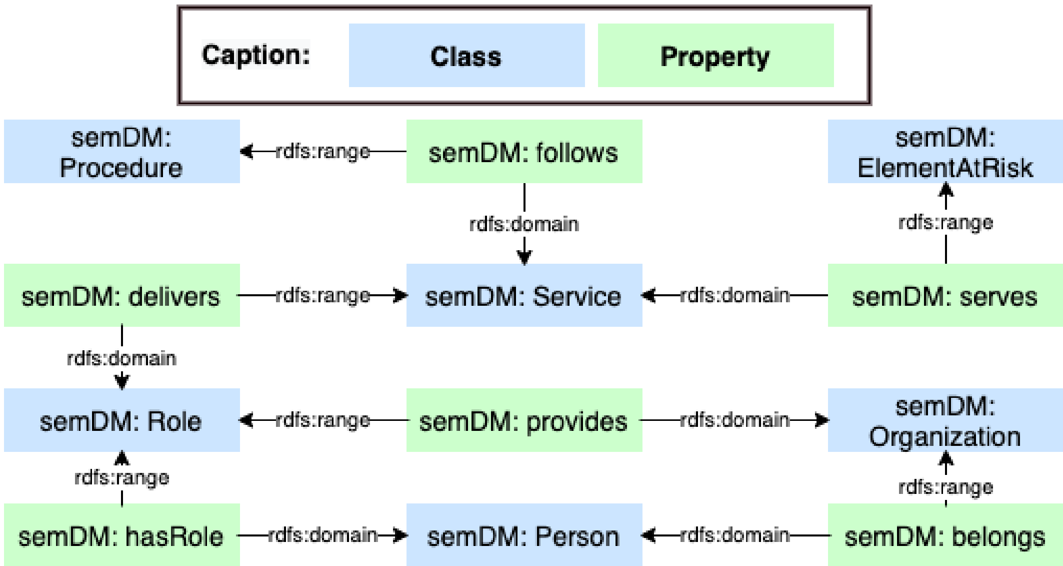


Figure 5.8: Modeling related to the concept *semDM:Service* representing disaster action service

Modeling related to the concept *semDM:Event* representing events related to a disaster The class *semDM:Disaster* is linked to the classes *semDM:Event* and *semDM:GovernmentalEchelon* through the properties *semDM:hasEvent* and *semDM:hasGovernmentalEchelon*, respectively. *semDM:Event* impacts *semDM:ElementAtRisk*. An event triggers the need for services achievement by impacting elements at risk and, thus, the application of procedures. Service triggering depends on the *semDM:GovernmentalEchelon* of the disaster. An event has a location, which is represented through the property *semDM:isLocatedAt* that links an instance of *semDM:Event* to an instance of *geo:SpatialObject*. Thanks to its location, it is possible to determine the governmental echelon of the disaster (i.e. the administrative territorial entity[2] such as a municipality[3] or a prefecture[4]), and

[2]an administrative territorial entity corresponds to the concept *Q56061* in Wikidata

[3]corresponding to the concept *Q15284* in Wikidata, which is a subclass of an administrative territorial entity (*Q56061*)

[4]a french administrative division, corresponding to the concept *Q179831* in Wikidata, which is a subclass of an administrative territorial entity (*Q56061*)

thus trigger the suitable service. Figure 5.9 illustrates the event modeling.

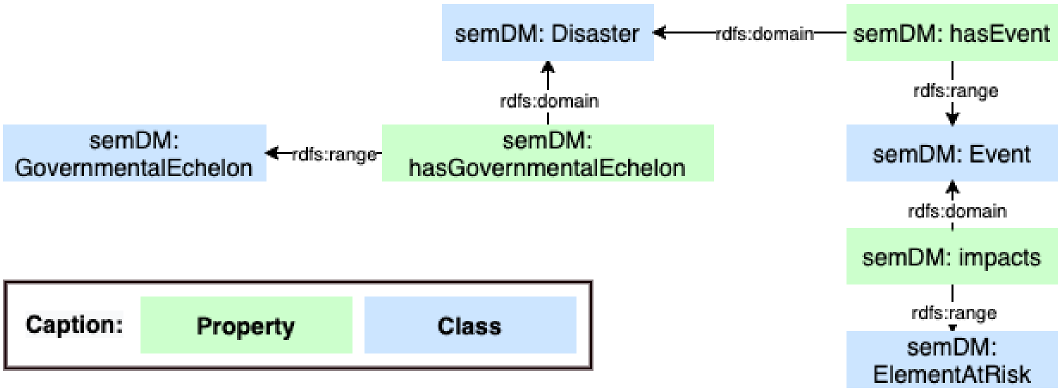


Figure 5.9: Disaster event modeling

Modeling related to the concept *semDM:Procedure* representing disaster management procedures The procedure specification allows distinguishing the different procedures types. This distinction is made according to their granularity level and through the two concepts: *semDM:Plan* and *semDM:Protocol*. A plan is a procedure with a high-level description. It is composed of tasks achieved through services that follow procedure. On the contrary, a protocol has a lower level description that means it is a procedure describing actions made on the ground. Therefore, a protocol is composed of an action sequence. An action sequence is modeled through the property *semDM:nextAction* that allows defining an action sequence. Figure 5.10 presents the procedure modeling.

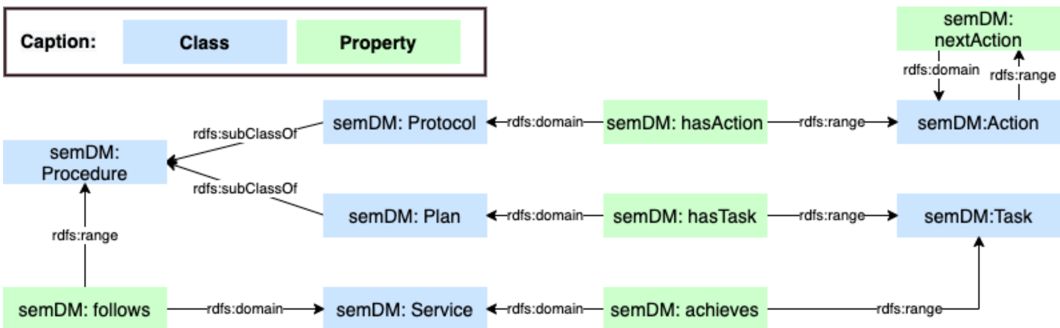


Figure 5.10: Procedure modeling

Modeling related to the plan's effectiveness assessment This thesis's goal is to enrich the knowledge base by the results of the plans'assessment. Therefore, the SemDM ontology contains a vocabulary to represent the plan's effectiveness. Figure 5.11 presents the plan's effectiveness modeling. The plan's applicability can be global or specific to situations. In the case of global applicability, the plan has global effectiveness. In the case of specific applicability, a plan has effectiveness related to a situation, which is characterized.

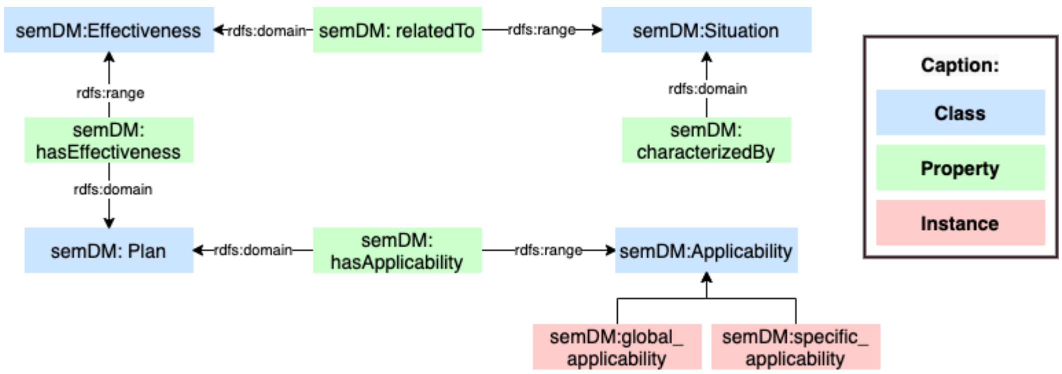


Figure 5.11: Plan's effectiveness modeling

Modeling related to the concept *semDM:Resource* representing disaster management resources The achievement of actions and tasks require resources (e.g., equipment, vehicle, role). A *semDM:Resource* is defined through a quantity or a proportion. It can be expressed in terms of *semDM:Role*, *semDM:Equipment*, or *semDM:Vehicle*, for example. Several elements such as organisations have resources. These resources are assigned to them through the property *semDM:hasResource*. Figure 5.12 presents the resource modeling.

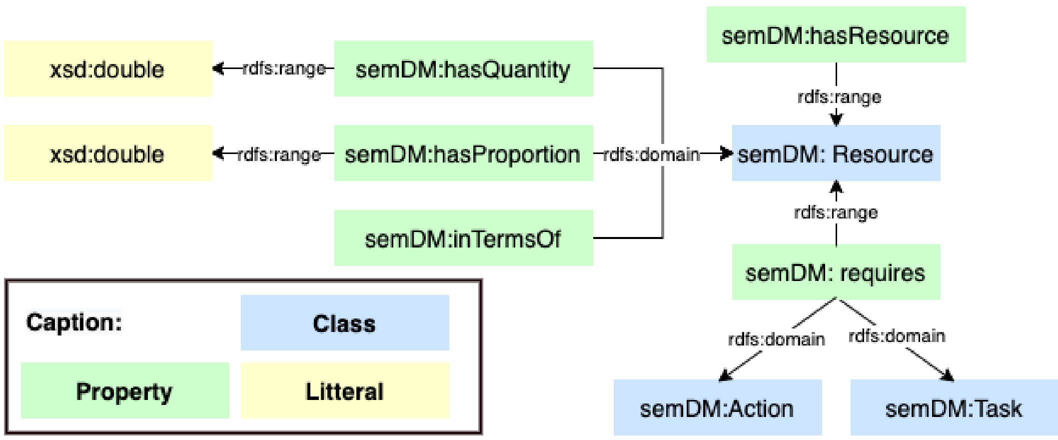


Figure 5.12: Modeling related to the concept *semDM:Resource* representing disaster management resources

Geospatial modeling The SemDM ontology uses the GeoSPARQL vocabulary [Perry and Herring, 2012] to represent geospatial information. This vocabulary allows describing a spatial object through the concept *geo:SpatialObject*. This concept has two main subclasses: a feature (*geo:Feature*) and a geometry (*geo:Geometry*). A feature can have a spatial location that cannot be precisely defined, whereas a geometry is any geometric shape, used as a representation of a feature's spatial location [Battle and Kolas, 2011]. These two concepts are associated with an object property (*geo:hasGeometry*). It exists different types of *geo:Feature* as *semDM:Infrastructure* or *semDM:GovernmentalEchelon*. Some elements of disaster management as an event or an action are not a *geo:Feature*, but have a location that can be related to a *geo:Feature* or a *geo:Geometry*. Therefore, the SemDM ontology contains the property *semDM:isLocatedAt*, which has a *geo:SpatialObject* as range to describe the location of perdurant or moving elements. Figure 5.13 presents the geospatial modeling.

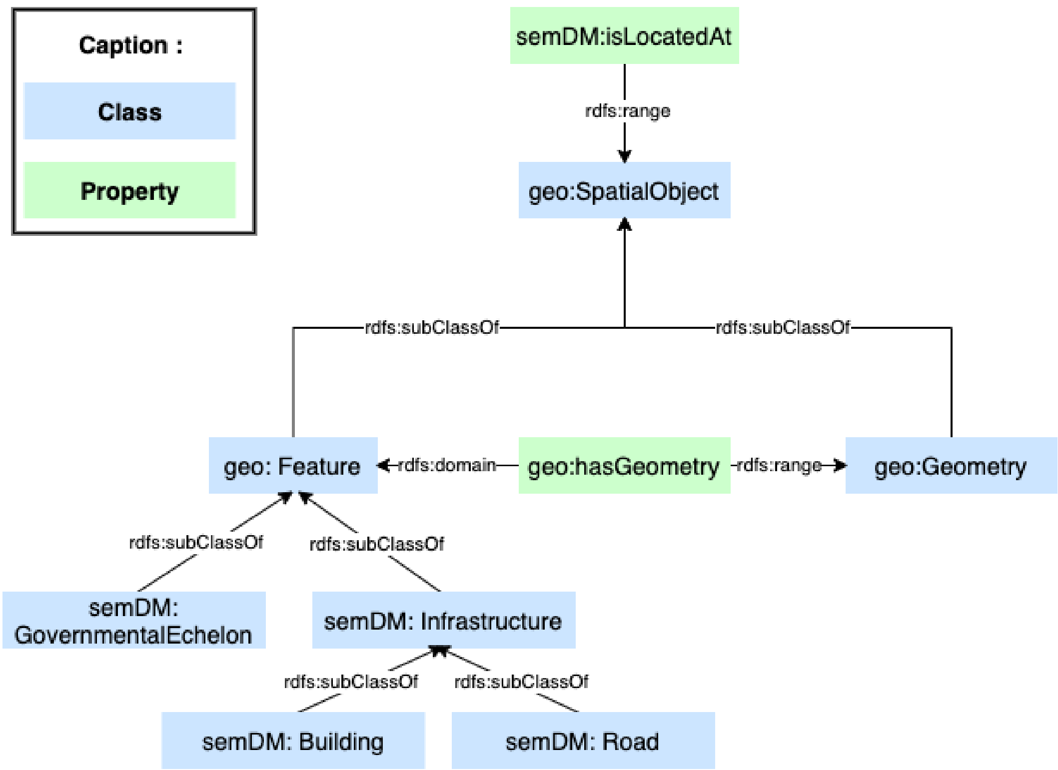


Figure 5.13: Geospatial modeling

5.3.2 SemMAS ontology: multi-agent simulation

Models representation The SemMAS ontology represents the multi-agent simulation domain. The simulation design is composed of a set of modeling steps resulting in different models. Therefore, the SemMAS ontology contains a primary concept *semMAS:Model*, which is specified by different simulation model types: *semMAS:Conceptual_model*, *semMAS:Programmed_model*, *semMAS:Experimental_model*. A conceptual model is a platform-independent model based on the concepts of the paradigm agent. A programmed model is a representation of a conceptual model, specific to a platform. It has a software representation and a software toolkit. An experimental model represents the different configurations of programmed model parameters to achieve experiments. Therefore, a programmed model has an experimental model. Figure 5.14 illustrates the representation of these different models in the terminological box of the SemMAS ontology. Such models description corresponds to a meta-model, in which different models can be represented. Such a meta-model allows defining various models and various platforms through the concept *semMAS:SoftwareToolkit*. It thus allows the definition of various conceptual models, various programmed models to execute simulation on different platforms, and various experimental models to adapt simulation experiments to a plan assessment.

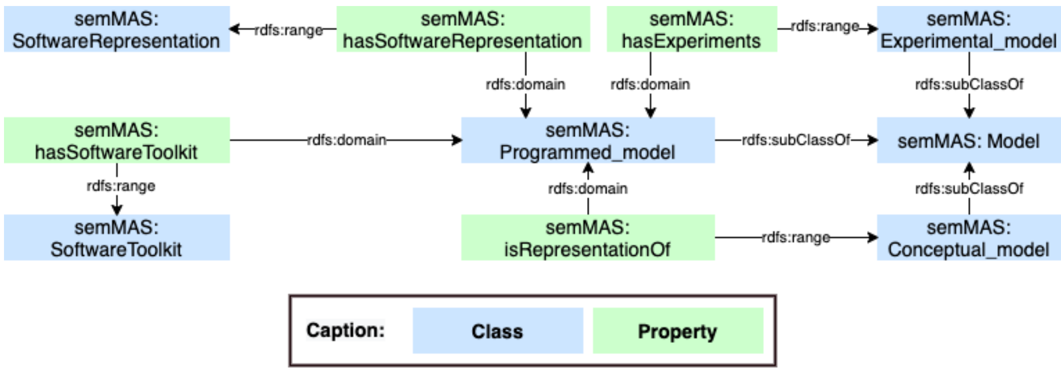


Figure 5.14: Multi-agent simulation modeling

The SemMAS ontology aims at representing the disaster management simulation models. Therefore a conceptual simulation model gathers concepts of the paradigm agents and concepts specific to disaster management simulation. These concepts aim at adapting the conceptual model according to the disaster management model described in SemDM. The execution of a simulation model requires a platform and, thus, both programmed models and experimental models adapted to the platform. In this thesis, the chosen platform is the GAMA platform. Therefore, the SemMAS ontology gathers concepts to represent the components of conceptual models, and GAMA-specific programmed models for disaster management simulations. The representation of these components is further detailed in the rest of this section.

5.3.2.1 Components modeling of the conceptual model for multi-agent simulations of disaster management

Conceptual simulation model representation The conceptual multi-agent simulation model is defined through concepts of paradigm agent. A conceptual model has an environment, which contains agents and artifacts. It is then, specified according to specificities of disaster management simulation. An artifact is a passive component of the environment that has a location. It can represent a certain type of disaster management resources as equipment or a geospatial feature. In this case, the property *semMAS:hasMembership* allows linking artifacts as building to the different buildings of the environment through the concept *geo:Feature* in the SemDM ontology. According to the related work presented in section 3.2.2.1, there are three types of agents and thus, three concepts to represent disaster management responders: *semMAS:Actor_agent*, *semMAS:Manager_agent*, and *semMAS:Central_agent*. An actor agent is a reactive agent, whereas the manager and central agents are defined as a cognitive BDI agent. The concepts *semMAS:Reactive_agent* and *semMAS:Cognitive_agent* are subclasses of the concept *semMAS:Agent*. It also contains a specific representation for a conceptual DM simulation model through the concept *semMAS:DM_model*. Figure 5.15 presents an overview of the SemMAS ontology's specification for the conceptual MAS model.

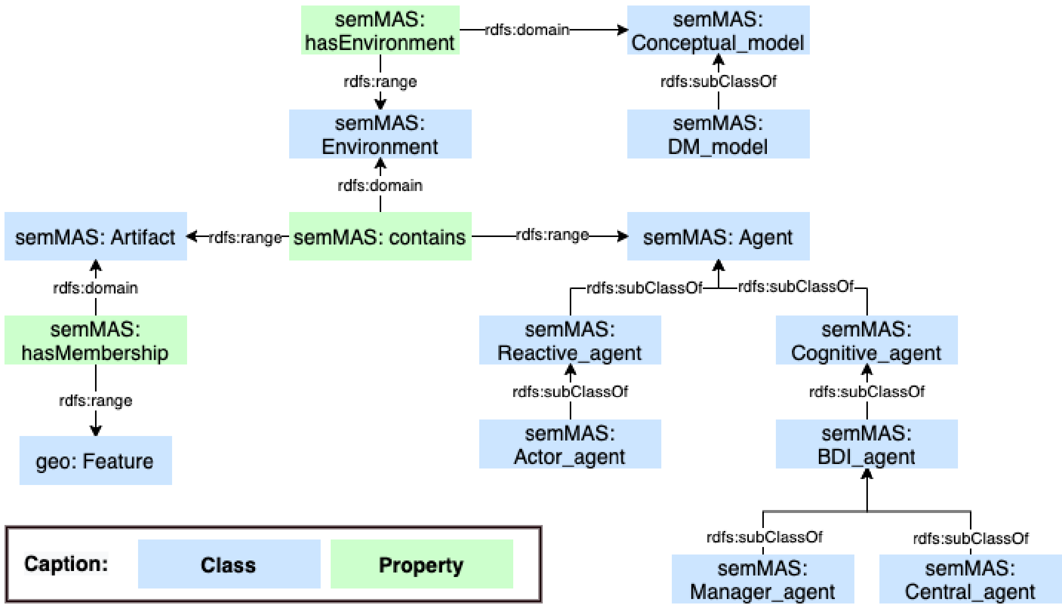


Figure 5.15: Representation of the conceptual simulation model

Agent representation Agents are active components of the environment. They perceive their environment and act on it. The SemMAS ontology allows defining an agent having a perception and behaviors. A behavior is composed of actions. In the case of a reactive agent, an action is triggered according to the agent's status. Therefore, the SemMAS ontology contains a concept *semMAS:Status* that can be assigned to an agent through the property *semMAS:hasStatus*. Agents can also have a state. The concept *semMAS:State* is often assigned to victim agents to define their health state, which is a specific state. Figure 5.16 illustrates the agent representation in the terminological box of SemMAS.

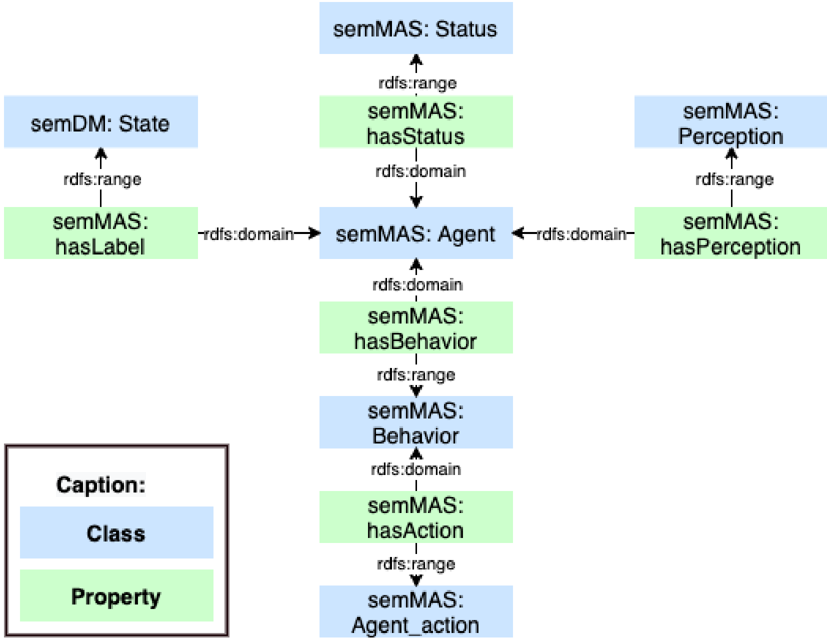


Figure 5.16: Agent modeling

BDI Agent representation The BDI agents are characterized by the concepts *semMAS:Belief*, *semMAS:Desire*, and *semMAS:Plan*. Contrary to reactive agents, the BDI agents choose actions to do according to their beliefs, desires, and plans. They select their intentions, among their desires, and according to their beliefs. An intention becomes a goal to attempt. The plans of a BDI agent aims at achieving an intention. Therefore, actions to do are chosen according to their intentions and their plans. In SemMAS ontology, a plan is defined as having an intention, which is a desire and being composed of *semMAS:Agent_action*. Figure 5.17 illustrates the BDI agent representation in the terminological box of SemMAS.

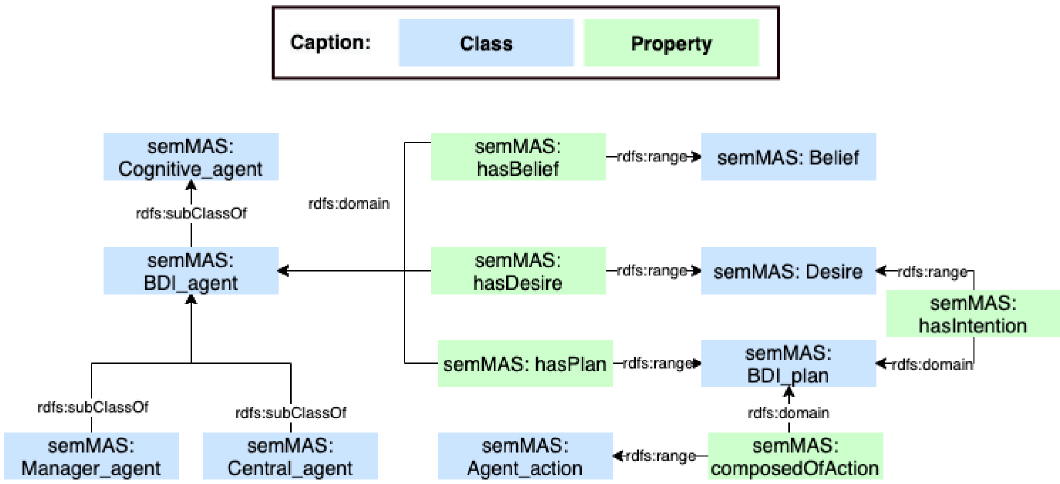


Figure 5.17: Agent BDI modeling

Action representation The agent's actions are specified according to the implemented action into the GAMA platform, whose extension of actions, explained previously in section 5.2.2. Each functionality of skills are represented as a subclass of *semMAS:Agent_action*. Properties allow the definition of their parameters. Among the most common properties, the property *semMAS:where* specifies the location of the action; the property *semMAS:what* provides its target; the property *semMAS:requiresObjective* defines the objective required to apply an action. This objective is produced by another action linked to it through the property *semMAS:produces*. It enables the creation of an action sequence made by a reactive agent. The property *semMAS:requiresStatus* assigns a required state, which can be, according to the type of action, the state of the target or the agent applying the action. Finally, an action can result (property *semMAS:results*) in a new state for the target or for the agent that owns the action. Figure 5.18 illustrates the representation of the agent's action with some examples of specifications in the terminological box of SemMAS.

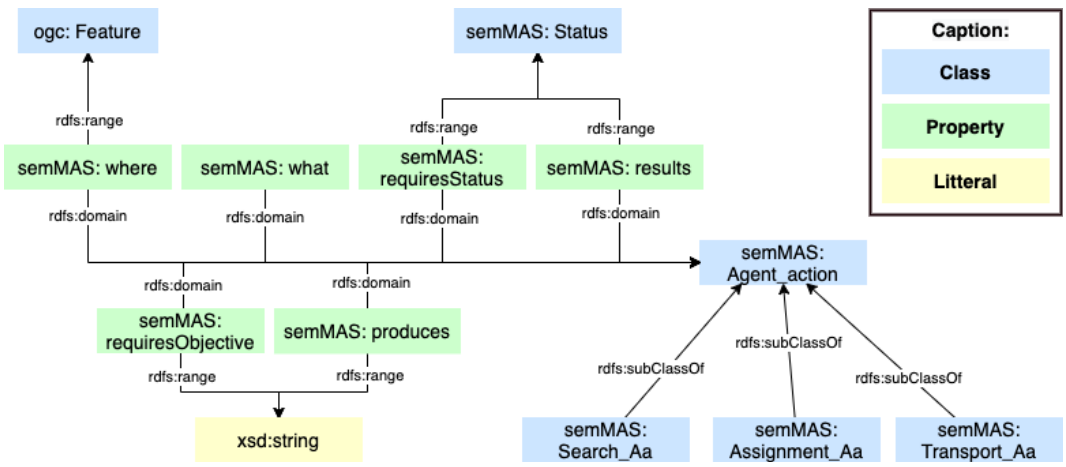


Figure 5.18: Action modeling

5.3.2.2 GAML components to program a disaster management simulation model

All concepts related to the programming into GAMA must be integrated into the SemMAS ontology to link concepts associated with the conceptual model to the concepts related to the programmed model automatically. These links between the components of the conceptual model and parts of the programmed model aim at generating all knowledge required as the input of the generative programming process explained in section 6.3.1 of the next chapter.

Programmed simulation model representation A programmed model is specific to a platform. Therefore, the properties *semMAS:hasExecutionPlatform* and *semMAS:expressedIn* associate a *semMAS:SoftwareToolkit* that gathers different platforms and a *semMAS:SoftwareRepresentation* to a programmed model, respectively. The GAMA platform is represented through the individual *semMAS:GAMA_platform*, defined as a *semMAS:SoftwareToolkit*. The language that it uses GAML is represented through the individual *semMAS:GAML*, defined as a *semMAS:SoftwareRepresentation*. The programmed model specific to the GAMA platform is represented through the concept *semMAS:GAML_model*, which is a subclass of the concept *semMAS:Programmed_model*. As presented in section 5.2.1, a GAML model is characterized by a name, a specific species called global, a set of standard species and one or several experiments. Therefore, the properties *semMAS:hasSpecies*, *semMAS:hasGlobal*, and *semMAS:hasExperiments* assign *semMAS:Species*, *semMAS:Global_species*, and *semMAS:GAML_Experimental_model* to a *semMAS:GAML_model*, respectively. The four different experiments types available in GAML are represented as subclassess of the concept *semMAS:GAML_Experimental_model*. Figure 5.19 illustrates the representation of a programmed model specific to the GAMA platform.

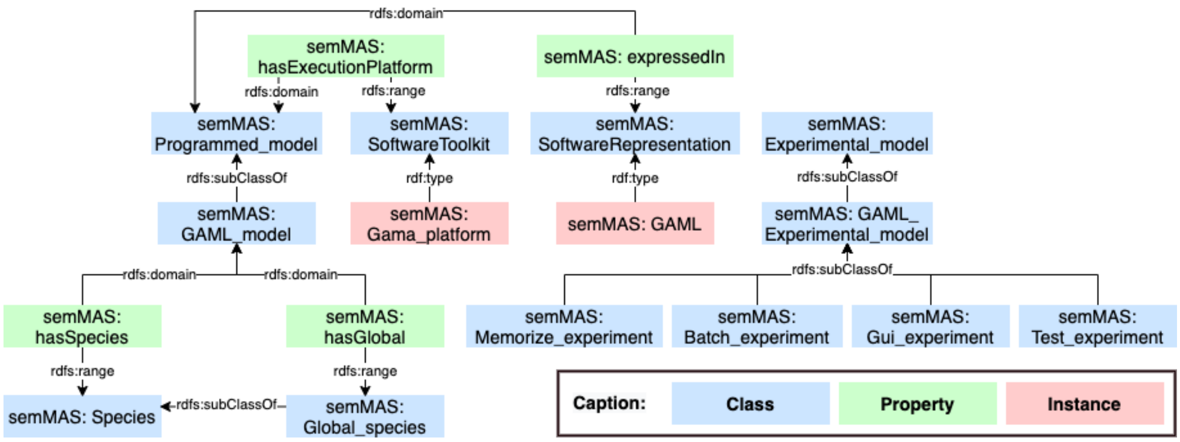


Figure 5.19: Representation of the programmed simulation model

GAML model specification The concept of the *semMAS:GAML_model* is defined as having the instance *semMAS:GAML* as software representation, the instance *semMAS:GAMA_platform* as execution platform and at least one *semMAS:GAML_Experimental_model*. This specification is illustrated through the Manchester syntax in Code 5.1.

```
GAML_model:
expressedIn value GAML and
hasExecutionPlatform value GAMA_platform and
hasExperiments min 1 GAML_Experimental_model and
hasGlobal min 1 Global_species
```

Code 5.1: GAML model specification through the Manchester syntax

Species representation The concept *semMAS:Species* in GAML is used to define both artifacts and agents. Its definition uses the attribute and property associated with the agent or the artifact that it represents. Species have the specificity to be able to inherit from another species. This relation can be defined through the property *semMAS:herits*. When a species represents an agent, the agent's action becomes reflex of a species by associating them through the property *semMAS:hasReflex*. A reflex associated with an agent implies that the agent has specific skills, that can be associated through the property *semMAS:hasSkill*. Finally, GAML provides a concept specific to define a BDI agent, called *semMAS:SimpleBDI*. Similarly to a BDI agent, it is characterized through beliefs, desires, and plans. Therefore, the concept *semMAS:SimpleBDI* is defined as a subclass of *semMAS:BDI_agent*. Besides, it can have rules that can be associated through the property *semMAS:hasRule*. Figure 5.20 illustrates the representation of a species.

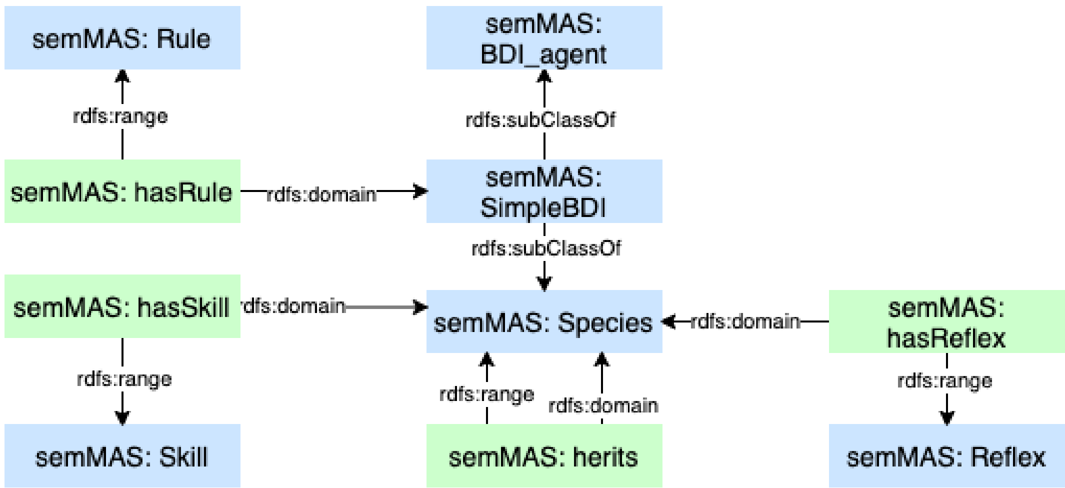


Figure 5.20: Species modeling

Skill representation The different types of reflex, corresponding to the agent's action, are functionalities of implemented agent's skills. The definition of a species representing an agent in GAML requires specifying its associated skills. Therefore, the SemMAS ontology contains an individual of the concept *semMAS:Skill* for each implemented skill in GAML. The association of skills to an agent is managed through reasoning according to the reflex that an agent has. It aims at generating the programmed model for the generative programming, as it is further explained in section 6.2.2 of the next chapter.

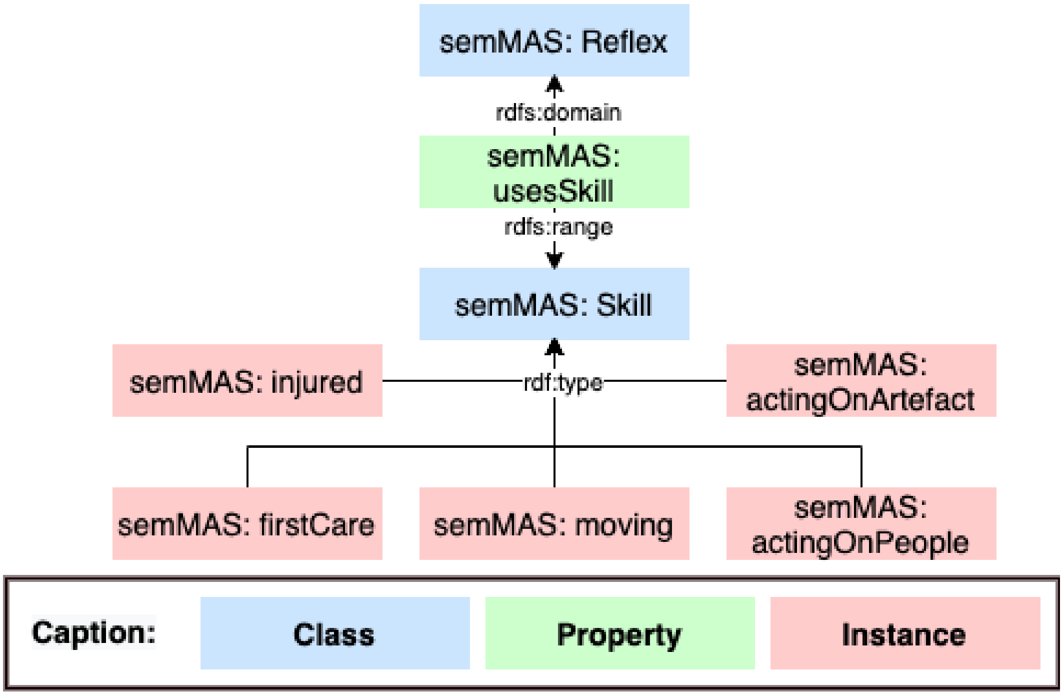


Figure 5.21: Skill modeling

Reflex specification The concept *semMAS:Reflex* is defined as equivalent to the concept *semMAS:Agent_action* through the property *owl:EquivalentClass*. As explained previously, each implemented reflex is linked to skills through the property *semMAS:usesSkill* (c.f. Figure 5.21). Let us take the example of the concept *semMAS:Transport_Aa*, which is a subclass of *semMAS:Reflex* and *semMAS:Agent_action* as illustrated previously in Figure 5.18. This concept represents the functionnality *Transport* of the new implemented skill *semMAS:actingOnPeople* as illustrated previously in Figure 5.5. It also uses the skill *semMAS:moving* as explained in Figure 5.4. Therefore, this reflex is based on the skills *semMAS:actingOnPeople* and *semMAS:moving*. Code 5.2 presents the OWL restriction of the concept *semMAS:Transport_Aa* through the Manchester syntax.

```
Transport_Aa:
 usesSkill value (moving and actingOnPeople)
```

Code 5.2: Specification of the concept *semMAS:Transport_Aa* through the Manchester syntax

5.4 Client

The client, corresponding to the red architecture's component 1 in Figure 5.22, aims at modeling the disaster management system and scenarios to launch the plan assessment. It can be achieved through direct knowledge modeling or knowledge extraction from data and requires integrating three main elements: the disaster management's system description, the events description, and the geospatial description. Those modelings are based on the terminological box of the SemDM ontology presented in subsection 5.3.1 of the previous section. The terminological box provides a model to describe a specific disaster management system and specific scenarios through assertions in the SemDM ontology. Each of these descriptions is the base for the simulation modeling that allows plan assessment. These user inputs are provided through the client's interface presented in section 6.1.1 of the next chapter.

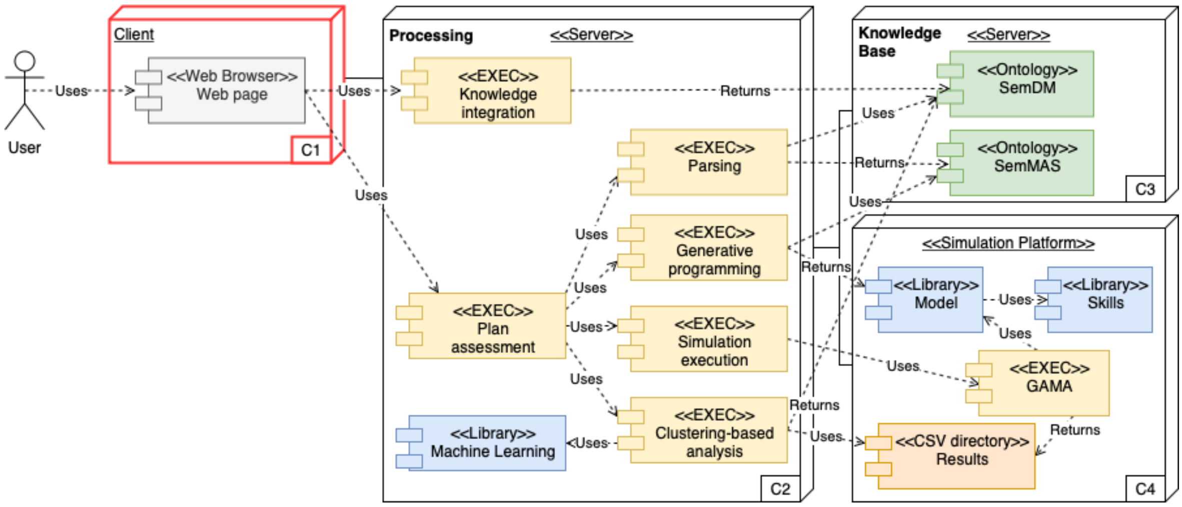


Figure 5.22: Component 1 of the architecture: the Client

5.4.1 Disaster management system's modeling

The disaster management system's modeling requires assertions that follow the service and procedure representations. The process of plan assessment requires the representation of at least one plan with its associated service. The associated service corresponds to the service that follows the plan to assess. Three elements are essential to describe a service: the elements at risk that the service serves, the role that delivers the service, and the organizations with their resources in terms of role. These concepts are the base for simulation modeling. Indeed, the role description with their services and associated procedures are used to generate agents and their associated behavior. Similarly, the description of the procedure has a primary role in the design of agent behaviors. On the one hand, a protocol must be defined with at least one action. On the other hand, a plan must be defined with at least a task achieved by a service. These description requirements are recursive. A plan description requires thus, the description of its tasks, their associated services, and plan's sub-procedures with their tasks or actions. Resources associated with each task and action also play an essential role. Resources must be characterized at least through the property *semDM:inTermsOf*. They can have a proportion or a quantity, but if they do not have, they will be considered as variables of simulation, and their proportion will evolve according to the complete quantity if available, otherwise, with random quantities. The elements at risk served by a service that follows a plan are used to define the simulation's observable variables and to compute the plan's effectiveness rating.

5.4.2 Event modeling

A scenario's modeling requires assertions that follow the disaster event representation. An event must be described through at least a location and elements at risk that it impacts. The location of the event allows the definition of the governmental echelon of the disaster. The element at risk is used to determine the plan to assess according to the service that follows it and serves the element at risk. The different events described in the SemDM ontology are mainly used for configuring simulation experiments during the simulation modeling process.

5.4.3 Geospatial modeling

The geospatial modeling concerns firstly the modeling of roads and buildings in the governmental echelon of described events. Secondly, it concerns the location of the different elements intervening in the description of the disaster management system, such as organizations, resources, and elements at risk. The geospatial modeling is essential for the definition of the environment modeling, whose location modeling of its artifacts and agents during the simulation modeling process.

The integration of disaster management systems and events is managed by direct user modeling, whereas geospatial modeling is managed through knowledge extraction from data. The different modeling aspects are further illustrated in a use case in section 6.1 of the next chapter. Begin with the geospatial modeling is better to allow the association of integrated geospatial elements to the description of a disaster management system and scenario during the direct user modeling. The web interface allowing users to interact with the processing server for knowledge integration and to launch the plan assessment process is presented in section 6.1.1 of the next chapter.

5.5 Processing server

This section presents the processing server, which is the red architecture's component 2 illustrated in Figure 5.23. This component executes the different processes intervening in the thesis' method presented section 4.3.

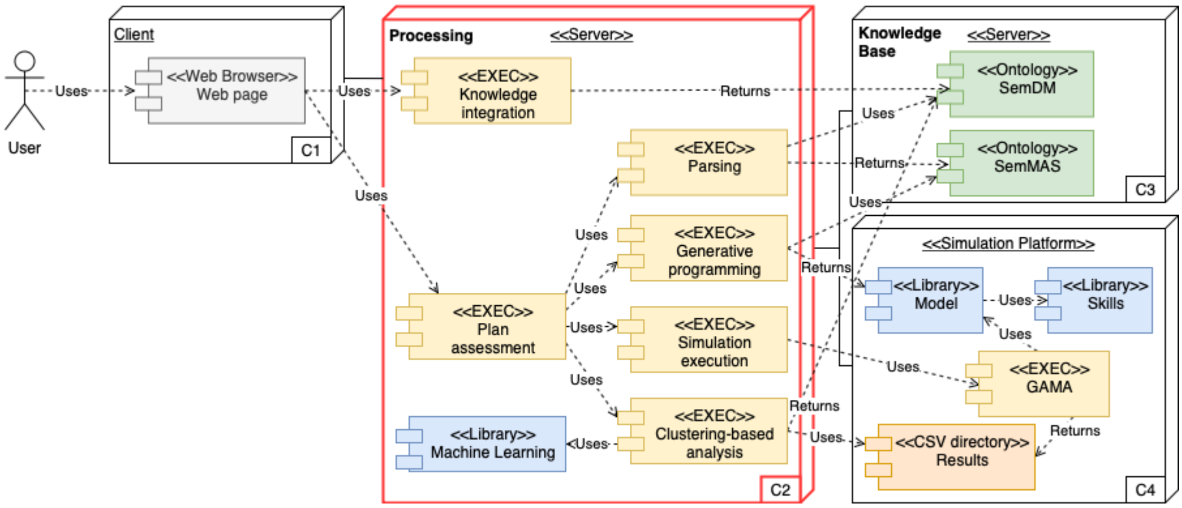


Figure 5.23: Component 2 of the architecture: Processing server

Figure 5.24 provides an overview of the sequence executed by the processing server related to the system components and related to the method's steps (presented in Figure 4.10). The processing server's processes are achieved in two primary steps: the disaster management system's modeling and the plan's assessment.

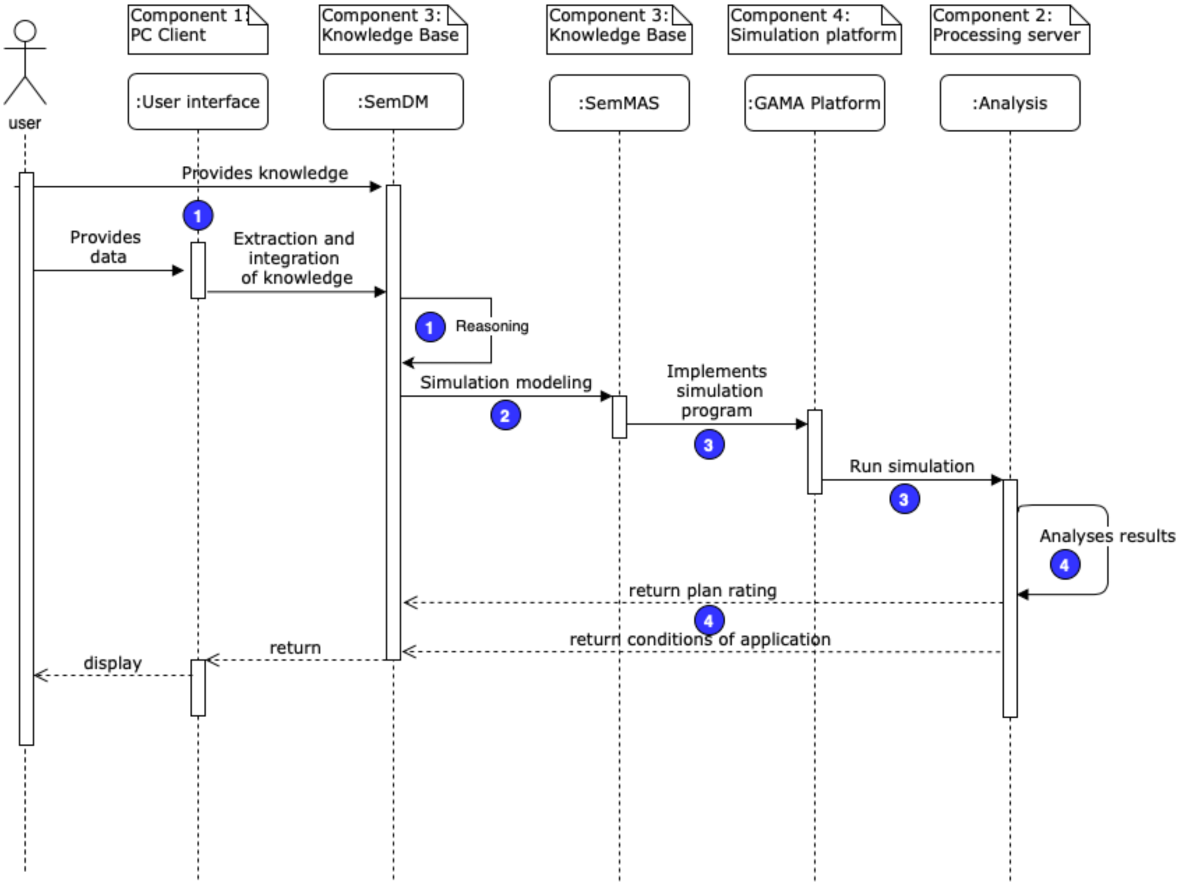


Figure 5.24: Overview of the proposed sequence

5.5.1 Disaster management system's modeling

The disaster management system's modeling corresponds to the first step of the method illustrated in Figure 4.10 in chapter 4. This main step is composed of the sub-steps illustrated in Figure 5.24 with the number 1. The first sub-steps correspond to disaster management knowledge integration. The knowledge integration can be done directly by integrating the user's definition of a disaster management model and by the knowledge extraction from data provided by the user. These first sub-steps of knowledge integration fulfill the SemDM ontology. The system modeling is then completed through rule-based reasoning on geospatial knowledge according to event knowledge into SemDM ontology. The implementation of these sub-steps is detailed in section 6.1 of the next chapter.

5.5.2 Plan assessment

Based on the disaster management system's modeling, the processing server can achieve the plan's assessment. Plan's assessment process corresponds to the steps

2, 3, and 4 of the method, both illustrated in Figure 4.10 in chapter 4 and in Figure 5.24. They correspond to simulation modeling, simulation design, and clustering-based analysis, respectively. The simulation design (step 3) aims at implementing and executing simulation experiments. Therefore the processing server achieves this step firstly by executing generative programming and secondly by running the simulation experiments. Plan's assessment process is thus composed of four processing sub-steps, which are the simulation modeling, the generative programming, the simulation execution, and the clustering-based analysis.

Simulation modeling Simulation modeling corresponds to the second step of the method. This process aims to create the conceptual simulation model first and then, the representation of the programmed model. The conceptual simulation model is a multi-agent model based on concepts that characterize the paradigm agent. The programmed model is also a multi-agent model, but specific to the GAMA platform, which has been chosen to execute simulation experiments. This model is defined through concepts specific to the GAML programming (language used by the GAMA platform). The simulation modeling process uses the disaster management system's model into SemDM ontology to generate the simulation models into the SemMAS ontology. It achieves simulation modeling through a succession of reasoning processes. The implementation of this processing's sub-steps is detailed in section 6.2 of the next chapter.

Generative programming Generative programming is the first sub-step of the simulation design (step 3 of the method). This process uses the representation of the programmed model in SemMAS ontology to generate the simulation program for the GAMA platform. Once created, it adds the simulation program to the simulation platform's model library to allow its execution. The implementation of the generative programming is detailed in section 6.3.1 of the next chapter.

Simulation execution The second sub-step of simulation design is simulation execution. The processing server executes the simulation experiments through the GAMA platform from the generated simulation program, stored in the library. The simulations execution produces results that are stored in a CSV directory in the simulation platform. The simulation execution process is detailed in section 6.3.2 of the next chapter.

Clustering-based analysis Finally, the last step of the method is the clustering-based analysis. The process of clustering-based analysis computes the effectiveness rating of plans by using simulation results stored in the CSV directory. It then applies the first clustering on the effectiveness score to identify the applicability of a plan. In the case of specific applicability, it applies a second clustering to dissociate the different situations impacting the plan's effectiveness. This process uses the CURE unsupervised clustering approach [Guha et al., 2001] (c.f. §Clustering approaches by unsupervised learning of section 3.1.2.4 in chapter 3) to achieve and configure clustering processes. The plan's assessment results are then represented through an RDF graph and added to the knowledge base to enrich the SemDM ontology through an automatic process. The sub-steps implementation of the clustering-based analysis is detailed in section 6.4 of the next chapter.

5.6 Summary

This chapter has presented the architecture on which the method's implementation is based. The method processes are achieved by the processing server, which uses the knowledge base and the simulation platform. The knowledge base contains the ontologies SemDM and SemMAS corresponding to approaches A2 and A6 presented in the previous chapter. The SemDM ontology aims at representing disaster management knowledge for plan assessment. This knowledge domain gathers vocabulary to represent geospatial objects, disaster management activities, and plan's effectiveness. The SemMAS ontology aims at representing the multi-agent simulation knowledge domain. It gathers vocabulary to represent a conceptual MAS model and a programed simulation model for the GAMA platform. The simulation platform aims at executing the simulation experiments. It executes the programmed simulation model. This programmed model is represented in the SemMAS ontology and is based on the newly developed skill plugin for disaster management, corresponding to the approach A5 presented in the previous chapter. The results of simulation experiments are stored in CSV files that are then analyzed by the processing server. A user interface allows the integration of disaster management knowledge in the SemDM ontology and the plan's assessment launching. The processing server requested by the user interface performs the different processes belonging to the method proposed in this thesis. The next chapter details the implementation of the method's processes corresponding to the approaches of automatic integration (A7), knowledge-driven simulation modeling (A3), knowledge-driven simulation programming (A4), and clustering-based analysis (A1) presented in the previous chapter.

6 Method's implementation

Contents

6.1 Disaster management system modeling 162
6.1.1 User interface 163
6.1.2 Extraction and integration of knowledge from data 166
6.1.3 Rule-based reasoning on the ontology SemDM 171
6.2 Simulation modeling 173
6.2.1 Multi-agent simulation modeling from a disaster management model 175
6.2.2 Modeling of the programmed simulation model 184
6.3 Simulation design 187
6.3.1 Generative programming 187
6.3.2 Simulation execution 192
6.4 Clustering-based analysis 194
6.4.1 Effectiveness score-based segmentation 195
6.4.2 Instance case segmentation 196
6.4.3 Knowledge enrichment 197
6.5 Summary 201

This chapter aims to present the implementation method based on the architecture presented in the previous chapter 4. This architecture comprises a user interface, a processing server, a knowledge base, and a simulation platform. The user interface allows users to interact with the processing server, prepare the processing, and request the plan assessment. The processing server manages and applies the processing intervening in the method presented in chapter 3. The method is composed of four main steps. The first step consists of modeling the studied disaster management system through knowledge integration and rule-based reasoning on SemDM ontology. The second step corresponds to the simulation modeling through a conceptual model and a programmed model representation specific to GAMA platform into SemMAS ontology. The results of modeling processes are stored into the knowledge base, which is used to guide the simulation design. The third step is the simulation design through an automatic process of generative programming and the simulation experiments' execution. Simulation experiments are executed through the GAMA simulation platform. Finally, the last step consists of enriching the ontology SemDM with the plan's assessment obtained from a clustering-based analysis of the simulation experiment results.

This chapter explains the different steps of the method achieved by the processing server. Section 6.1 presents the studied system modeling achieved by the integration of knowledge extracted from data and the rule-based reasoning on the ontology SemDM. Then, simulation modeling is explained in section 6.2. It presents the processes to generate the conceptual simulation model from the SemDM ontology and the programmed simulation model's representation for the GAMA platform. Section 6.3 presents the simulation design by explaining the implementation of the generative programming process and the simulation execution. Finally, section 6.4 explains the process of plan's assessment using the clustering-based analysis and the enrichment of the ontology SemDM with the plan's effectiveness representation.

6.1 Disaster management system modeling

A disaster management system modeling for plan assessment requires modeling at least one plan and one scenario. It also requires the geospatial information related to the plan and the locality of the scenario. The scenario consists mainly of describing the disaster event and, eventually, knowledge related to a plan specific to this disaster event. This knowledge is either extracted from data or directly provided by a user through the user interface presented in section 6.1.1. The knowledge modeling of disaster management is the first processing required for the plan assessment. As illustrated in Figure 6.1, it constitutes the first step of the processing timeline.

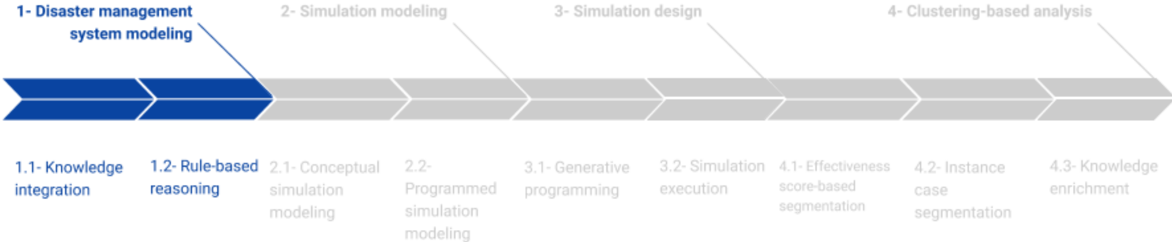


Figure 6.1: Disaster management modeling in processing timeline.

6.1.1 User interface

The user interface aims firstly at allowing users to achieve the disaster management system's modeling. It is achieved through knowledge modeling or knowledge extraction from data. Secondly, it aims to enable the user to visualize the knowledge base content and launch the plan's assessment.

User interface for knowledge modeling related to plan and scenario The user must provide event modeling and specific knowledge related to the application of a plan. The system is addressed to the disaster management community, not experts in computer sciences, whose semantic modeling. Therefore, the user interface allows adding an event description and other knowledge related to applying a plan ergonomically. The usability of this interface has been developed to enable non-computer scientists to represent disaster management knowledge. Knowledge modeled by the user is integrated into a copy of the ontology to check its consistency. If the consistency is validated, the knowledge base is updated by user modeling. Otherwise, the user is notified about modeling problems to make the necessary change in its modeling. Figure 6.2 illustrates interfaces providing the modeling functionalities.

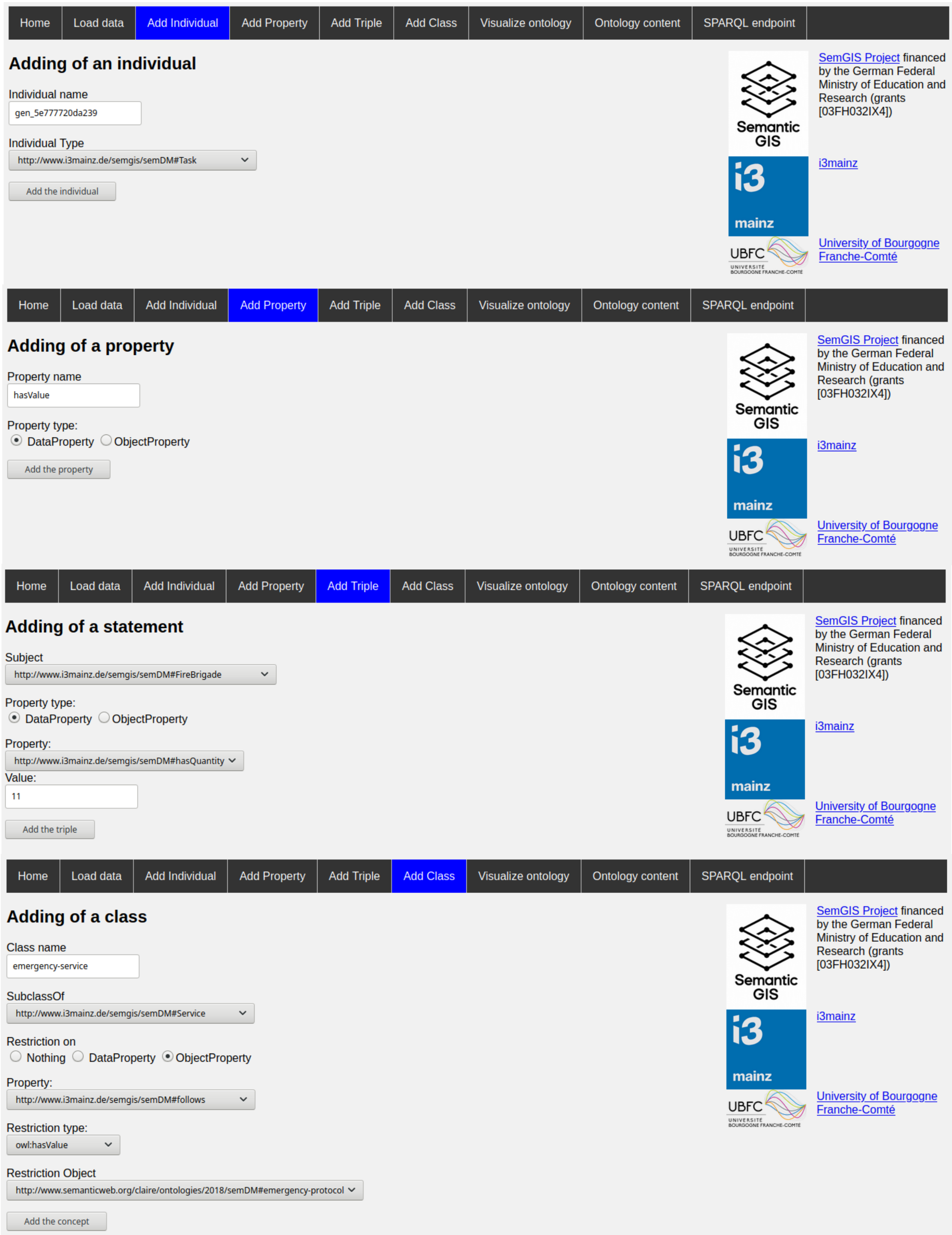


Figure 6.2: Screenshot of the user interface for the knowledge modeling.

User interface for knowledge extraction from data and integration into SemDM ontology The disaster management community has many data that contain knowledge related to their plans. Therefore, the second way to represent disaster management knowledge is to extract knowledge from data to enrich the knowledge base. This functionality, described in section 6.1.2, is realized by the processing server but requires the user's request. The functionality of extraction and integration of knowledge from data is provided by the user interface shown in Figure 6.3.

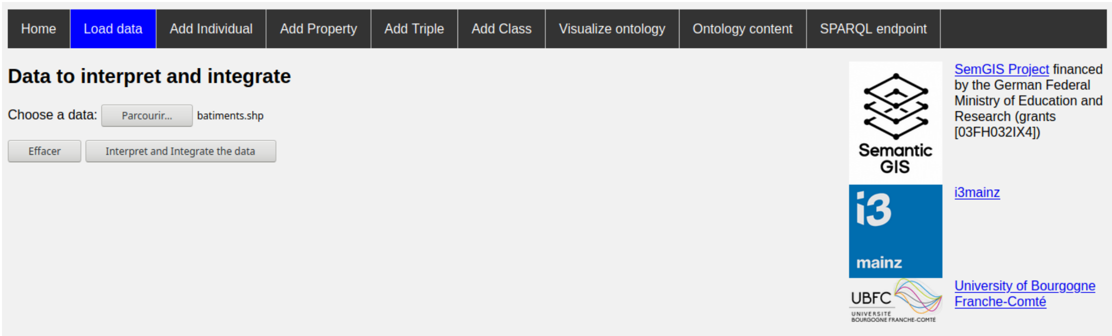


Figure 6.3: Screenshot of the user interface for the integration of knowledge extracted from data.

User interface for executing plan assessment process Once users have modeled knowledge and provided data to extract knowledge, they can verify the knowledge base's content. Figure 6.4 shows the interface allowing the visualization of the knowledge base's content. This interface uses WebVOWL 1.1.7[1]. They can then execute the assessment of plans by requesting the plan's assessment to the processing server. Simulation results are returned and displayed as a table in the web interface. Plan's assessment results are added to the knowledge base and can thus, be consulted through the interface, illustrated in Figure 6.4.

[1]WebVOWL website: `http://vowl.visualdataweb.org/webvowl.html`

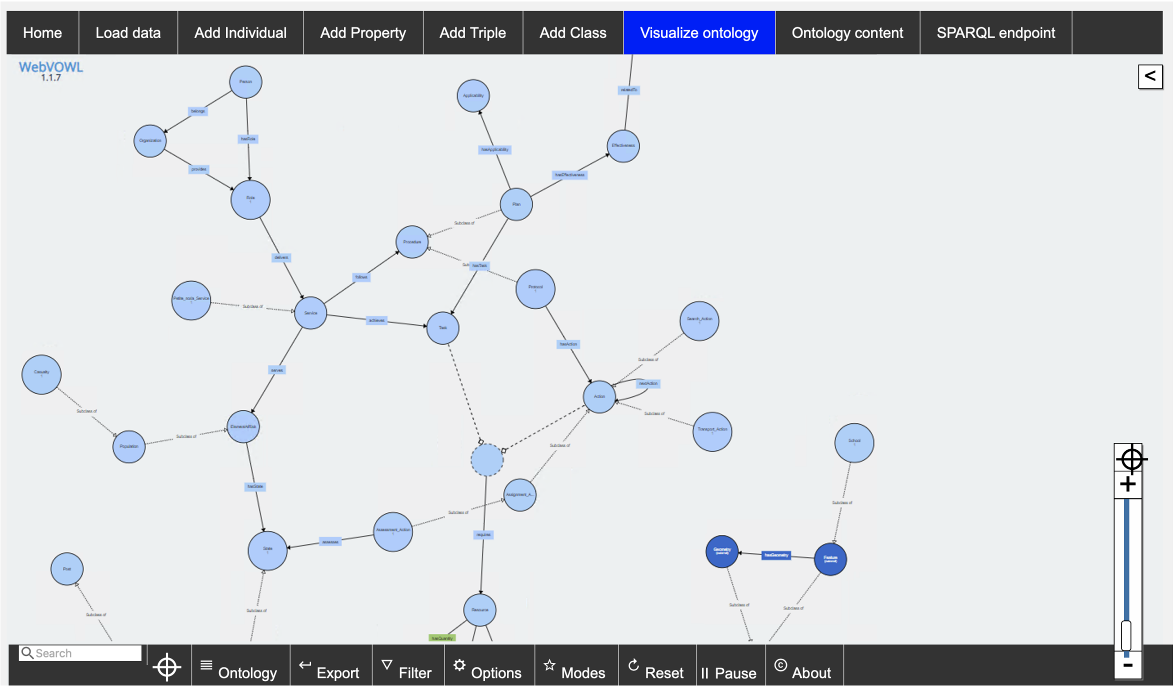


Figure 6.4: User interface to visualize knowledge base's content using WebVOWL 1.1.7.

6.1.2 Extraction and integration of knowledge from data

This section presents the approach for interpreting heterogeneous geospatial data semantically. This approach aims at structuring data to add it a meaning through a RDF graph representation. This approach has been developed and experimented in the context of the SemGIS project[2]. The experiments have been applied to interpret data with DBpedia and Wikidata concepts. Among these two experiments, the interpretation through Wikidata concepts has obtained more interesting results [Prudhomme et al., 2017b]. Therefore, this thesis applies the approach implementation using Wikidata concepts and extends it to integrate knowledge extracted from data into SemDM ontology. The process of semantic data interpretation is also called uplift (c.f. section 3.3.2). Heterogeneous data is mainly geodata with a table structure containing a collection of objects with a description of their geometry and other features related to these objects. Therefore, the integration process focuses mainly on the process of Shapefiles [Environmental Systems Research Institute (ESRI), 1998] and geospatial databases for geospatial data. This approach uses natural language processing, Semantic Web technologies and geospatial content to match information from a data with a concept or an individual of an ontology. Then, it creates an RDF graph with the

[2]SemanticGIS project, financed by the German ministry of education and research (grant: 03FH032IX4) and financing this thesis

chosen vocabulary to represent the data content. It uses schema mapping, ontology matching, and ontology mapping techniques:

- the implicit schema mapping between the table structure of the data and its representation through an RDF graph allows the data interpretation.
- the ontology matching technique enables the enrichment of information inside the data set by knowledge from Wikidata,
- the ontology mapping technique is used to integrate the RDF graph representing the data set into the SemDM ontology of the knowledge base.

These techniques guide the process of data interpretation and integration. This process makes intervene three sub-processing explained in appendix B.1, which are the geometry analysis (illustrated as P1 in Figure 6.5b and explained in appendix B.1.1), the feature value analysis, which corresponds to the analysis of the cell content of the tabular structure (illustrated as P2 in Figure 6.5b and explained appendix B.1.2), and the feature descriptor analysis, which corresponds to the analysis of columns name of the tabular structure (illustrated as P3 in Figure 6.5b and explained appendix B.1.3).

This approach aims to determine concepts from Wikidata to represent the data set components and allow its enrichment. The schema mapping is defined between a relational data set and the Web Ontology Language [Antoniou and Van Harmelen, 2004][3] by creating *owl:Class*, *owl:Individual*, *owl:ObjectProperty* and *owl:DataProperty*. This schema mapping is obtained by an analysis of the data set detailed in the next paragraph. The ontology matching is applied to discover a URI from Wikidata, representing the data set's content. The process of data interpretation aims at identifying potential concepts through the schema matching, searching for identified concepts in Wikidata through the ontology matching, and creating the appropriate representation according to the ontology matching result. The combination of schema mapping and ontology matching produces a local ontology linked to concepts in Wikidata, which is populated by the content of the data set (see Figure 6.5c). The population (set of individuals and their properties) from the data set content is annotated with their provenance information. It allows following updates of information and linking them without ignoring their provenance information to provide the possibility to recreate or

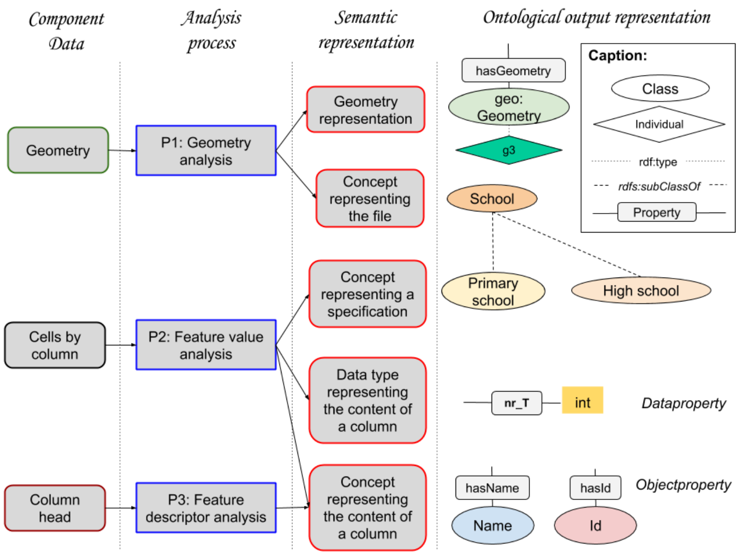
[3]compatible with OWL2 ontologies: "More importantly, backwards compatibility with OWL 1 is, to all intents and purposes, complete: all OWL 1 Ontologies remain valid OWL 2 Ontologies, with identical inferences in all practical cases" (W3C Recommendation, `https://www.w3.org/TR/2012/REC-owl2-overview-20121211/`, *Relationship to OWL 1*)

extend the data set. Finally, the population representing the data set is integrated into the knowledge base thanks to an ontology mapping between the SemDM ontology and Wikidata ontologies. These three main steps are illustrated through an example in Figure 6.5.

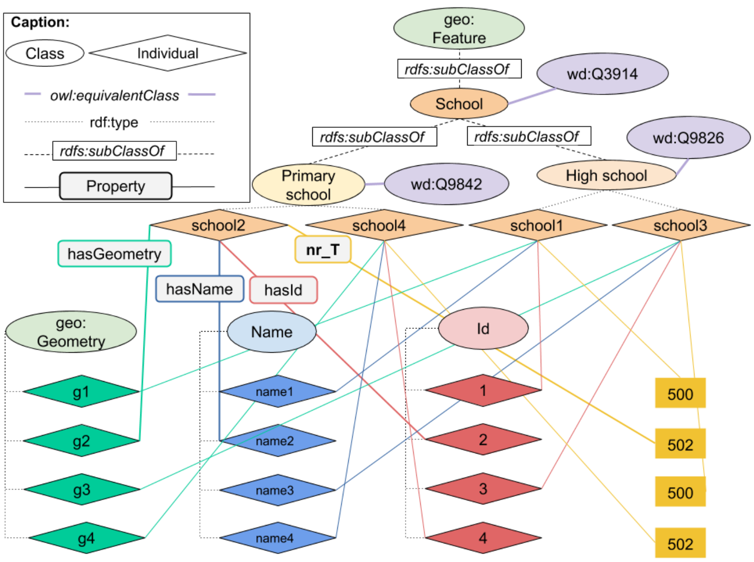
School.shp / School.dbf

Geometry	Id	Name	School_type	nr_T
g1	1	name1	Highschool	500
g2	2	name2	Primary school	502
g3	3	name3	Highschool	500
g4	4	name4	Primary school	502

(a) Example of input data: a Shapefile of schools



(b) Schema matching example according to the result of the steps of the ontology matching



(c) Output example with Wikidata matching

Figure 6.5: Methodology overview applied to an example [Prudhomme et al., 2017b]

Schema Mapping The schema defines a concept (*owl:Class*) to represent the type of data set. Besides, it defines an instance for each row of the data set and its geometry. Each column represents a piece of related information of an instance. Therefore it is represented by a *rdf:Property*. However, the column's specification as *owl:DataProperty* or *owl:ObjectProperty* depends on the ontology matching result performed during the feature value and the descriptor analysis. A specification of the general concept representing the data set can be identified by combining the feature value analysis, which identifies reoccurring values and the feature descriptor analysis, determining a column name classification.

Ontology matching The used ontology matching is a hybrid matching using String-based and instance-based techniques. The string-based technique uses the Google translate API[4] to determine the language of the data set. Next, this language is used in the process of string similarity matching with the label of the same language of a chosen ontology from Wikidata. The string similarity is computed through the Levenshtein measure [Levenshtein, 1966]. This technique is mainly applied to column heads (see Section B.1.3). In the case of a non-matched concept and a non-English label, the method translates the column head into English and repeats the process. A column head may also be a compound noun that must first be split to match with Wikidata. The instance-based technique introduced in appendix B.1.2 is also applied by an analysis of the cells' content to determine a potential existing matching individual from the Wikidata. After their identification, their classes are retrieved. The most frequently occurring class is then assigned as the concept of the column.

Combination for designing the RDF graph Executing the different steps of analysis provides four sets of concepts, which are used to build the resulting local ontology as follows:

1. Geometry Detection Set: If the geometry detection process detects a class, this class (or the highest-ranked class) will be considered to represent the data set.
2. File Name Detection Set: If the geometry detection process detects no appropriate geometry class, but the analysis of the filename identifies a class, then this class will represent the data set.
3. Property Detection Set: Properties and their respective ranges as detected by the Feature Descriptor Analysis are created and used for address columns as

[4]http://code.google.com/p/google-api-translate-java

determined by its respective analysis.

4. Individual Detection Set: Individuals are created according to the recognized classes and values that can be resolved to URIs will be produced as the corresponding individuals.

Thanks to the schema matching, the data set is considered as a derived graph. Through this derived graph and the concept matching, our process dynamically creates the different identified concepts, their individuals, the object properties related to them, and the column's data properties without concept, to ultimately link them together according to our derived graph.

Ontologies mapping for integrating the RDF graph into SemDM ontology The concepts defined in SemDM ontology are interlinked to concepts from Wikidata. Let us take as example, the concept *semDM:School*, which is a subclass of *semDM:ElementAtRisk* and *semDM:Building*. This concept is defined as an equivalent class to *https://www.wikidata.org/wiki/Q3914*. Let us then take the example of the integration of an individual as *semDM:school1* interpreted as an individual of the class *https://www.wikidata.org/wiki/Q9826*, which is equivalent to the concept *semDM:HighSchool* created by the previous process and defined in Wikidata as a subclass of *https://www.wikidata.org/wiki/Q3914*. The integration of the RDF graph containing this individual linked to its classes is then followed by an axiomatic-based reasoning step, enabling this individual to be connected to the concept *semDM:School*. Indeed, thanks to the reasoning and concepts interlinked, the concept *semDM:HighSchool* is automatically defined as a subclass of *semDM:School*, and thus, *semDM:school1* is automatically defined as an individual of *semDM:School*. This process is illustrated in a use case in section 7.2.1.1 of the next chapter.

6.1.3 Rule-based reasoning on the ontology SemDM

As explained previously, the user has only to provide knowledge related to plans, description of a disaster through events, and data specific to the locality. Then, rule-based reasoning is applied to these elements to deduce geospatial relationships. Two rules have been defined for this purpose. These rules are expressed in SHACL (c.f. the structure explanation of such rule in appendix B.2), and prefixes used in these rules are defined in Appendix B.3.

The first rule, whose central part of the SHACL rule is illustrated in Code 6.1, aims at inferring the relation $semDM : isIn$ between the different features according to their geometry.

```
CONSTRUCT { ?x semDM:isIn ?f.}
WHERE {
 ?x geo:hasGeometry ?g1 .
 ?f rdf:type geo:Feature.
 ?f geo:hasGeometry ?g2 .
 FILTER(geof:sfWithin(?g1, ?g2))
}
```

Code 6.1: Main part of the SHACL rule for determining the relation *semDM:isIn* between geospatial features

The second rule, whose the central part of the SHACL rule is illustrated in Code 6.2, aims at inferring the governmental echelon of a disaster, by using the relation $semDM : isIn$ between geospatial features. The governmental echelon of a disaster corresponds to the "smallest" governmental echelon, i.e., the governmental echelon in which is located the disaster event and which does not contain another governmental echelon containing the disaster event.

```
CONSTRUCT {?d semDm:hasGovernmentalEchelon ?ge.}
WHERE {
 ?d rdf:type semDM:Disaster.
 ?d semDM:hasEvent ?e.
 ?e rdf:type semDM:Event.
 ?e semDM:locatedIn ?f .
 ?f rdf:type geo:Feature.
 ?f semDM:isIn ?ge .
 FILTER NOT EXISTS(?x semDM:isIn ?ge && ?f semDM:isIn ?x && ?x rdf:
    type semDM:GovernmentalEchelon.)
}
```

Code 6.2: Main part of the SHACL rule for determining the governmental echelon of an event

This rule-based reasoning applied to the case study is presented in section 7.2.2. Rule-based reasoning, which is an inference of close world assumptions (c.f. appendix A.1.2), has the advantage of providing complex inference. However, it also has the risk to be an undecidable problem when combined with axiomatics and to conduct to the inconsistency of the knowledge base (c.f. section 4.2.2.1). The SHACL rule-based system used to apply this reasoning is the system provided by the API Jena [5]. The reasoning is applied without axiomatics by deactivating the rule-based system's entailment to overcome the risk of an undecidable problem. Each rule-based reasoning made by the system is applied to an ontology copy to

[5] Apache Jena Shacl

verify its consistency before updating the knowledge base. It aims at overcoming the risk of inconsistency. In addition to guarantee the knowledge base consistency, the consistency checking allows the verification of the proper ontology specification and the good definition of rules. At the end of the rule-based reasoning, whose consistency of its inference has been validated, the knowledge base contains the geospatial relationship *semDM:isIn* between the different concerned features of the SemDM ontology and the governmental echelon associated with the disaster.

6.2 Simulation modeling

The automatic simulation conceptualization process uses knowledge about disaster management in the SemDM ontology to fulfill the multi-agent-based simulation model into the SemMAS ontology. It corresponds to the second step of the processing timeline illustrated in Figure 6.6.

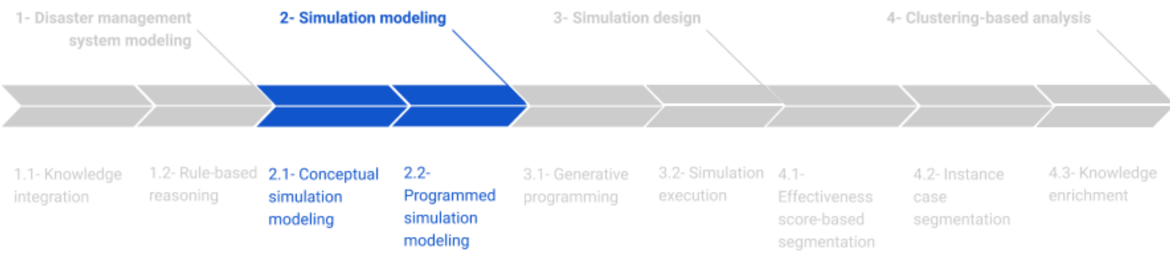


Figure 6.6: Multi-agent simulation modeling in processing timeline

This step aims to create the conceptual model in the agent paradigm corresponding to the disaster management model. Secondly, it aims at creating a GAML model to program the conceptual model and execute it through the GAMA platform. Figure 6.7 illustrates the overview of simulation modeling process.

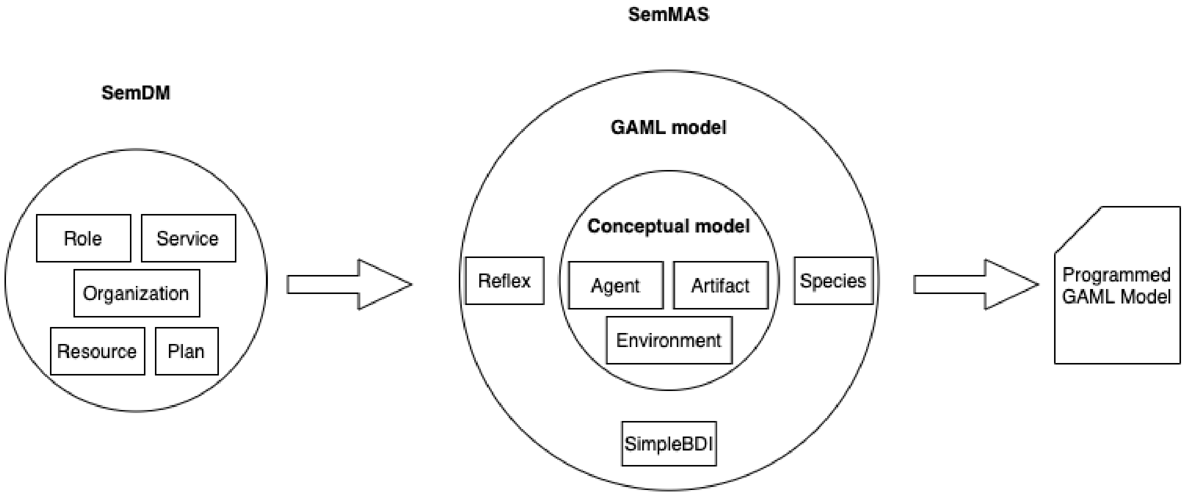


Figure 6.7: Overview of simulation modeling process

Built-ins for rule-based reasoning These two processes use rule-based reasoning. The used rules require to create new instances in the SemMAS ontology. Such a creation of instances is based on URI generation. Therefore, new rule built-ins have been designed for allowing the URI generation with different parameters. They have *semTransform* as namespace. SHACL rule built-ins being based on SPARQL, imply the implementation of new SPARQL functions as URI generators. Three different built-ins have been created to generate URI. The most simple $semTransform : generateURI(prefix, string)$ generates a URI with the prefix given as parameter and the string as a base to define the URI. The second one $semTransform : lowerCaseURI(prefix, concept)$ generates a URI with the prefix given as a parameter and an existing concept. Finally, the built-in $semTransform : createURI(prefix, individual, individual\_type, new\_type)$ generates a URI from a URI of another individual. It takes a prefix for the new individual, an individual, the type of the individual, and the type of the new individual. The process of conceptual simulation modeling also requires the generation of the geospatial file. Therefore, a built-in called $semTransform : geoCreate(feature, governmental\_echelon)$ has been created. It takes two parameters as input: a concept representing the geospatial feature to extract for creating the geospatial file and the governmental echelon addressed by the simulation model to retrieve only the defined features in the defined governmental echelon. Identifying the model's parameters through list definition requires identifying the disaster related to individuals having a list and its associated simulation model. Therefore a rule's built-in, called $semTransform : searchModel(individual)$ has been created. It takes an individual, as built-in's input, and returns the model associated with the disaster linked to the input individual. Further, rule built-ins have been developed for another purpose, which is the enrichment with knowledge of the plan's effectiveness, presented later in section 6.4 of this chapter. These other built-ins have *semStatistics* as namespace and are used to compute statistics from a set of values provided as inputs. This set of built-ins allows computing average, median, and mean square values through $semStatistics : GetAverage()$, $semStatistics : GetMedian()$, and $semStatistics : GetMeanSquare()$, respectively.

6.2.1 Multi-agent simulation modeling from a disaster management model

The creation of the conceptual model from the disaster management model is achieved through rule-based reasoning. Similarly to the previously explained rule-based reasoning, it is based on SHACL [Knublauch and Kontokostas, 2017] rules and benefits from a consistency checking before updating the knowledge base. This section describes rules, which allow the dynamic modeling of simulation components according to disaster management knowledge. This dynamic modeling inits at first a simulation model, then the environment and its artifacts that compose the model, and finally distinguishes the agents from the other artifacts. Among the different stakeholders, some are represented through cognitive BDI agents, and others are represented as reactive agents. In disaster management, stakeholders are characterized by an instance of *semDM:Role* that delivers instances of *semDM:Service* (through the property *semDM:delivers*). An instance of *semDM:Service* follows an instance of *semDM:Procedure* (through the property *semDM:follows*). The services delivered by a role allows defining the agent type. A role providing services following only plans (instances of *semDM:Plan*) is represented through a cognitive BDI agent (i.e., an instance of *semMAS:BDI_agent*), which can be a manager (i.e., an instance of *semMAS:Manager_agent*) or a central agent (i.e., an instance of *semMAS:Central_agent*). On the contrary, a role providing services following only protocols (i.e., instances of *semDM:Protocol*) is represented as an actor (i.e., an instance of *semMAS:Actor_agent*), a reactive agent.

Conceptual simulation modeling The conceptual simulation model represents the disaster management system to respond to a disaster. Therefore, it is characterized by the property *semDM:adaptedWith* an individual of *semDM:Disaster*. It has an environment that represents the governmental echelon of its associated disaster. The definition of a conceptual simulation model is achieved through the SHACL rule, whose central part is presented in Code 6.3.

```
CONSTRUCT {
?m rdf:type semMAS:DM_model.
?e rdf:type semMAS:Environment.
?m semMAS:adaptedWith ?d.
?m semMAS:hasEnvironment ?e.
?e semMAS:represents ?b.
}
WHERE {
?d rdf:type semDM:Disaster.
?d semDM:hasGovernmentalEchelon ?ge.
?m semTransform:generateURI(semMAS, "model").
?e semTransform:generateURI(semMAS, "env").
}
```

Code 6.3: Main part of the SHACL rule for creating the conceptual simulation model

A simulation model also has parameters. There are two types of parameters. Those define through a list of values, and the resources define without values. Parameters representing a list of values are identified through the property *semDM:hasList* that provides a *rdf:List* with values. The rule's built-in *semTransform:searchModel(individual)* presented previously is used to determine the simulation model associated with the individual having a list to link it as a parameter of this model. The definition of this type of parameter is achieved through the SHACL rule, whose central part is presented in Code 6.4.

```
CONSTRUCT {
?m semMAS:hasParameter ?l.
}
WHERE {
?x semDM:hasList ?l.
?l rdf:type rdf:List.
?m semTransform:searchModel(?x).
}
```

Code 6.4: Main part of the SHACL rule for defining the conceptual simulation model's parameters representing a list of value

In the case of resources defined without associated value, i.e., without value associated through the properties *semDM:hasQuantity* and *semDM:hasProportion*, a parameter representing through an interval is created to explore the different possible proportions of a resource. The definition of this type of parameter is achieved through the SHACL rule, whose central part is presented in Code 6.5.

```
CONSTRUCT {
?m semMAS:hasParameter ?p.
```

```
?p rdf:type semMAS:Interval.
?p semMAS:hasMin "0".
?p semMAS:hasMax "100".
?p semMAS:hasStep "10".
?p semMAS:definesProportion ?r.
}
WHERE {
?r rdf:type semDM:Resource.
FILTER NOT EXISTS (?r semDM:hasQuantity ?q).
FILTER NOT EXISTS (?r semDM:hasProportion ?x).
?m semTransform:searchModel(?r).
?p semTransform:generateURI(semMAS, "parameter").
}
```

Code 6.5: Main part of the SHACL rule for defining the conceptual simulation model's parameters represented through an interval

Environment Modeling The scale of the simulation corresponds to the governmental echelon that manages the disaster situation. As presented through the disaster event modeling in section 5.3.1 of the previous chapter and the rule-based reasoning applied on it presented in section 6.1.3, the individual of *semDM:GovernmentalEchelon* associated to the instance of the class *semDM:Disaster* depends on the location of events that compose the disaster (i.e. instances of *semDM:Event* that are linked to the instance of *semDM:Disaster* through the property *semDM:hasEvent*). This disaster provides information about the elements at risk that can be impacted by this disaster and the damage they can undergo. The governmental echelon of a simulation model represents the space of an environment. Thanks to the property *semDM:isIn* linking the different geospatial elements, the environment's components can be determined from the geospatial elements that are linked to the instance of *semDM:GovernmentalEchelon* through *semDM:isIn*. These components are defined as a *semMAS:Artifact* of the environment (i.e. instance of *semMAS:Environment*). The main artifacts composing the environment are instances of *semDM:Road* and *semDM:Building*. The modeling of such artifacts is generated from SHACL rule, whose the example of the central part for buildings is presented in Code 6.6. This rule creates an instance of *semMAS:Artifact* representing the set of buildings, linked to instances of *semDM:Building* through the property *semMAS:hasMembership*. It also associates a file path (created from the built-in *semTransform:geoCreate()*) to this instance through the property *semMAS:hasfile*. In addition to return a file path, the built-in *semTransform:geoCreate()* creates a geospatial data with the returned file path from instances of the class given as first parameter, which are in the governmental echelon given as second parameter. A similar rule exists for creating an instance of *semMAS:Artifact* representing the set of roads.

```
CONSTRUCT {
?b rdf:type semMAS:Artifact.
?b semMAS:hasfile ?f.
?b semMAS:hasMembership ?bdm.
?e semMAS:contains ?b.
}
WHERE {
?e rdf:type semMAS:Environment.
?e semMAS:represents ?ge.
?ge rdf:type semDM:GovernmentalEchelon.
?bdm rdf:type semDM:Building .
?b semTransform:lowerCaseURI(semMAS, semDM:Building).
?f semTransform:geoCreate(semDM:Building, ?ge).
}
```

Code 6.6: Main part of the SHACL rule for creating environment's buildings

Reactive Agents Modeling A set of SHACL rules with different priorities (c.f. the structure explanation of a SHACL rule in appendix B.2) achieves the modeling of the multi-agent simulation. Let us take the example of reactive actor agent modeling to illustrate concepts, facts, rules, and their priority intervening in the process of transformation. The priority of a rule gives an order of execution. Therefore, the SHACL rules allowing the creation of agents have the highest priority. There are different rules to create agents according to the type of agent. The central part of the SHACL rule allowing creating reactive actor agents with their behaviors, is presented in Code 6.7.

```
CONSTRUCT {
?a rdf:type semMAS:Actor_agent.
?a semMAS:represents ?r.
?a semMAS:number ?q.
?b rdf:type semMAS:Behavior.
?b semMAS:represents ?s.
?a semMAS:hasBehavior ?b.
?env semMAS:contains ?a.
}
WHERE {
?r rdf:type semDM:Role.
?o semDM:provides ?r.
?o semDM:hasResource ?res.
?res semDM:inTermsOf ?r.
?res semDM:hasQuantity ?q.
```

```
?a semTransform:createURI(semMAS, ?r, semDM:Role, semMAS:Agent).
?r semDM:delivers ?s.
?s semDM:follows ?p.
?p rdf:type semDM:Protocol.
?b semTransform:createURI(semMAS, ?s, semDM:Service, semMAS:Behavior).
?s semDM:serves ?er.
?ev semDM:impacts ?er.
?d semDM:hasEvent ?ev.
?m semDM:adaptedTo ?d.
?m semDM:hasEnvironment ?env.
FILTER NOT EXISTS (?p rdf:type semDM:Plan).
}
```

Code 6.7: Main part of the SHACL rule for creating actor agents

SHACL rules creating actions of behaviors have the second priority. These rules depend on the types of actions that compose a protocol. The characteristics of an action depend on its type. Let us illustrate this type of rule for the transport actions, which have the type *semDM:Transport_Action* in the SemDM ontology and will be represented by an agent's action of the type *semMAS:Transport_Aa* (c.f. section 5.3.2.2) in the SemMAS ontology. Code 6.8 presents the main part of the SHACL rule for creating a transport action of agents.

```
CONSTRUCT {
?t rdf:type semMAS:Transport_Aa.
?b semMAS:hasAction ?t;
?t semMAS:what ?x.
?t semMAS:requiresObjective ?obj.
?t semMAS:where ?d.
?t semMAS:represents ?act.
}
WHERE {
?a rdf:type semMAS:Actor_agent.
?a semMAS:hasBehavior ?b.
?b semMAS:represents ?s.
?s semDM:serves ?er.
?x semMAS:represents ?er.
?s semDM:hasDestination ?d.
?s semDM:follows ?p.
?p semDM:hasAction ?act.
?act rdf:type semDM:Transport_Action.
?t semTransform:createURI(semMAS, ?act, semDM:Transport_Action, semMAS:
    Transport_Aa).
?obj semTransform:generateString(?t, ?x);
}
```

Code 6.8: Main part of the SHACL rule for creating transport action of agents

In the ontology SemDM, a protocol is composed of actions that can have successive action. This relation is represented through the property *semMAS:hasNext* between two actions. In SemMAS ontology, an agent's behavior represents a protocol, and an agent's action represents a protocol's action. The sequence of an agent's actions is managed through a status that must be generated from the SemDM ontology. These status depend on if action is preceding or succeeding by another action. Therefore, a rule with a third priority has been added to manage the agent's action succession through status according to a protocol definition. The central part of this rule is presented in Code 6.9. A new generated status links two agent's actions that represent two protocols actions linked by the property *semDM:hasNext*. The link between two agent's actions through a status is expressed with the properties *semMAS:results* and *semMAS:requiresStatus*.

```
CONSTRUCT {
?status rdf:type semMAS:Status.
?a1mas semMAS:results ?status.
?a2mas semMAS:requiresStatus ?status.
}
WHERE {
?a1mas rdf:type semMAS:Agent_action.
?a1mas semMAS:represents ?a1dm.
?a1dm semDM:hasNext ?a2dm.
?a2mas semMAS:represents ?a2dm.
?status semTransform:generateURI(semMAS, "status").
}
```

Code 6.9: Main part of the SHACL rule for creating required and resulting status of an agent's actions

Finally, two rules with a fourth priority are added to manage the cases of an action that has no previous action or no successive action. In the case of the action is not an object of the property *semDM:hasNext*, which means it has no previous action and retrieves status from the property *semDM:targetsStates* of its associated service. In the case of the action is not a subject of this property, that means it has no successive action and retrieves status from the property *semDM:producesStates* of its associated service. Let us illustrate the SHACL rule for no previous action through the central part presented in Code 6.10.

```
CONSTRUCT {
?a1mas semMAS:requireStatus ?status.
?status rdf:type semMAS:Status.
}
WHERE {
?a1mas rdf:type semMAS:Agent_action.
```

```
?a1mas semMAS:represents ?a1dm.
?p semDM:hasAction ?a1dm.
?s semDM:follows ?p.
?s semDM:targetsState ?status
FILTER NOT EXISTS (?a2dm semDM:hasNext ?a1dm).
}
```

Code 6.10: Main part of the SHACL rule for defining a required status in the case of no previous action

Cognitive Agents Modeling Stakeholders, whose role is to organize and manage tasks' achievement, are represented in this thesis through BDI agents. Let us take the example of these BDI agents, which are cognitive agents, to illustrate rules used for generating cognitive agents. These agents correspond to roles that deliver services that follow only plans. They are created according to the definition of roles, services, and plans in SemDM through a SHACL rule, whose central part is presented in 6.11 and which has the highest priority.

```
CONSTRUCT {
?a rdf:type semMAS:BDI_agent.
?a semMAS:represents ?r.
?a semMAS:number ?q.
?env semMAS:contains ?a.
}
WHERE {
?r rdf:type semDM:Role .
?r semDM:hasQuantity ?q .
?r semDM:delivers ?s .
?s semDM:follows ?p .
?p rdf:type semDM:Plan.
?a semTransform:createURI(semMAS, ?r, semDM:Role, semMAS:BDI_agent).
?s semDM:serves ?er.
?ev semDM:impacts ?er.
?d semDM:hasEvent ?ev.
?m semDM:adaptedTo ?d.
?m semDM:hasEnvironment ?env.
FILTER NOT EXISTS (?p rdf:type semDM:Protocol).
}
```

Code 6.11: Main part of the SHACL rule for creating cognitive BDI agents

A BDI agent has a set of beliefs, desires, and plans linked to an intention, which is a desire selected to be achieved. The SHACL rule, whose central part is presented in Code 6.12 and which has the second priority, presents the generation of desires and plans of a BDI agent according to SemDM concepts. Desires of a BDI agent are defined through the services that it can deliver. In disaster management, a plan followed by a service is composed of tasks achieved by other services delivered by other roles. The BDI agent, which aims at organizing the task's achievement of a disaster management plan, assigns the tasks to other agents through messages. A BDI agent's plan is triggered when the intention linked to the plan is the agent's current intention. This current intention is chosen among the agent's desires according to its knowledge about the situation. A desire corresponding to the will of providing a service becomes an intention when the situation requires this service.

```
CONSTRUCT {
?d rdf:type semMAS:Desire.
?d semMAS:represents ?s.
?a semMAS:hasDesire ?d.
?p rdf:type semMAS:BDI_Plan.
?p semMAS:represents ?dmp.
?a semMAS:hasPlan ?p.
?p semMAS:hasIntention ?d.
}
WHERE {
?a rdf:type semMAS:BDI_agent.
?a semMAS:represents ?r.
?r semDM:delivers ?s .
?d semTransform:createURI(semMAS, ?s, semDM:Service, semMAS:Desire).
?s semDM:follows ?dmp.
?p semTransform:createURI(semMAS, ?dmp, semDM:Plan, semMAS:BDI_Plan).
}
```

Code 6.12: Main part of the SHACL rule for creating desires and plans of BDI agents

The distinction between manager and central agents depends on the services delivered by the role they represent. A manager agent is characterized by services that achieve a task, whereas the central agent's services do not achieve a task because central agents are at the top of the coordination pyramid. Beliefs of BDI agents are used to define new desire, which corresponds to service in the SemDM ontology. Therefore, manager agents' beliefs correspond to tasks that are achieved by the services, which are represented by their desire. Code 6.13 present the central part of the SHACL rule, alowing the definition of manager agent's beliefs and their association to the type *semMAS:Manager_agent*.

```
CONSTRUCT {
?b rdf:type semMAS:Belief.
?b semMAS:represents ?t.
?a semMAS:hasBelief ?b.
?a rdf:type semMAS:Manager_agent.
```

```
}
WHERE {
?a rdf:type semMAS:BDI_agent.
?a semMAS:represents ?r.
?r semDM:delivers ?s .
?s semDM:achieves ?t .
?b semTransform:createURI(semMAS, ?t, semDM:Task, semMAS:Belief).
}
```

Code 6.13: Main part of the SHACL rule for creating beliefs of manager agents

In the case of a central agent, its beliefs cannot be defined according to the tasks achieved by their services as they do not have it. Therefore, the central agent's beliefs are defined through the element at risk served by its services. Code 6.14 present the central part of the SHACL rule, allowing the definition of the central agent's beliefs and their association to the type *semMAS:Central_agent*. These rules defining an agent's beliefs have a second priority as they require the generation of BDI agents.

```
CONSTRUCT {
?b rdf:type semMAS:Belief.
?b semMAS:represents ?er.
?a semMAS:hasBelief ?b.
?a rdf:type semMAS:Central_agent.
}
WHERE {
?a rdf:type semMAS:BDI_agent.
?a semMAS:represents ?r.
?r semDM:delivers ?s .
?s semDM:serves ?er.
?b semTransform:createURI(semMAS, ?er, semDM:ElementAtRisk, semMAS:Belief
    ).
FILTER NOT EXISTS (?s semDM:achieves ?t).
}
```

Code 6.14: Main part of the SHACL rule for creating beliefs of central agents

Finally, the rules with the lowest priority are rules that use the generated agents. Indeed, these rules require the application of the previous rule in order to use the defined agent. Among these rules, there is the rule to define agents known by a BDI agent, whose central part is illustrated in Code 6.15 or the rule to define properties of a BDI plan that defines a communication whose interlocutor is another agent among their known agents. The agents known by a BDI agent correspond to an agent that has a service (represented through a desire) that achieves a task, which composes its plan.

```
CONSTRUCT {
?a semMAS:knows ?a2.
}
WHERE {
?a rdf:type semMAS:BDI_agent.
?a semMAS:hasPlan ?bdip.
?bdip semMAS:represents ?dmp.
?dmp semDM:hasTask ?t.
?s semDM:achieves ?t .
?r semDM:delivers ?s.
?a2 semMAS:represents ?r.
}
```

Code 6.15: Main part of the SHACL rule for defining agents that an agent knows

Such rules allow creating the conceptual model of the simulation. The result of their application on a case study is presented in section 7.3.1 of the next chapter. The model's simulation depends on a programming model that must be created from the semMAS ontology to be then executed. It produces an assessment of disaster management response. The process of SemMAS interpretation into a programming model is presented in the next section.

6.2.2 Modeling of the programmed simulation model

The simulation modeling step's second process is transforming the conceptual simulation model into the programmed simulation model. The programmed model depends on the chosen simulation platform. In this thesis, the chosen platform is GAMA. Therefore, the programmed model corresponds to a GAML model. The transformation is achieved through rule-based reasoning. It uses SHACL rules and verifies the consistency of the inference before updating the knowledge base. This rule-based transformation is preceded by an inference using OWL2-EL and RDF(S) to deduce the most straightforward relations between the instances of the conceptual model and the programmed model. The OWL2-El profile has been chosen because it provides the best compromise between expressivity and reasoning performance in applications using large ontologies [W3C OWL Working Group, 2012], such as those used in this thesis. This inference is a monotonic inference of open-world assumptions (c.f. appendix A.1.2). OWL2 is used to define restrictions on concepts. For example, the agent's reflexes are based on skills, and the definition of an agent in GAML requires specifying its skills, corresponding to those used by a reflex of the agent. In section 5.3.2 of the previous chapter, an example of OWL restriction on the concept *semMAS:Transport_Aa*, which is a subclass of the concept *semMAS:Reflex* has been given. Thanks to the axiomatic-based reasoning on OWL2, all individuals of this concept are defined as using the skills *moving* and *actingOnPeople*. After the reasoning based on OWL2, the reasoning based on SHACL rules is applied to complete the representation of the GAML model. To continue the example of the agent's skills definition, let us show the central part of the SHACL rule in 6.16 that defines an agent's skills according to the skills used by a reflex that he has.

```
CONSTRUCT {
?a semMAS:hasSkill ?s.
}
WHERE {
?a rdf:type semMAS:Species.
?a semMAS:hasReflex ?r.
?r semMAS:usesSkill ?s.
}
```

Code 6.16: Main part of the SHACL rule that defines skills of an agent

The SHACL rules aim at defining the model's components specific to a GAML model. They also sometimes use concepts from the SemDM ontology. For example, the function assigning a location to an agent depends if he is located at a certain point, another geometry, or a feature as a specific building. The SHACL rule, whose main part is presented in 6.17, defines the agent's location, which is a point through the property *at_location*. In contrary, the agent's location corresponding to another geometry or a feature is defined through the property *any_location_in* as shown in 6.18.

```
CONSTRUCT {
?a semMAS:at_location ?l.
}
WHERE {
?a rdf:type semMAS:Agent.
?a semDM:represents ?r .
?o semDM:ownerOf ?r.
?o semDM:isLocatedAt ?l.
?l rdf:type geo:Point.
}
```

Code 6.17: Main part of the SHACL rule for creating agent's location from a point

```
CONSTRUCT {
?a semMAS:any_location_in ?l.
}
WHERE {
?a rdf:type semMAS:Agent.
```

```
?a semDM:represents ?r .
?o semDM:ownerOf ?r.
?o semDM:isLocatedAt ?l.
?l rdf:type geo:SpatialObject.
FILTER NOT EXISTS (?l rdf:type geo:Point.)
}
```

Code 6.18: Main part of the SHACL rule for creating agent's location from a feature or a geometry that is not a point

In GAML, BDI agent are represented through *SimpleBDI*, which is a specific *Species*. It is defined in SemMAS as a subclass of *BDI_agent* as shown by Figure 5.20 in section 5.3.2. The implementation of *SimpleBDI* has the characteristics of a *BDI_agent* but also some further requirements (as instances of *semMAS:Rule*). Therefore, agents defined as a *BDI_agent* or as a subclass of it are defined as a *SimpleBDI* through the SHACL rule, whose main part is presented in 6.19.

```
CONSTRUCT {
?at rdfs:subClassOf semMAS:SimpleBDI.
?a rdf:type semMAS:SimpleBDI.
}
WHERE {
?a rdf:type ?at.
?at rdfs:subClassOf semMAS:BDI_agent.
}
```

Code 6.19: Main part of the SHACL rule for creating simpleBDI agent that is the GAML model for BDI agent

Finally, let us take the last example of GAML specification. The concept *semMAS:Rule* is specific to a simple BDI agent of GAML and is used to define a new desire from a belief. In the case of manager agent, the SHACL rule, whose central part is shown in Code 6.20, generates an individual of *semMAS:Rule* on which it associates a belief, a desire and a fixed strength of one. This rule uses the relationship between the task that a belief represents and the service that achieves it to identify the associated desire. A similar SHACL rule exists for the central agent's beliefs. This other rule uses the relationship between the element at risk that a belief represents and the service that serves it to identify the associated desire.

```
CONSTRUCT {
?rule rdf:type semMAS:Rule.
?rule semMAS:requiresBelief ?b.
?rule semMAS:definedDesire ?d.
?rule semMAS:definedStrength "1"^^xsd:integer.
?a semMAS:hasRule ?rule.
```

```
}
WHERE {
?a rdf:type semMAS:SimpleBDI.
?a semMAS:hasBelief ?b.
?b semMAS:represents ?t.
?s semDM:achieves ?t .
?d semMAS:represents ?t.
?rule semTransform:generateURI(semMAS, "rule").
}
```

Code 6.20: Main part of the SHACL rule for creating rules of BDI agents that define a new desire according to a belief

6.3 Simulation design

The third step of the method is the simulation design. As illustrated in Figure 6.8, it begins by a generative programming process guided by the simulation model into SemMAS ontology. This process generates a simulation program and experiments. It is then followed by the execution of simulation experiments based on the simulation program.

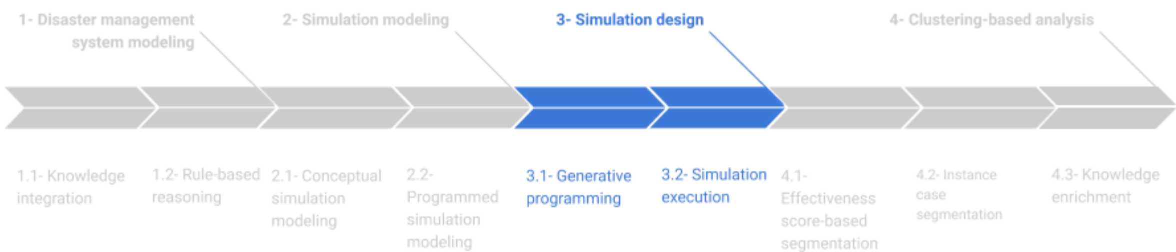


Figure 6.8: Multi-agent simulation design and execution in processing timeline

6.3.1 Generative programming

The simulation code is designed through an automatic approach of generative programming based on an ontology. This process aims at generating the simulation program and the simulation experiments to execute. This generative programming takes the programmed simulation model represented in the knowledge base to implement the simulation and simulation experiments. Figure 6.9 illustrates this process of generative programming by highlighting the steps corresponding to the different structure blocks illustrated in Figure 6.10. The generative programming has been developped in *Java* and uses *Apache Jena* for manipulating the ontology in *Java*.

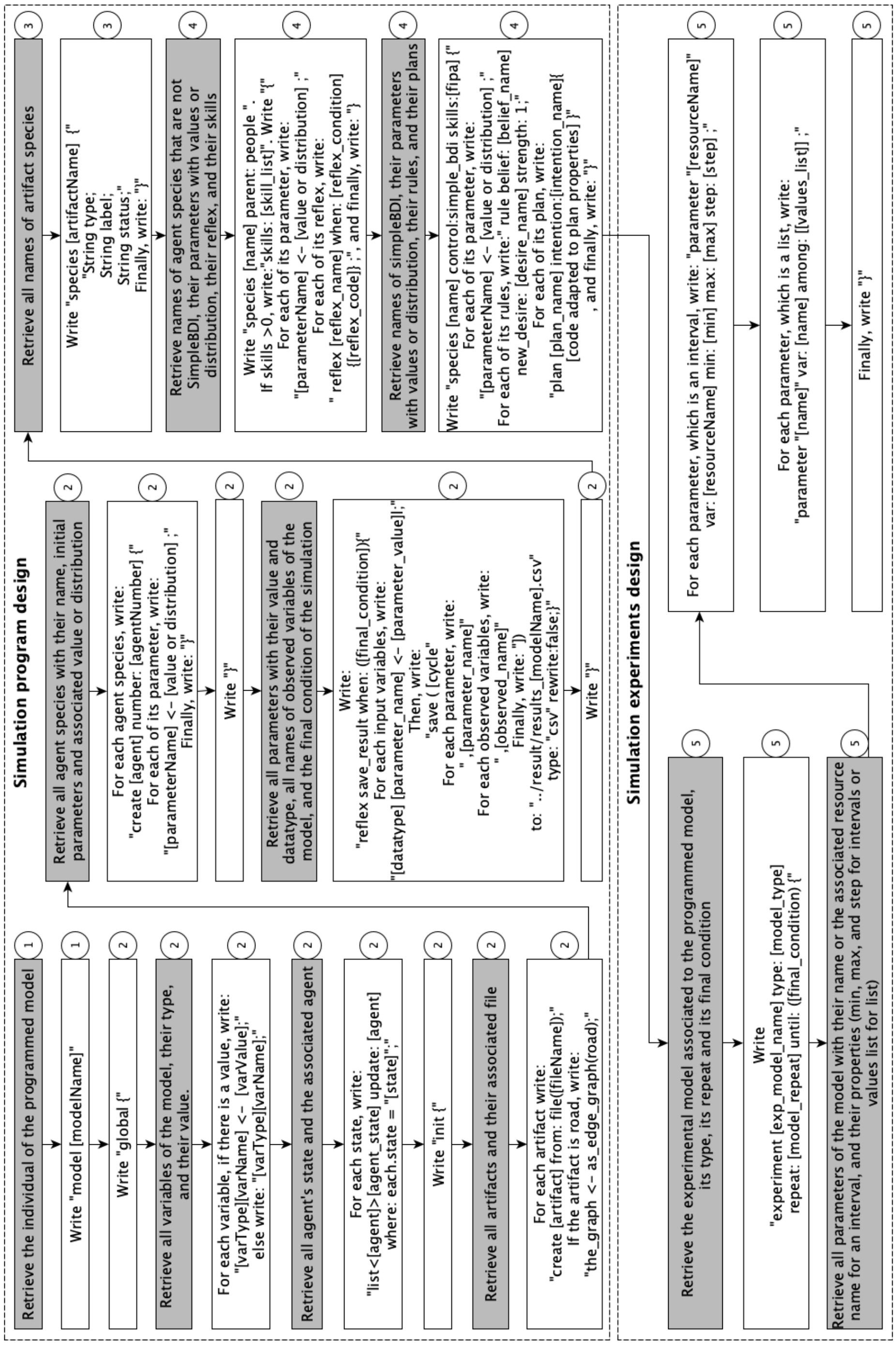


Figure 6.9: Main steps of the generative programming process, using retrieving actions through SPARQL queries highlighted in grey

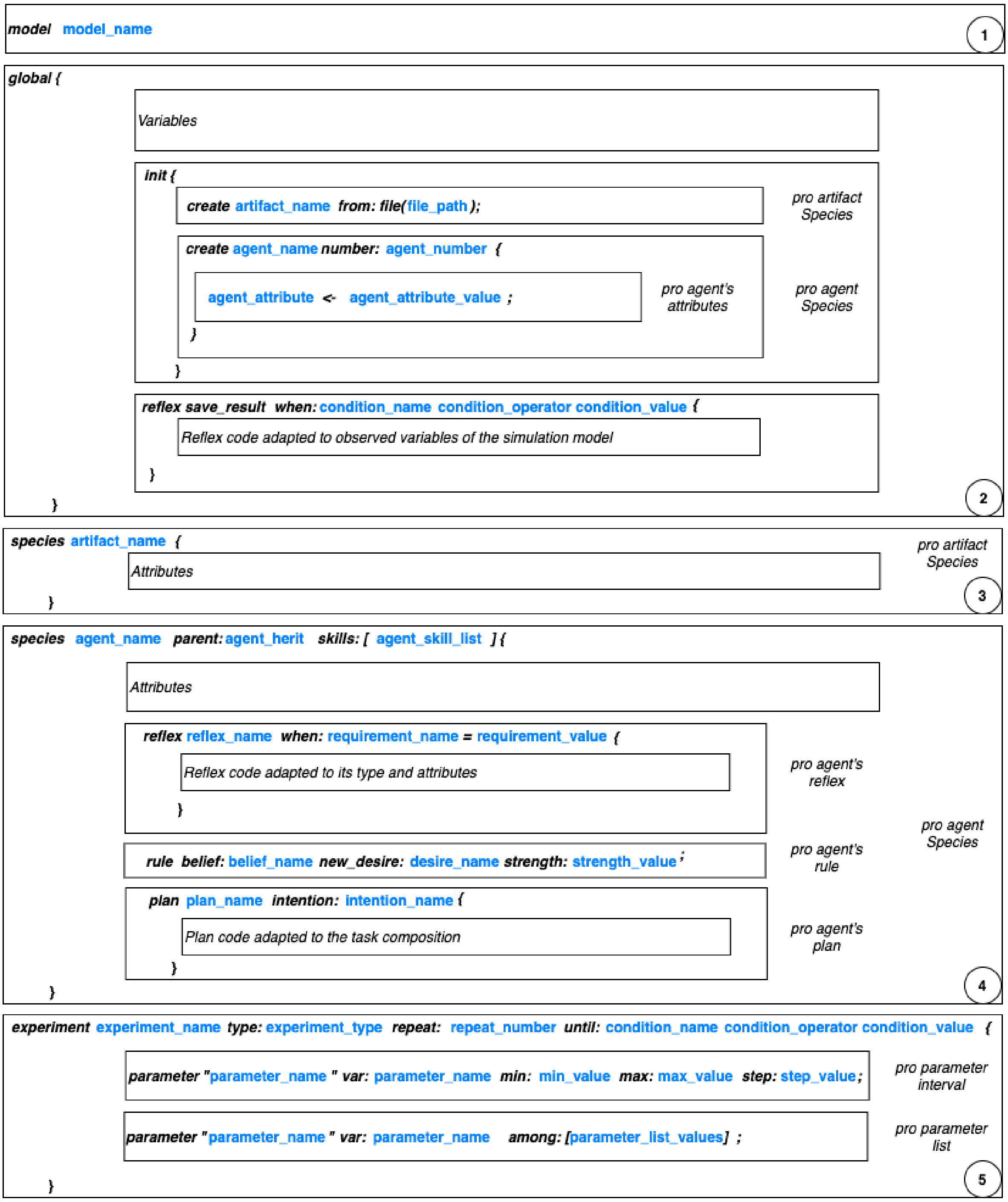


Figure 6.10: Program structure resulting from the generative programming process, where variables depending on the simulation model are represented in blue.

This process results in a GAML program whose structure is illustrated in Figure 6.10. The simulation program is designed from the knowledge base content. As presented in section 6.2.2, the knowledge base contains all knowledge related to the programming in GAML. Generative programming uses SPARQL to retrieve the different information related to GAML programming in a specific order, then write the simulation program. This programming begins by retrieving the model name to write the block 1 of the Figure 6.10. Then, the process programs the model's components that are:

- the environment, also called simulation's world representing by the *global species* and corresponding to the block 2 of the Figure 6.10,
- the regular species that represent:
  - the artifacts, illustrating as the block 3 of the Figure 6.10,
  - the agents, illustrating as the block 4 of the Figure 6.10.

Finally, it programs the simulation experiment, corresponding to block 5 of Figure 6.10.

Implementation of *global species* (c.f. block 2 in Figure 6.10) The global species definition is composed of three main parts: the global variables definition, the components initialization, and the reflex for saving simulation results. The global variables part aims mainly at defining the global variables of the simulation's world. These variables aim at defining the world components, supporting its evolution, and observing the simulation evolution. For example, a road network graph, if the model has road artifacts, is a variable used as a world component. Another example is the list of agents per state and their quantity linked to the list length. Such a list and quantity variables allow the monitoring of the evolution of agent by category. They are used to support the world evolution and observe the simulation evolution. The components' initialization part aims mainly at initializing the different artifacts and agents. Artifacts are composed of roads and buildings. They are initialized through their respective individuals in the ontology SemMAS and the geospatial file associated with them through the property *semMAS: hasFile* represents them. In the case of roads, their initialization is followed by the initialization of the graph. Agents are initialized through the retrieving of the individuals of the concept *semMAS: Agent*, their associated quantity (c.f. the property *semMAS:number*) and attributes (c.f. the property *semMAS:hasAttribute*) with their value. It also initializes the value of the model's parameters. For that, the process retrieves the individuals of the list or interval representing the model's parameter through the property *semMAS:hasParameter*. These individuals provide a parameter's name and an init value, obtained from the *rdf:first* of a list or the *semMAS:hasMin* of an interval. Finally, the process creates the reflex to save results from the model's parameters *semMAS: hasParameter*, its observed variables *semMAS: hasObservedVariable* and its output file's path.

Implementation of *species* representing an artifact (c.f. block 3 in Figure 6.10) Species representing artifacts are defined through the keyword *species* followed by the artifact name and its attributes between brackets. Therefore, the process firstly retrieves all individuals of the concept *semMAS: Artifact*. Secondly, it defines three attributes:

1. *String type;* represents the direct type of the artifact individual, which has no subclass. It allows the definition of subgroups of buildings.

2. *String label;* that corresponds to the building name, used to retrieve a specific building, and

3. *String status;* that corresponds to an artifact's status, which generally represents an impact of the disaster (e.g., damaged, flooded, blocked).

Implementation of *species* representing an agent (c.f. block 4 in Figure 6.10) The definition of species representing agents begins similarly than the artifact species, by retrieving individuals of agents and their attributes (e.g., status, objective, target, known agents). The species implementation begins with the keywords *Species* followed by the agent name. Then, it retrieves the property values that characterized the agent type. One the one hand, it retrieves the agent's reflex in the case of an actor agent, which is a reactive agent. On the other hand, it retrieves agent's rules and plans in the case of a central or manager agent, which is a cognitive BDI agents. For each reflex of a reactive agent, the process retrieves the individual representing it, all properties linked to this reflex individual and its direct type. The reflex header is built from the individual name of the reflex and according to the value associated to the properties *semMAS:what*, *semMAS:requiresStatus*, and *semMAS:requiresObjective* to define the requirement of the reflex. Then, the reflex type and the values of its properties are used to define its code. For each rule of a BDI agent, the process retrieves its associated beliefs (*semMAS:requiresBelief*), desire (*semMAS:definedDesire*), and strength (*semMAS:definedStrength*) to define the rule. For each plan of a BDI agent, the process retrieves its individual and its associated properties. The individual name and its associated intention (*semMAS:hasIntention*) allows the definition of its header. It then, retrieves the other associated properties of the plan (e.g. *semMAS:hasFinalCondition*, *semMAS:requests*, *semMAS:requestsQuantity*) to define the plan's code.

Implementation of *experiment* (c.f. block 5 in Figure 6.10) The end of the generative programming in GAML consists of defining simulation experiments. The process retrieves individuals of *semMAS: Experimental_model* and their closest type corresponding to a subclass of *semMAS: Experimental_model*, firstly. Secondly, it retrieves the number of experiment repetition for a same configuration and the final condition of the experimental model through the properties *semMAS:repeat* and *semMAS:hasCondition*, respectively. The associated experiment repetition number is 100 by default. The final condition is characterized by an operator through the property *semMAS:hasOperator* and a value through the property *semMAS:hasValue*. Finally, it retrieves the parameters of the experimental model through the property *semMAS:hasParameter*. There are two types of parameters: lists and intervals. In the case of a list, the process, retrieves the individual name and all its values thanks to the property *rdf:first* of the list to define the parameter. In the case of an interval, it retrieves its minimal value (through the property *semMAS:hasMin*), its maximal value (through the property *semMAS:hasMax*), its step value (through the property *semMAS:hasStep*), and the individual name of the resource, whose it defines the proportion (through the property *semMAS:definesProportion*) to define the parameter.

6.3.2 Simulation execution

The GAMA platform provides different ways to execute simulation experiments: through user interface[6], through command line in terminal[7], through command line in headless mode [8]. The simulation is executed by the processing server using the GAMA platform. This execution is achieved through the command line in headless mode. The processing server uses the bash command configured with:

1. the shell script dedicated to run a GAMA experiment in headless mode,

[6]See documentation for execution through user interface
[7]See documentation for execution through command line in terminal
[8]See documentation for execution through command line in headless mode

2. the input parameter file (*$1*), which is an XML file defining experiment parameters and attended outputs, stored in model libraries illustrated as A in Figure 6.11, and

3. the output directory path (*$2*), which is the directory that contains simulation results, illustrated as B in Figure 6.11.

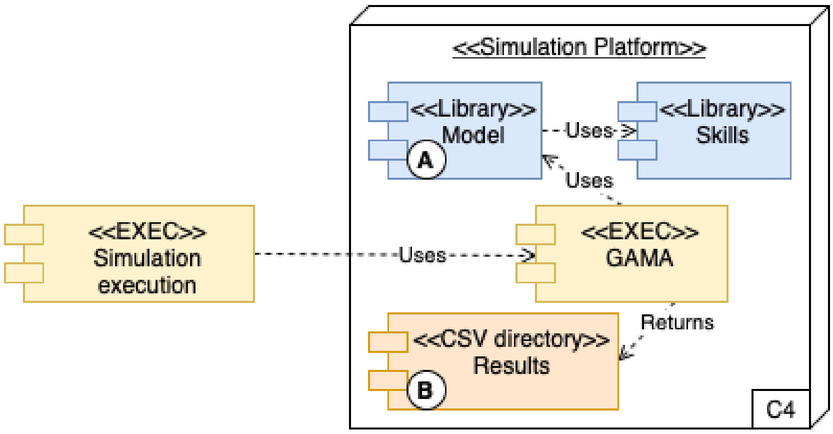


Figure 6.11: Simulation execution in deployement diagram

The shell script and the output directory path are fixed and do not change from one simulation to another. Only the input parameter file varies according to the experimental model of the GAML model represented in the SemMAS ontology.

This file describes the experiment plan and is composed of three parts for each experiment description.

1. *Heading* that specifies a simulation id, the source path of the GAML model, the number of simulation step to run, and the experiment name to run on the model, which has been defined in the GAML model,

2. *Parameters* that are defined through their name, type, and value,

3. *Outputs* that are defined by a name and a frequency of the monitoring.

Each experiment description is defined iteratively through a loop to represent each configuration with the experiment plan's parameter values. This experiment plan file is generated after the GAML model and is stored in a sub-directory (experiment) of the *model libraries*. After adding the GAML model and the experiment plan in model libraries, the bash command is executed through the processing server to run the simulation experiments, whose results will be added in the *CSV directory*. Section 7.4.2 of the next chapter illustrates this process and the simulation results for a case study.

6.4 Clustering-based analysis

The fourth and last step of the method is the clustering-based analysis process, as illustrated in Figure 6.12.

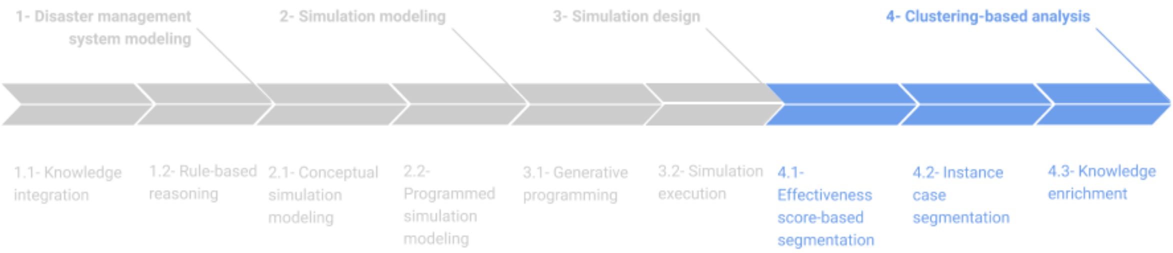


Figure 6.12: Clustering-based analysis in processing timeline

The assessment of the plans consists of enriching the knowledge base to represent the assessed plan's effectiveness regarding the different simulation experiments. Figure 6.13 illustrates the input and output of the analysis module that achieves the plan assessment.

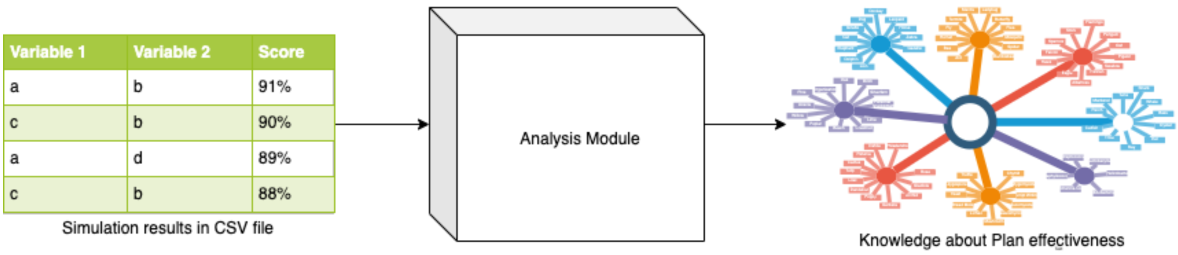


Figure 6.13: Clustering-based analysis overview.

The plan assessment is performed through a clustering-based analysis composed of two or three steps according to its first processing step results. Figure 6.14 illustrates the workflow of the analysis module.

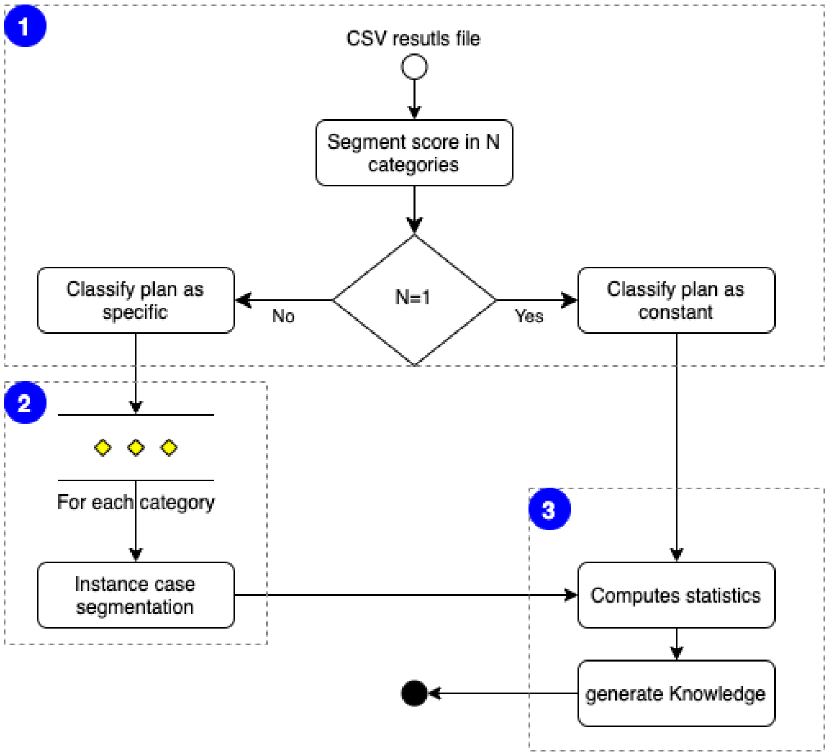


Figure 6.14: Plan Assessment workflow.

The first step consists of computing effectiveness rating for each simulation and applying an effectiveness score segmentation. If the categories number resulting from the segmentation is equaled to one, the plan is classified as global. Otherwise, it is classified as specific. If the plan is considered global for the case study, the analysis module enriches the knowledge base with statistics of the effectiveness rating (the third step in figure 6.14). In the case where the plan is considered specific, the analysis module segments the specific cases according to their similarity (second step in figure 6.14). The knowledge base is then enriched with statistics on each case's effectiveness rating (the third step in figure 6.14). The three sub-steps highlighted in Figure 6.14 are (1) the effectiveness score-based segmentation, (2) the instance case segmentation, and (3) the knowledge enrichment corresponding to 4.1, 4.2, and 4.3 of Figure 6.12, respectively. Each of these sub-steps is further detailed in the next subsections.

6.4.1 Effectiveness score-based segmentation

The plan's effectiveness is defined using the percentage of satisfaction of the plan's objective on the different simulation experiments carried out. The objectives can be quantitative (such as not exceeding a financial blow, a percentage of mortality, protecting specific areas) or qualitative (such as stabilizing the victims so that their state of health does not worsen). For example, a quantitative disaster management plan may protect sensitive areas (such as power plants, hospitals) from flooding. In this case, the percentage of the plan's objective being satisfied corresponds to the percentage of adequately protected areas out of the total area. An example of a systematic qualitative plan is victim stabilization, which consists of preventing the victim's state of health from worsening. In this case, the fulfillment percentage of the plan's objective corresponds to the victims' percentage of who have not changed their condition out of the number of victims who have changed their condition. That said, plan objectives can be very complex. For example, the change in a victim's condition must be considered differently if a victim's condition changes from severe to fatal or mild to severe. Therefore, such complexity requires a high degree of flexibility in the calculation based on the simulation variables. The simulation results and all the states variation are exported in a spreadsheet where each line represents an experiment of the simulation, to allow maximum flexibility. The plan objective's percentage of satisfaction is calculated using functions adequately defined for the elements at risk.

Thanks to the plan's effectiveness' value for each experiment, the plan can be classified into two categories:

Global : The plan has constant effectiveness for the case study.

Specific : The plan has variable effectiveness depending on specific situations (i.e., conditionally related to a situation).

Classifying a plan as global or specific requires determining whether the effectiveness rating of a plan varies significantly with the simulation experiments conducted. A significant variation implies being able to differentiate whether values are sufficiently different to represent different categories. Thus, the variation study in values is reduced to the segmentation of values into several categories.

6.4.2 Instance case segmentation

When a plan is relatively effective in a specific situation, it is necessary to determine the factors related to its effectiveness to define the plan's effectiveness related to situations. It is necessary to study the commonalities and differences encountered in experiments with similar scores to identify the factors impacting the plan's effectiveness. For this purpose, the experiments that produce a score classified in the same group in the previous process presented in section 6.4.1 are grouped to form a study set (see figure 6.16).

The analysis of these sets aims to generate cases specifying what the design's effectiveness is related to as a function of the situation (variable or set of variables). Figure 6.15 illustrates the workflow of this analysis.

Figure 6.15: Segmentation workflow

It begins with the segmentation of each variable constituting a case study. The different variables in each case study (representing the columns of the spreadsheet containing the experiments) are segmented into groups by the same strategy as in section 6.4.1. Thus each experiment is linked to different clusters according to the different variables studied (see figure 6.16).

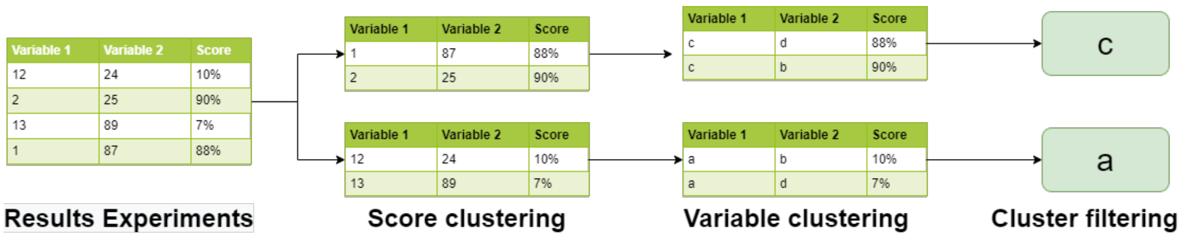


Figure 6.16: Illustration of Segmentation

These clusters are then filtered to keep only the clusters common to each case study. The set of clusters retained constitutes the main factors impacting the effectiveness of the design. They are then analyzed to enrich the knowledge base, as explained in section 6.4.3).

6.4.3 Knowledge enrichment

The final step in assessing the plan is knowledge enrichment. It allows SemDM to be enriched with the potential effectiveness of the plan.

If the plan has constant effectiveness, it is classified as a "Global Plan," and the basic statistics (minimum, maximum, average, median, and mean square) are assigned.If, on the contrary, the plan has relative effectiveness in different situations, the knowledge is enriched by these same statistics but defined for each situation considered as specific. Thus the plan has a clearly defined effectiveness according to various situations composed of factors impacting its effectiveness.

6.4.3.1 Global effectiveness

When a plan is considered to have global effectiveness following the classification step described in section 6.4.1, it is classified as global by assigning to it the instance *semDM:global_applicability* through a SPARQL insert query. Then, an instance of the concept *semDM:Effectiveness* is assigned to the plan in the SemDM ontology and statistics are assigned to its effectiveness using the properties *hasMin*, *hasMax*, *hasAverage*, *hasMedian*, *hasMeanSquare*. These assignments are automatically performed by the analysis module, performing a SPARQL insert query with a predefined structure.

Figure 6.17 illustrates the enriched knowledge for a plan classified as global.

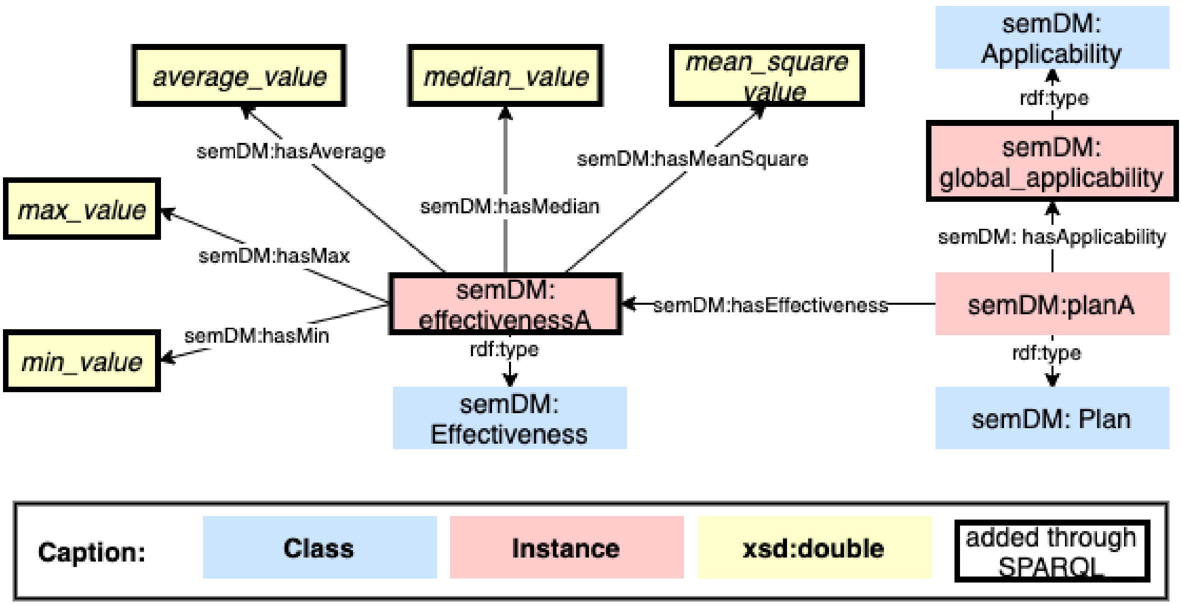


Figure 6.17: Illustration of Knowledge enrichment for global Plan

6.4.3.2 Specific effectiveness

When the evaluated plan is considered relatively effective, the knowledge base is enriched by details of each situation for which the plan's effectiveness is relative. Then statistics on the effectiveness rating are added for each of them. Figure 6.18 illustrates the enriched knowledge for a plan classified as relative. This enrichment is achieved through a SPARQL insert query.

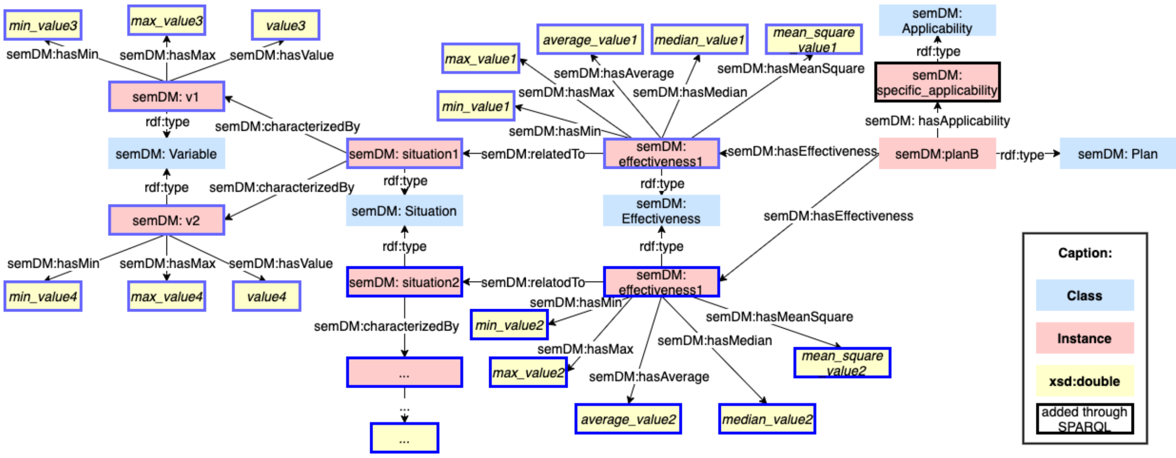


Figure 6.18: Illustration of Knowledge enrichment for specific Plan (with different shades of blue box to represent the different effectiveness clusters)

The specific situations are grouped as a cluster set after the clustering step explained in section 6.4.2. The statistics on the effectiveness rating corresponding to each situation are therefore linked to each of these sets. They form the "constructor" part of the query. Each cluster relates to a specific variable and has a set of values. The minimum and maximum values set are used as a bound for the variable's value specific to the cluster. Each variable (VAR) is thus linked to bounds and provides the different conditions required for a plan to be a deterministic effectiveness statistic. Thanks to the Hierarchical clustering process of variables and the Cluster filtering process, each heterogeneous (i.e., no common variable value) set is discomposed into a homogeneous set (i.e., at least one common variable). Thus the knowledge enrichment process is similar for every cluster and consists of fulfilling a predefined SPARQL query. This query is hardcoded through Embedded JavaScript templating ("ejs")[9] in which elements encapsulated by $'' < \%\% > ''$ are dynamically fullfill through a JSON object. Code 6.21 presents this SPARQL query and code 6.22 presents a model of JSON object.

```
CONSTRUCT{
    semDM:<%=data.plan%> semDM:hasEffectiveness ?e.
    ?e semDM:relatedTo ?s.
    ?e semDM:hasMin <%=data.e.minValue%>.
    ?e semDM:hasMax <%=data.e.maxValue%>.
    ?e semDM:hasAverage ?average.
    ?e semDM:hasMedian ?median.
    ?e semDM:hasMeanSquare ?meanS.
    ?s rdf:type semDM:Situation.
    <%for(var i = 0; i < data.v.length; i++) { %>
    ?s semDM:characterizedBy ?v<%=i%>.
```

[9] https://ejs.co/

```
    ?v<%=i%> rdf:type semDM:Variable.
    ?v<%=i%> semDM:linkedWith ?init<%=i%> .
    <% if(data.v[i].hasValue) {%>
    ?v<%=i%> semDM:hasValue <%=data.v[i].hasValue %> . <% }%> <% if(data.v[i].hasValueMax) {%>
    ?v<%=i%> semDM:hasMax <%=data.v[i].hasValueMax %>. <% }%> <% if(data.v[i].hasValueMin) {%>
    ?v<%=i%> semDM:hasMin <%=data.v[i].hasValueMin %>.<% }%> <% }%>
    <% }%>
  } WHERE {
    ?e semTransform:generateURI(semDM, "effectiveness").
    ?s semTransform:generateURI(semDM, "situation").
    <%for(var i = 0; i < data.v.length; i++) { %>
    ?v<%=i%> semTransform:generateURI(semDM, <%=data.v[i].name %> ).
    ?init<%=i%> semTransform:searchIndividual(semDM, <%=data.v[i].name %> ).
    <% }%>
    ?average semStatistics:GetAverage(<%for(var i = 0; i < data.e.values.length; i++) { %> "<%=data.e.values[i] %>"<% if(i< data.e.values.length-1) {%>,<% }%> <% }%>) .
    ?median semStatistics:GetMedian(<%for(var i = 0; i < data.e.values.length; i++) { %>"<%=data.e.values[i] %>" <% if(i< data.e.values.length-1) {%> , <% }%> <% }%>) .
    ?meanS semStatistics:GetMeanSquare(<%for(var i = 0; i < data.e.values.length; i++) { %>"<%=data.e.values[i] %>"<% if(i< data.e.values.length-1) {%> , <% }%> <% }%>) .
  }
```

Code 6.21: Predefined SPARQL CONSTRUCT query to enrich knowledge

```
{
  "plan":"Plan_A",
  "e":{
    "values":[min,max],
    "minValue":min,
    "maxValue":max
    },
  "v":[{
    "name":"name1",
    "hasValue": val1,
    "minValue":valMin1,
    "maxValue":valMax1
    },
    {
    "name":"name2",
    "hasValue":val2,
    "minValue":valMin2,
```

```
        "maxValue":valMax2
        }]
    }
```

Code 6.22: JSON Object model used to fulfill the SPARQL query

6.5 Summary

This chapter has presented the method's implementation composed of four primary steps: the disaster management system's modeling, the simulation modeling, the simulation design, and the plan assessment.

Disaster management system's modeling The disaster management system's modeling is achieved by the user directly and from the user's data. Data provided by a user is processed to extract knowledge and integrate them into the ontology SemDM presented in section 5.3.1 of the previous chapter. This approach uses a schema mapping between the table structure of the data and OWL concepts to transform the data into an RDF graph. A set of analysis processes (based on geometries, column names, cell names) identifies concepts and individuals of Wikidata to represents the data semantically. It produces an RDF graph based on Wikidata. This RDF graph is integrated into the SemDM ontology, on which axiomatic-based reasoning is applied. Thanks to the interlinking between SemDM and Wikidata, the reasoning allows SemDM ontology enrichment with knowledge extracted from data. The disaster management modeling is completed through rule-based reasoning on the SemDM ontology. This reasoning allows the definition of geospatial relations according to event knowledge. All rule-based reasonings of the proposed method have been implemented to avoid an undecidable problem and guarantee knowledge base consistency.

Simulation modeling Simulation modeling is based on the disaster management model. It generates the conceptual simulation model through rule-based reasoning into the SemMAS ontology presented in section 5.3.2 of the previous chapter. This model is composed of primary components of a multi-agent simulation as the environment and agents. The SemMAS ontology is then completed with a representation of the programmed simulation model specific to the GAMA platform. This second model corresponds to the implementation representation of the conceptual model. It is obtained through two reasoning steps. The first one is axiomatic-based reasoning applied to the SemMAS ontology and allows specific instances according to the vocabulary specific to the GAMA platform. Then, rule-based reasoning is applied to instantiate the programmed model with elements specific to the platform. These rules are generally both based on concepts from SemDM and SemMAS ontologies.

Simulation design The simulation design is composed of two steps: a generative programming process and the simulation experiments execution. The generative programming uses the programmed model's representation into SemMAS ontology to design the simulation program and experiments adapted to the GAMA platform. This process retrieves knowledge on the programmed model through SPARQL queries in a specific order to respect the structure of a GAML model. The generative programming results in a GAML model containing both a simulation model and an experimental model stored in the model library. Then, an experiment plan is generated and stored in the model library to execute the simulation experiments. The execution of simulation experiments is achieved through a command line in headless mode. Simulation results are added to the CSV directory.

Plan assessment The plan assessment uses a clustering-based analysis of simulation results. This process firstly computes the effectiveness rating of each simulation based on the observed variable values. Secondly, it applies a hierarchical clustering to segment simulation experiments according to their effectiveness rating. If it results in a cluster number of one, the plan is classified as global; otherwise as specific. In a specific plan, a second clustering is applied, for instance, case segmentation. This clustering aims at determining variables that characterize a situation by impacting the plan's effectiveness. Finally, this process enriches the SemDM ontology with an adequate representation of the plan's effectiveness. In a global plan, the plan has global applicability and unique effectiveness defined through a set of values (e.g., min, max, median). In the case of a specific plan, the plan has specific applicability and several effectiveness values related to a situation. A situation is characterized by a set of variables that are defined through minimum and maximum values.

The next chapter presents the method applied to a use case, allowing the illustration of the method application's results step by step.

7 Use Case

Contents

7.1 Case study . . . 204

7.1.1 Plan description . . . 204

7.1.2 Scenarios . . . 206

7.2 Modeling of disaster management case study . . . 207

7.2.1 Knowledge integration . . . 208

7.2.2 Reasoning on disaster management model according to an event . . . 219

7.3 Simulation modeling . . . 220

7.3.1 Conceptual simulation modeling from disaster management case study . . . 220

7.3.2 Program modeling from the conceptual simulation model . 227

7.4 Simulation design . . . 229

7.4.1 Generative programming . . . 230

7.4.2 Simulation execution . . . 239

7.5 Clustering-based analysis . . . 241

7.5.1 Effectiveness score-based segmentation . . . 242

7.5.2 Instance case segmentation . . . 244

7.5.3 Knowledge enrichment . . . 246

7.5.4 Plan assessment result . . . 248

7.6 Summary . . . 253

The previous chapter has explained the implementation of the proposed method to assess disaster management plans. This chapter aims at illustrating the results of the method's implementation on a use case, step by step. A use case requires at least a plan to assess. The french NOVI plan has been chosen as a case study for plan assessment. This plan is assessed through different scenarios based on three configurations of a disaster event. This chapter explains the case study firstly through the description of the NOVI plan and the three scenarios. Secondly, it presents the modeling of the disaster management case study, through knowledge integration and rule-based reasoning on SemDM ontology. From the SemDM ontology, the results of the conceptual and programmed simulation modeling are presented. Then, it shows the programs, experiments, and results obtained by the simulation design step. Finally, it describes the results provided by the clustering-based analysis process.

7.1 Case study

The case study presented in this section is the NOVI (NOmbreuses VIctimes) plan[1].

This plan aims to rescue a large number of victims (more than 100) in the same place and organize first aid. It can be triggered in the event of a significant fire, building collapse, road, rail or air traffic accident, criminal acts (e.g., hostage-taking, terrorism), natural or technological disaster. The NOVI plan was chosen because of the diversity of situations in which it can be triggered, but also because it is a derivative of the French ORSEC[2] plan. The ORSEC plan, also known as the ORSEC facility, was developed following "*The law on the modernization of civil security of 17 August 2004* [defining] *the measures to be put in place to prevent the population and infrastructure from major risks*" [Barkaoui et al., 2016].

The ORSEC plan is the national french strategy and the guideline that provides advice to prepare a disaster management plan. This national strategical plan concerns, thus, all locality in France. However, small towns generally have inadequate resources and not enough money to train and be prepared for such a plan. Therefore, this plan has been assessed in three configurations of a disaster event impacting the french town Montbard.

This section describes the NOVI plan, firstly, and secondly, the specificities of the three scenarios in which the plan is assessed.

7.1.1 Plan description

The NOVI plan aims to manage the situation of multi-victims and ensure the rapid and correct treatment of the many victims while avoiding hospital overcrowding.

[1] NOVI Plan: `https://www.gouvernement.fr/risques/plan-NOVI`

[2] ORSEC (Organisation de la Réponse de SEcurité Civile) Plan : `https://www.interieur.gouv.fr/Le-ministere/Securite-civile/Documentation-technique/Planification-et-exercices-de-Securite-civile`

It is based on a victim triage system to optimize victim management.
After a disaster happens, rescuers are the first informed and the first to see and discover the situation. Their role is to trigger the arrival of suitable reinforcements, carry out priority rescue actions and enable the ORSEC system to be triggered with numerous victims, commonly known as NOVI [Départementale-métropolitaine des jeunes sapeurs pompiers, 2017]. The NOVI plan structure is illustrated in Figure 7.1.

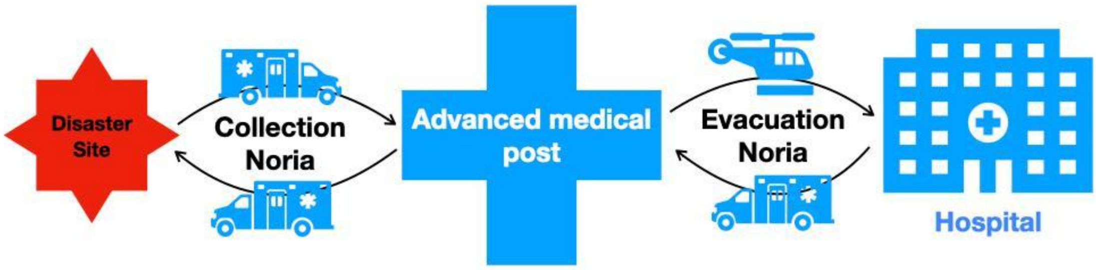


Figure 7.1: NOVI Plan schema.

The NOVI plan is composed of three main areas of victim treatment: the disaster site, the advanced medical post, and the hospital. The access of the disaster site is limited to rescuers empowered to manage the disaster event. During an event of terrorism, the rescuers empowered to manage the situation are the national gendarmerie's response unit[3]. However, in the majority of events, the rescuers empowered to manage the situation are the firemen. These rescuers' role is to categorize victims and extract them from the site to the advanced medical post.

Five categories are distinguished from classifying the victims:

- Emergency Overwhelmed/Death (D),
- *absolute emergencies*:
  - Extreme Emergency (EU)
  - and Critical Injury (U1)
- *relative emergencies*:
  - Serious Injury (U2)
  - and Minor Injury (U3)

[3]The national gendarmerie's response unit corresponds to "Groupe d'Intervention de la Gendarmerie Nationale (GIGN)" in french

The more the emergency is high, the more the priority of evacuation of the victim is top.

The evacuation of victims from the disaster site to the advanced medical post is called *Collection Noria*. This advanced medical post is an intermediate point between the disaster site and the hospital. It can be an existing building or a modular system (e.g., tents) and must be close to the disaster site and close to the road allowing rapid access to rescue supplies. The advanced medical post aims at examining victims to distribute them appropriately in hospitals and avoid an overload of hospitals. Victims sent to the advanced medical post, receive stabilization first aid and are then re-sorted for evacuation to a hospital. The evacuation of victims to a hospital is called *Evacuation Noria*.

The NOVI plan begins thus, when the advanced medical post is set up.

Therefore, to wait for the beginning of the NOVI plan with an operational advanced medical post actively, there is a timeframe of management where rescuers follow the NOVI plan alpha (presented in [Départementale-métropolitaine des jeunes sapeurs pompiers, 2017]).

The NOVI plan alpha is a pre-step of the NOVI plan. The first rescuers are aware of the situation by identifying victims, assessing their health state, and labeling them. Thus, when the NOVI plan begins, we can consider that the situation awareness on site is done, and all stakeholders are at their position of work. The studied case focuses only on the NOVI plan. Its modeling contains the results of the NOVI plan alpha as an input, in the sense of the number and states of victims are known, when the NOVI plan begins.

7.1.2 Scenarios

Montbard The concrete case study corresponds to the NOVI plan application in the town of Montbard in France. This town is composed of more than 5000 inhabitants, has a railway station on the line Paris-Marseilles, and some industries, whose a Seveso[4] site. This town is thus exposed to natural risks as a flood in case of dam break or intense rainfall and a lot of technological or human-made risks such as train or truck accidents and explosion with chemical exposure for inhabitants. Moreover, as other french communes, Montbard is also exposed to terrorist attacks. All of these hazards can trigger the set up of plans, whose NOVI plan.

Events in Montbard The disaster event defined for the three scenarios impacts the

[4]Seveso directive: `https://ec.europa.eu/environment/seveso/`

Hall *Paul Éluard* in Montbard. This site has been chosen for its capacity of 750 people and its frequent use for hosting events. This site is thus, a potential site of disaster with more than 100 victims. The three scenarios correspond to three different configurations of victims: one with 250 victims, one with 500 victims, and one with 750 victims. The distribution of the victim's health state has not been defined.

Resources in Montbard In these scenarios, the town of Montbard has 19 physicians, 11 ambulances, and its fire brigade is composed of 28 firefighters and 17 firefighter officers. The school closest to Hall *Paul Éluard* and called *Paul Langevin* has been chosen as an advanced medical post in case of a disaster in Hall *Paul Éluard*. It has been chosen for its proximity to the disaster site, its access facilitating the entry and exit of victims, and its emergency equipment. These scenarios have been designed from discussion with experts and documentation, but it is not an official one. The case study aims to assess the NOVI plan in Montbard for an event impacting the Paul Eluard hall. In this plan, the distribution of physicians between the various services for which they are responsible for the NOVI plan, i.e., the care of victims in hospitals or at the advanced medical post, has not to be planned. It will thus, be considered as a simulation parameter by the simulation conceptualization process.

7.2 Modeling of disaster management case study

This section aims at presenting the processes intervening in the application of the first method's step, as illustrated in Figure 7.2. It firstly presents knowledge integration in section 7.2.1. The knowledge integration concerns geospatial knowledge, NOVI plan description and events description. Secondly, it describes the rule-based reasoning results allowing the deduction of the geospatial relationship of the disaster management model.

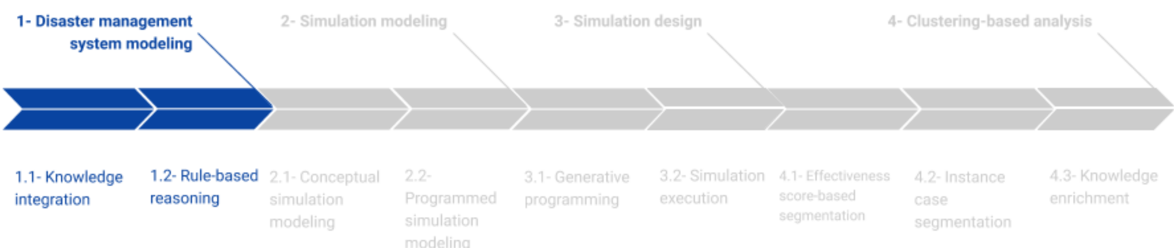


Figure 7.2: First step of the application: Design of the disaster management model corresponding to the case study

7.2.1 Knowledge integration

The knowledge integration aims to model the disaster management system corresponding to the case study described previously. It includes the description of the NOVI plan, the geospatial representation of Montbard, resources available in Montbard, and the events configuring the scenarios to assess the plan. As explained in section 5.4, it is suitable to begin by the integration of geospatial representation through the geospatial data interpretation. Therefore, this section presents firstly the results obtained from the data interpretation process. Secondly, it presents the direct modeling of the case study through events and the plan description.

7.2.1.1 Data Interpretation For Geospatial Knowledge Integration

As explained in section 5.4, geospatial modeling depends on the governmental echelon of the event. In this case study, the governmental echelon is Montbard. Therefore, the interpreted geospatial data represent the roads and the buildings of Montbard. These data have been retrieved as shapefile from *OpenStreetMap*[5]. Roads data contain information related to different types such as highway, waterway, or railway. Buildings data contain information related to infrastructures such as parking, residential building, hospital, fire station, or shop. These two data have been interpreted and integrated into the knowledge base using the knowledge extraction and integration process described in Section 6.1.2 of the previous chapter. Let us illustrate the result of this process on a sample of the road dataset presented in Table 7.1.

[5]OpenStreetMap: `https://www.openstreetmap.org`

osm_id	name	highway	waterway	railway
272265892	Rue des Roches	residential		
272265893	Rue du Faubourg	residential		
8043441	Route de Châtillon	primary		
279777837	Route de Semur	primary		
38819616	Quai Joseph-Maire	cycleway		
55943818	Ruelle des Renards	footway		
75279144	Ligne PLM de Paris à Lyon			rail
199689971	Ligne PLM de Paris à Lyon			rail
75839743	Canal de Bourgogne		canal	
254875356	Canal de Bourgogne		canal	
296096180	La Brenne		river	
296096185	La Brenne		river	

Table 7.1: Sample of the road dataset to process

This process depends on the interlinking between the concepts of SemDM ontology and Wikidata. Therefore, the mapping related to the sample of road dataset (Table 7.1) is presented in Table 7.2.

SemDM ontology	Wikidata ontology
Road	Q34442
Infrastructure	Q121359
Waterway	Q1267889
Highway	Q269949
Railway	Q22667

Table 7.2: Sample of mapping between concepts from SemDM and Wikidata

The interpretation of the sample data has resulted in the creation of a concept hierarchy based on the identification of noun occurrences in the same column. The hierarchy resulting from this process is presented in Figure 7.3.

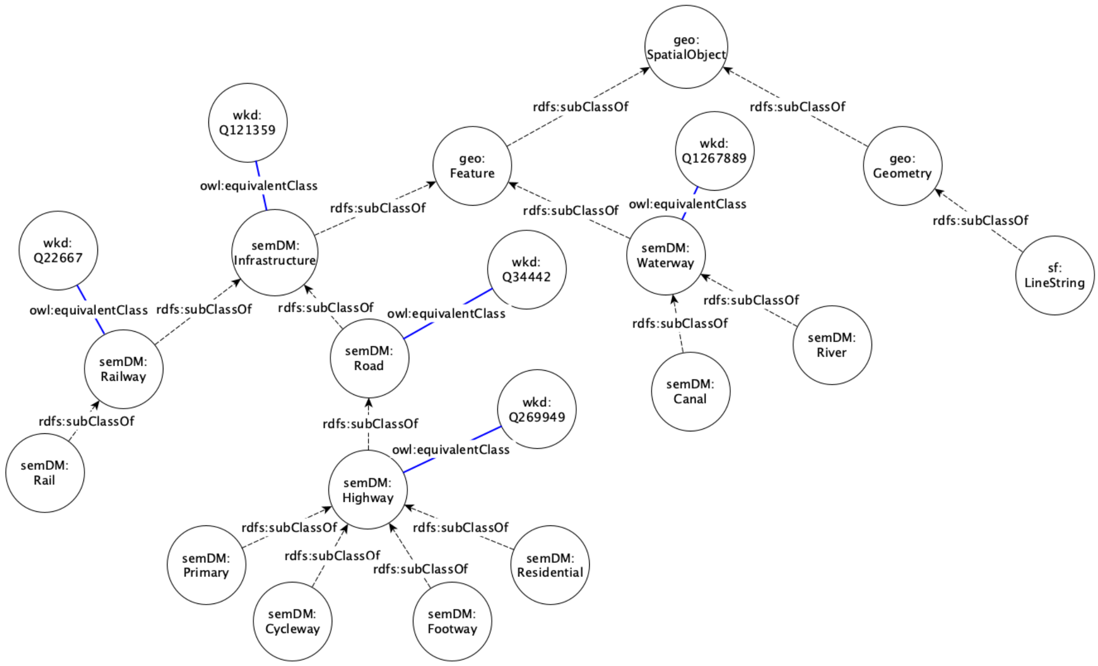


Figure 7.3: Illustration of the concept hierarchy resulting from the interpretation process of the data sample presented in Table 7.1 and the defined mapping sample is shown in Table 7.2

Individuals representing the knowledge contained inside the data are created by using existing concepts and newly created concepts. They are then added to the knowledge base. Thanks to the interpretation process, each created individual has a type and is linked to values or other individuals through properties. The definition of some individuals (corresponding to the grey rows in Table 7.1) resulting from the interpretation of the dataset sample are presented in Figure 7.4.

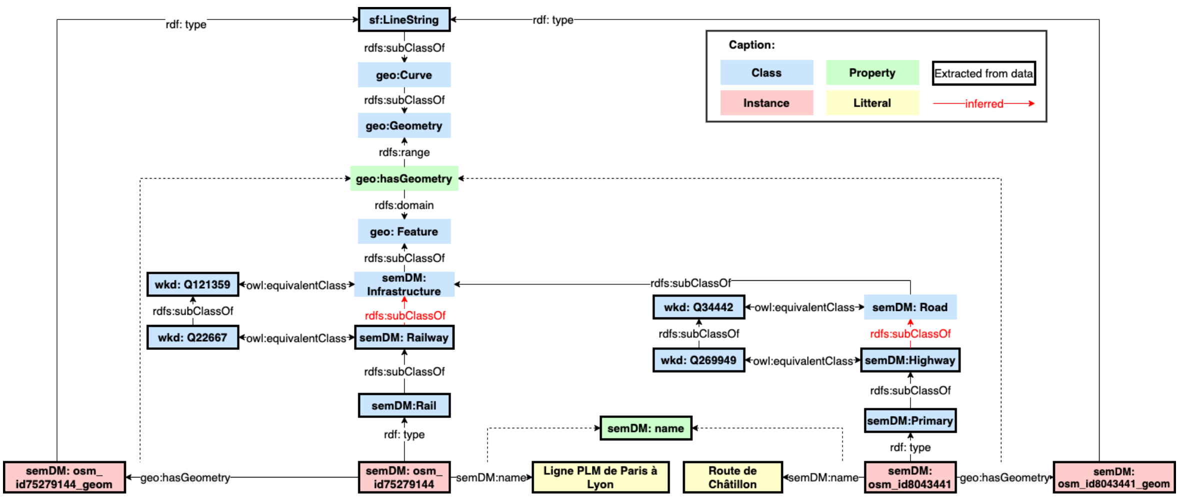


Figure 7.4: Illustration of some individuals resulting from the interpretation process of the data sample presented in Table 7.1

7.2.1.2 Event Modeling

After the interpretation of data, it is suitable to add disaster modeling before the plan description. The disaster description uses geospatial knowledge, whereas the plan description uses both geospatial and event knowledge. As explained in section 7.1.2, the case study is assessed on a disaster based on an event, represented through the individual *semDM:event1* and located at the Paul Eluard hall (*semDM:paulEluard_hall_montbard*). It impacts a casualty's individual that has the state *semDM:inDanger*. The case study aims at assessing the NOVI plan in three configurations of casualties quantity. These different configurations of casualties quantity is represented through a *rdf:List*[6]. The three casualties quantities correspond to the values of the property *rdf:first*, which are 250, 550, and 750. Figure 7.5 illustrates the modeling of this event with its three casualties quantity configurations.

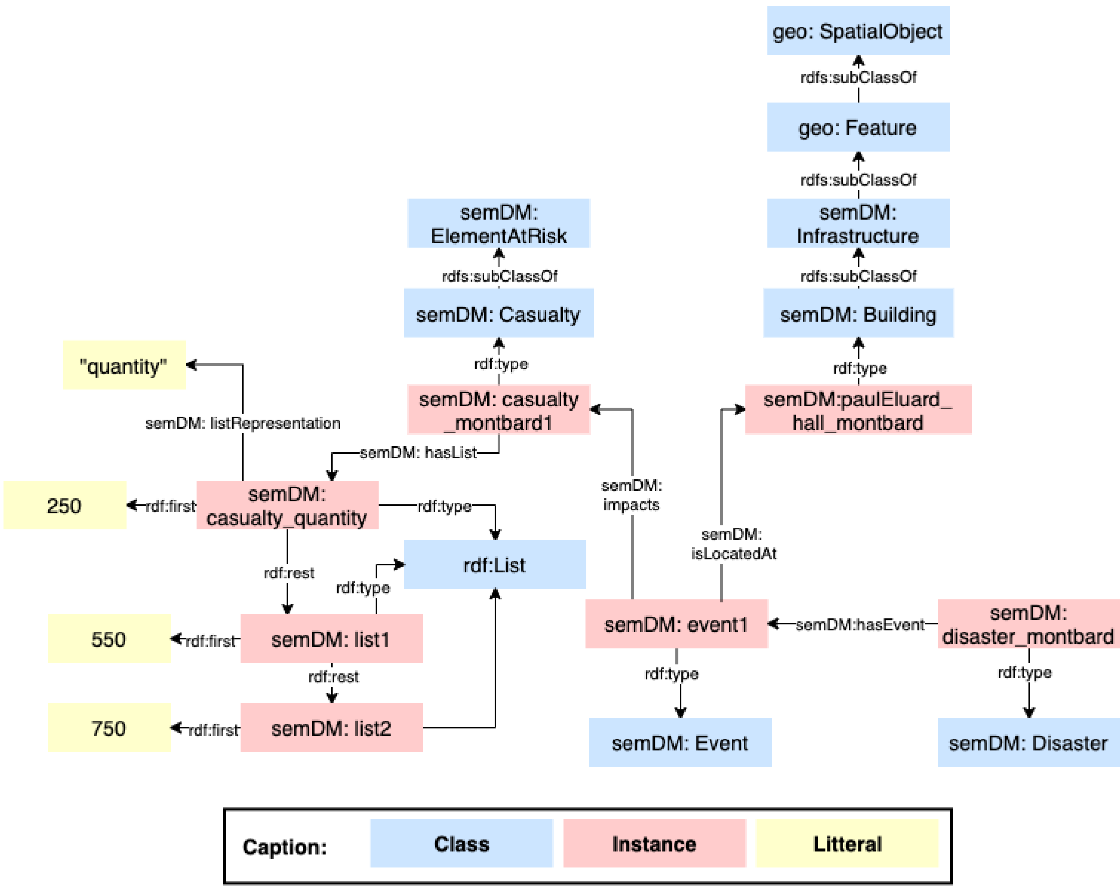


Figure 7.5: Disaster modeling composed of an event with three configurations of casualties quantity

[6]W3C Recommendation link for RDF List

7.2.1.3 NOVI Plan Modeling

As explained in section 5.4, the disaster management system's is defined through at least a plan to assess, which is the NOVI plan for this case study, the description of its associated service, role, tasks, and its sub-procedures with their tasks or actions. The description of an organization allows the definition of the stakeholders intervening in the disaster management system.

Organizations and their resources in terms of roles The case study concerns four organizations of Montbard: the municipal cuncil (*semDM:montbard_municipal_cuncil*), the fire brigade (*semDM:montbard_SDIS*), the ambulance organization (*semDM:montbard_SMUR*), the organization of Emergency Medical Service (EMS) (*semDM:montbard_hospital_center*). Each of these organizations provides some roles. The municipal cuncil provide a director of relief operations (*semDM:DOS*). The fire brigade provides a commander of relief operations (*semDM:COS*), 28 firemen, and 17 fireman officers. The ambulance organization provides 11 ambulances. The organization of Emergency Medical Service provides a director of medical relief (*semDM:DSM*) and 19 physicians. Figure 7.6 illustrates the relationships between organizations and their resources in terms of roles.

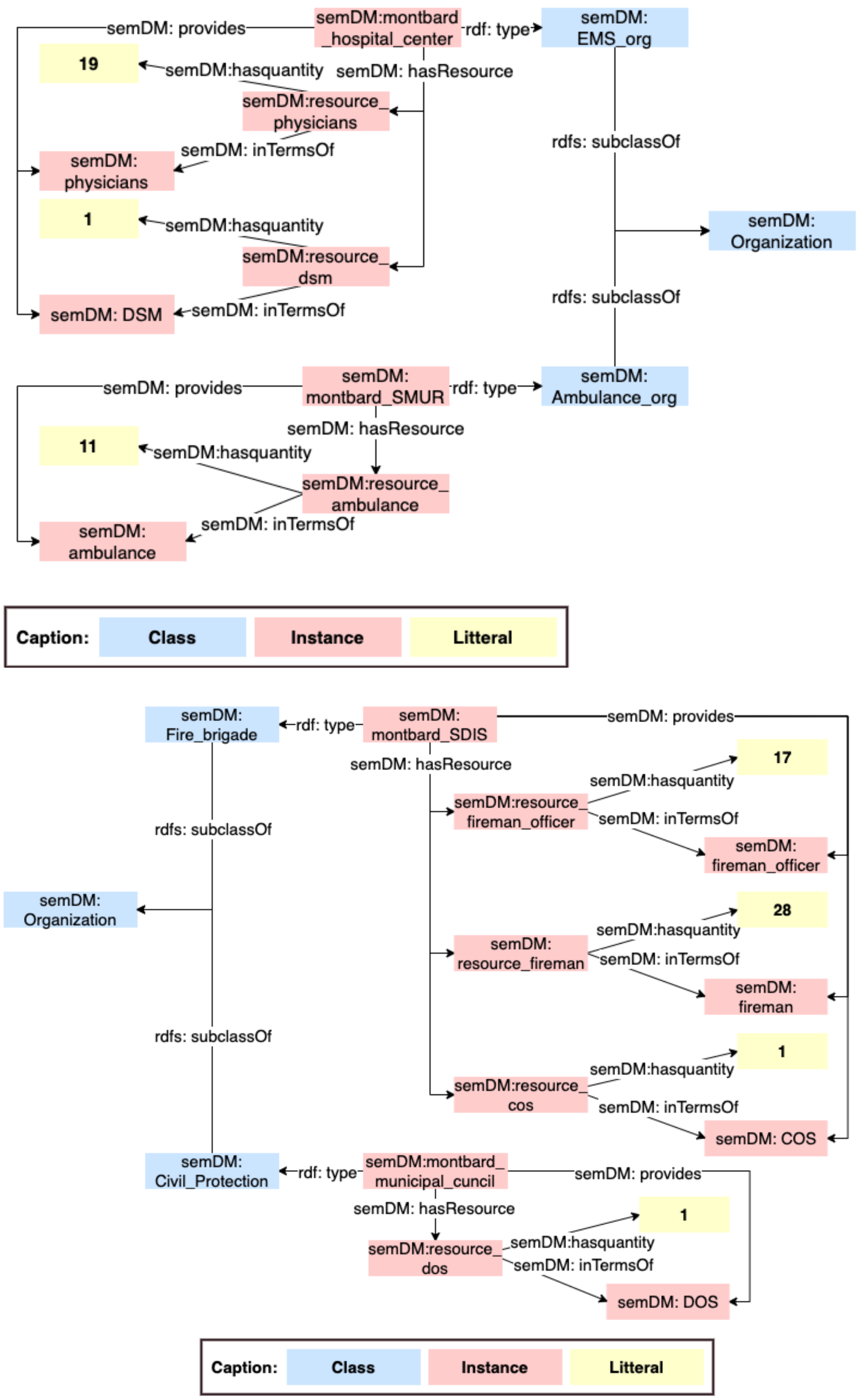


Figure 7.6: Modeling of organizations and their resources in terms of role.

Roles and their service The roles intervening in the NOVI plan are own by the organizations previously presented. Among these roles, the director of relief operations (*semDM:DOS*) aims at coordinating the NOVI plan. Therefore, it delivers the service that follows the NOVI plan, which is represented through the individual *semDM:multiple_victim_management*. The commander of relief operations (*semDM:COS*) aims at managing the rescue of victims on-site and delivers the service *semDM:multiple_victim_rescue*. This role manages the fireman officers, that deliver the service *semDM:collection_noria_management*. The fireman officers manage firemen that delivers the service *semDM: collection_noria*. The director of medical relief (*semDM:DSM*) aims at managing the coordination between the advanced medical post and hospitals. This role delivers the service *semDM:multiple_victim_emergency*. It consists in managing ambulances and physicians. Ambulances deliver the service *semDM:evacuation_noria*, whereas physicians deliver the services *semDM:amp_management* and *semDM:hospital_emergency_management*. Figure 7.7 illustrates the relationships between roles and services.

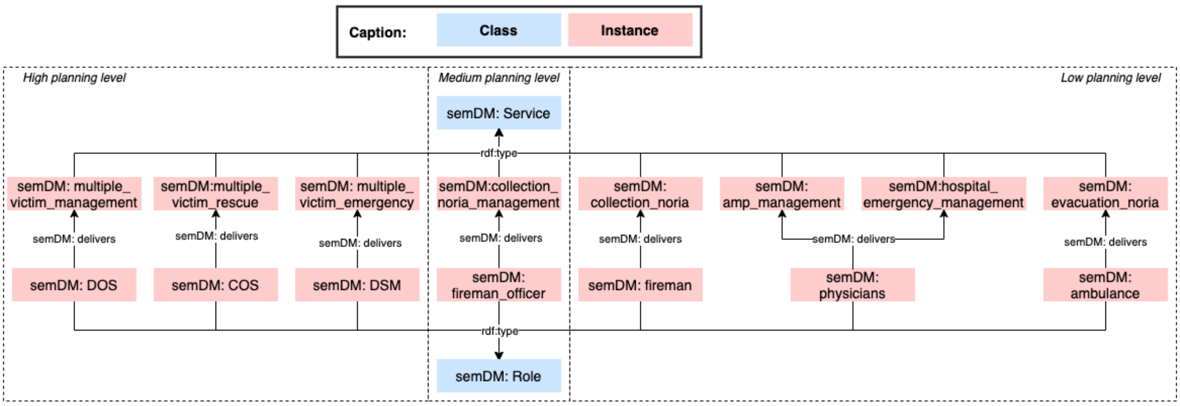


Figure 7.7: Modeling of roles and their associated services

Services and procedures As presented in section 5.3.1, the concept *Service* follows procedures and achieves tasks. Therefore, it makes the intermediate between the different levels of planning. Figure 7.8 shows the association between services and their procedure.

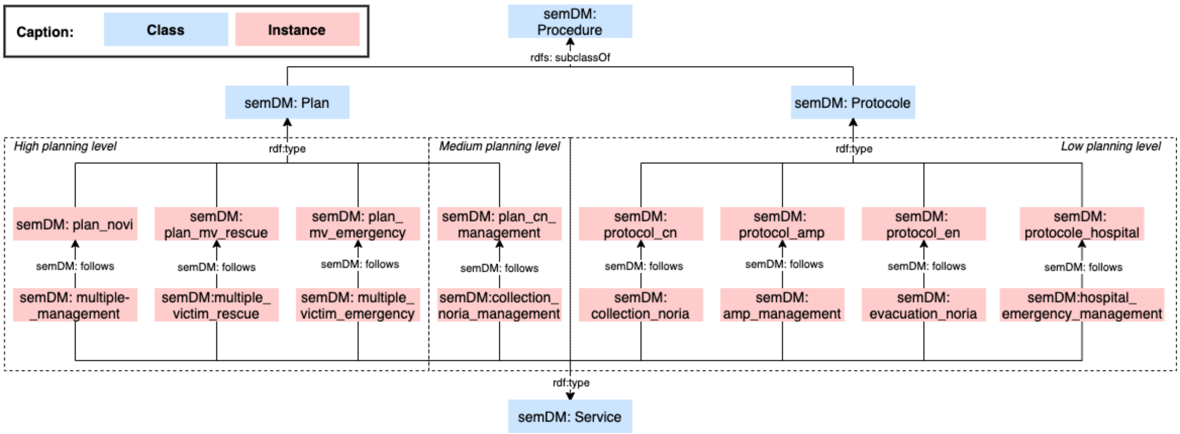


Figure 7.8: Modeling of services and their associated procedures.

The primary service of this case study is the service of multiple-victim management. This service follows the NOVI plan, which is composed of two tasks *semDM:task_mv_rescue* and *semDM:task_mv_emergency*. These tasks are achieved by the services *semDM:multiple_victim_rescue* and *semDM: multiple_victim_emergency*, respectively. These services follow the plans *semDM:plan_mv_rescue* and *semDM:plan_mv_emergency*, respectively. On the one hand, the plan *semDM:plan_mv_rescue* has one task *semDM:task_cn_management*, achieved by the service *semDM:collection_noria_management* delivered by fireman officers. This service follows the plan *semDM:plan_collection_noria* that has one task *semDM:task_collection_noria*. This task is achieved through the service *semDM:collection_noria* delivered by firemen and following the protocol *semDM:protocol_cn*. This task requires a resource in terms of fireman, with a proportion of 100%. On the other hand, the plan *semDM:plan_mv_emergency* has three tasks *semDM:task_amp_management*, *semDM:task_en*, *semDM:task_he_management*. The task *semDM:task_en* is achieved through the service *semDM:evacuation_noria* delivered by ambulances and following the protocol *semDM:protocol_en*. This task requires a resource in terms of ambulance, with a proportion of 100%. The task *semDM:task_amp_management* is achieved through the service *semDM:amp_management* delivered by physicians and following the protocol *semDM:protocol_amp*. The task *semDM:task_he_management* is achieved through the service *semDM:hospital_emergency_management* delivered by physicians and following the protocol *semDM:protocol_hospital*. When the NOVI plan is triggered, in general, the emergency service of the hospital follows the white plan [7]. However, to stay on a simple example, we have modelled a simple protocol composed of a care action instead of this white plan. The tasks *semDM:task_amp_management* and *semDM:task_he_management* require a resource

[7]White plan: `https://cutt.ly/6tp10ad`

in terms of physicians, but without defined quantity neither proportion. Figure 7.9 presents the relationship between services, plans, and tasks that allows the representation of the different levels of planning. All these services serve an element at risk with *semDM:Casualty* as type. However, they do not target the same state of this element at risk. The services *semDM:multiple_victim_rescue*, *semDM:collection_noria_management*, and *semDM:collection_noria* targets the state *semDM:inDanger*. The services *semDM:amp_management*, *semDM:evacuation_noria*, and *semDM:hospital_emergency_management* targets the states *semDM:rescued*, *semDM:caredInAMP*, and *semDM:evacuated*, respectively. The service *semDM:multiple_victim_emergency* targets these three last states. A service aims at producing a new state of the element at risk. In this case study, the services *semDM:collection_noria*, *semDM:amp_management*, *semDM:evacuation_noria*, and *semDM:hospital_emergency_management* produce the states *semDM:rescued*, *semDM:caredInAMP*, *semDM:evacuated*, and *semDM:caredInHospital* respectively.

Casualty Casualties are the element at risk at the heart of this case study. All the services previously presented serve an individual of *semDM:Casualty*. However, each of them targets different states of casualty. Therefore, casualty modeling is linked to a set of states. There are firstly those targeted by services, which are a kind of processing states. Secondly, there are health states that play an important role in the actions of a protocol. As explained previously in section 7.1.1, casualties are classified in five health states, which are *semDM:dead* for Emergency Overwhelmed/Death, *semDM:EU* for Extreme Emergency, *semDM:U1* for Critical Injury, *semDM:U2* for Serious Injury, and *semDM:U3* for minor injury.

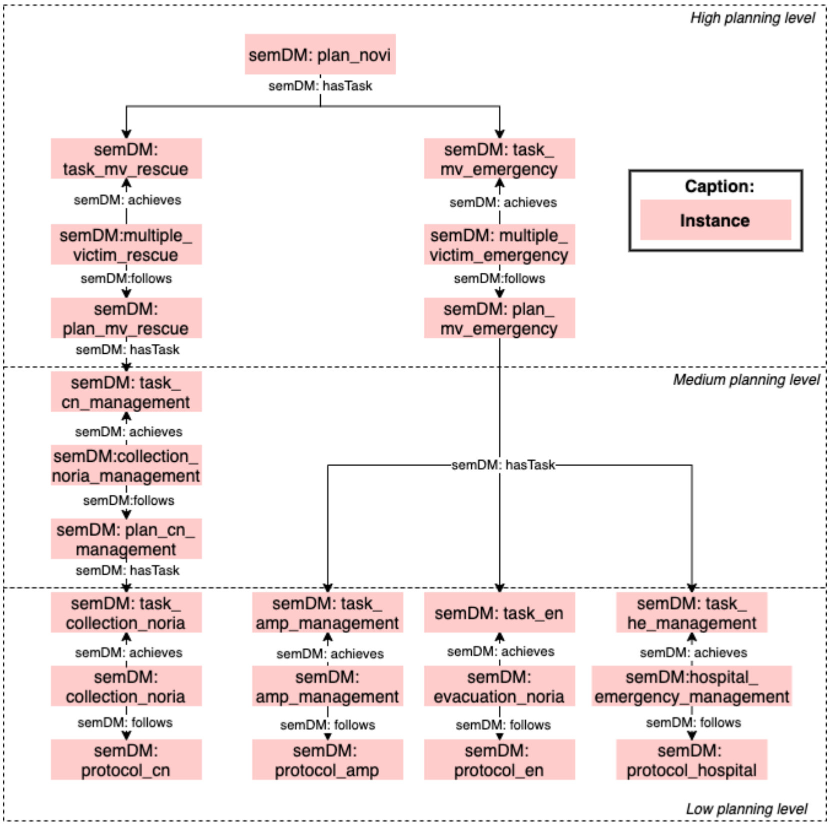
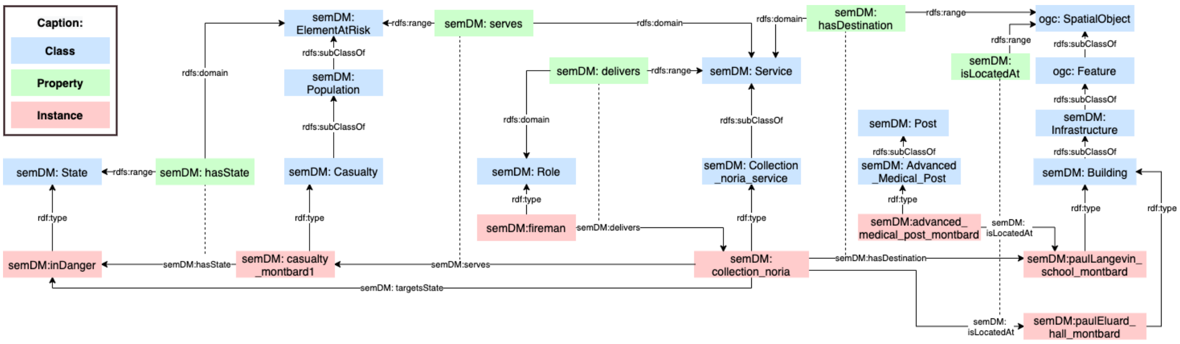
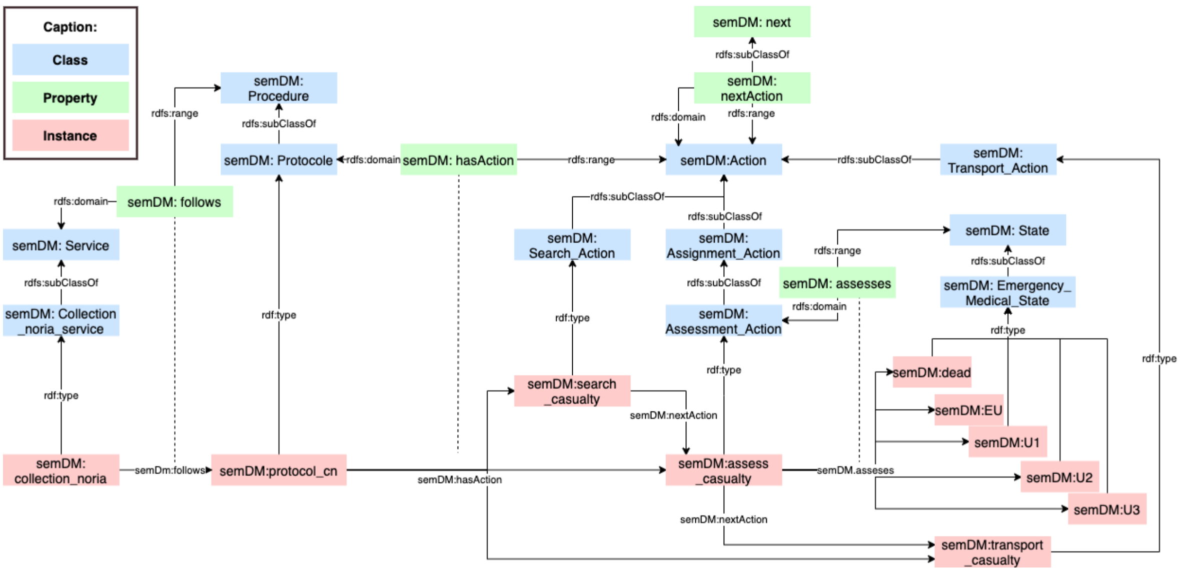


Figure 7.9: Modeling of NOVI plan and its sub-procedures.

Protocol for collection noria service Let us take the example of the *semDM:collection_noria* service and its protocol *semDM:protocol_cn* to explain in detail a protocol modeling and shows the role of casualty's health state. Figure 7.10 illustrates this example. This service is delivered by the role of fireman and locates at the Paul Eluard hall (*semDM:paulEluard_hall_montbard*). It targets casualties in danger. As explained previously in section 7.1.2, the Paul Langevin school represented by the individual *semDM:paulLangevin_school_montbard* is the location of the advanced medical post. The *semDM:collection_noria* service has the individual *semDM:paulLangevin_school_montbard* as destination since it aims at evacuating the casualties to the advanced medical post,. Figure 7.10a shows the modeling of this service. The protocol *semDM:protocol_cn* is composed of three ordered actions *semDM:search_casualty*, *semDM:assess_casualty*, and *semDM:transport_casualty*. The action *semDM:assess_casualty* allows the assessment of the health state of a casualty among the five health states (*semDM:dead*, *semDM:EU*, *semDM:U1*, *semDM:U2*, and *semDM:U3*). Figure 7.10b shows the modeling of this protocol.



(a) Description of role delivering the service collection noria and the element at risk that it serves



(b) Description of protocol followed by the service collection noria

Figure 7.10: Modeling of the service collection noria and its associated protocol.

After the disaster management system's modeling by the integration of interpreted data and user adding, the reasoning is applied to the SemDM ontology to deduce geospatial relations. The result of this reasoning step is explained in the next section.

7.2.2 Reasoning on disaster management model according to an event

The adding of a disaster scenario description into the knowledge base is done through the definition of events. An event is located and has an impact. The geospatial reasoning based on rules previously presented in code 6.1 and 6.2 of section 6.1.3, results in determining that the governmental echelon of the disaster is Montbard. This result comes from its event's location. Figure 7.11 illustrates the result of this inference.

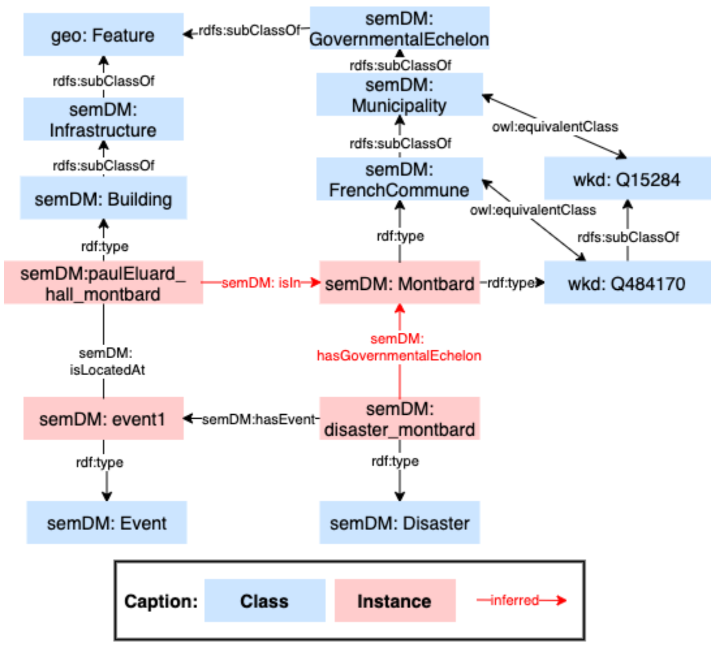


Figure 7.11: Result illustration of reasoning on an event representation.

This section has explained the different processes that intervene in the modeling of the disaster management system. This disaster management model is then used as an input for the design of simulations. The next section explains the simulation modeling processes.

7.3 Simulation modeling

The simulation modeling is the second step of the method (c.f. section 4.3), as illustrated in Figure 7.12. It is designed through reasoning and results in two simulation models, as explained in section 6.2 of the previous chapter. The first generated model is the conceptual model of the simulation, which is platform-independent. It results directly from the transformation of the disaster management model. This simulation model can be thus easily shareable among the multi-agent simulation community and offers the possibility to be implemented according to different simulation platforms. The second simulation model is the programming model of simulation, which is specific to the GAMA platform. It results from the transformation of the conceptual model of simulation and some knowledge on the disaster management model. It aims at being used to generate the simulation program.

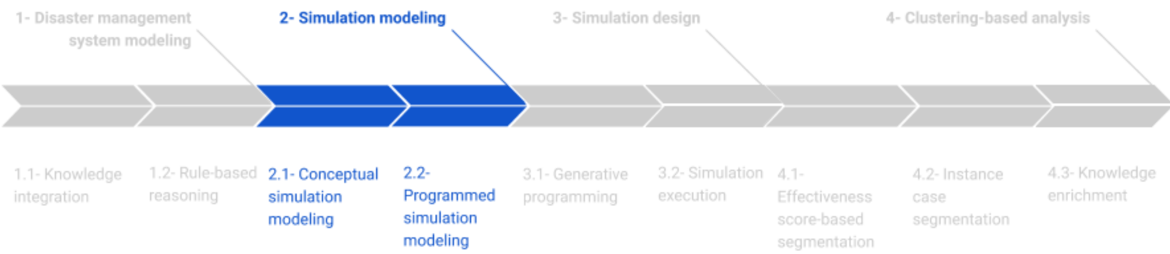


Figure 7.12: Second step of the application: Simulation modeling processes

7.3.1 Conceptual simulation modeling from disaster management case study

The information and knowledge previously integrated enable a disaster management model to be defined. To simulate this model, it is necessary to create a conceptual simulation model representing the disaster management model. The process of generating the conceptual model is carried out by rule-based reasoning, as presented in section 6.2.1 of the previous chapter. This section aims at showing the result of the rule-based reasoning for conceptual simulation modeling.

Conceptual simulation modeling As explained in section 6.2.1 of the previous chapter, the individual of conceptual simulation model is generated from a disaster description through rules presented in Code 6.3 and 6.4. The application of these rules on the disaster described Figure 7.11, generates an individual of *semDM:DM_model* adapted with the disaster *semDM:disaster_montbard*. Code 6.3

also generates an environment's individual *semMAS:env_59s37z283*, which represents *semDM:montbard* and is the environment of the created conceptual simulation model. Code 6.4 has allowed the definition of *semDM:casualty_quantity* as parameter of the model. Figure 7.13 illustrates the results of these rules to generate the conceptual simulation model.

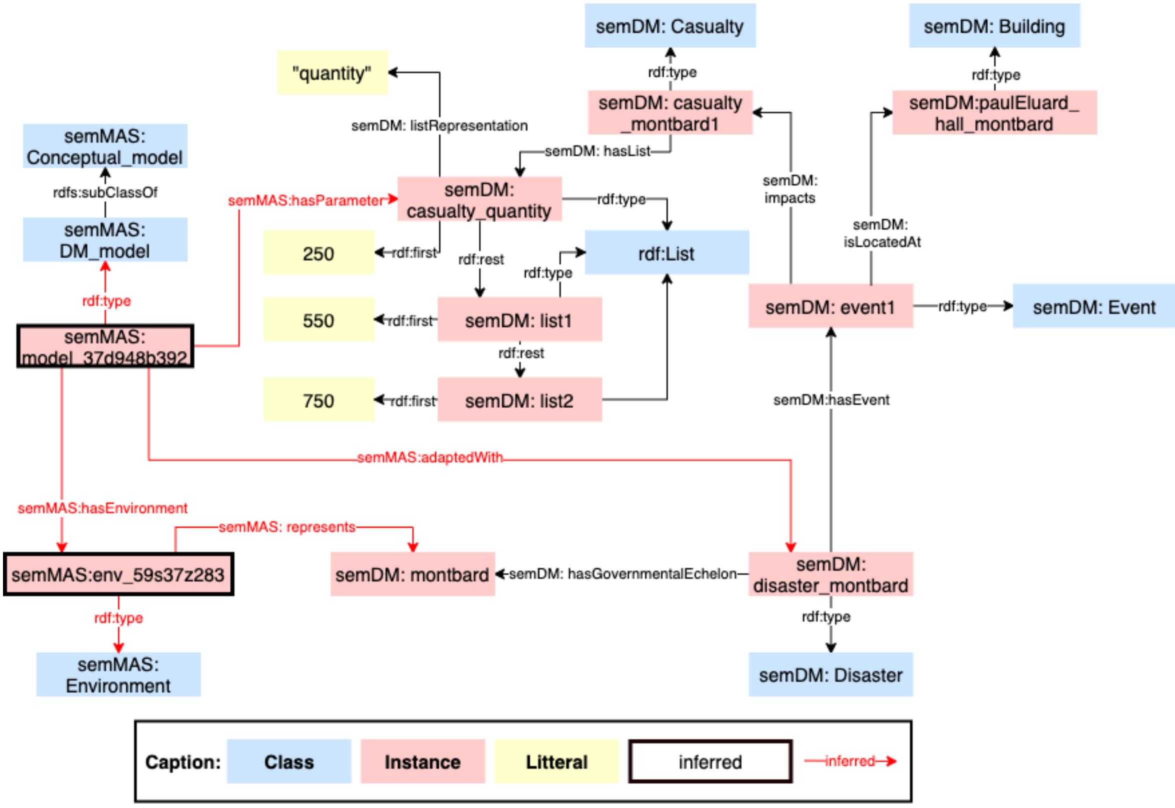


Figure 7.13: Conceptual simulation modeling resulting from rule-based reasoning

Environment modeling The model's environment is characterized by the artifacts representing buildings and roads of the governmental echelon impacted by the disaster event. Code 6.6 in the previous chapter has shown the rules allowing the design of the building artifacts. Figure 7.14 illustrates the results of this rule apply to two examples of buildings belonging to this case study: the Paul Langevin school and the Paul Eluard hall. The reasoning defines these two buildings as members of the artifact *semMAS:building*, which is contained in the simulation model's environment.

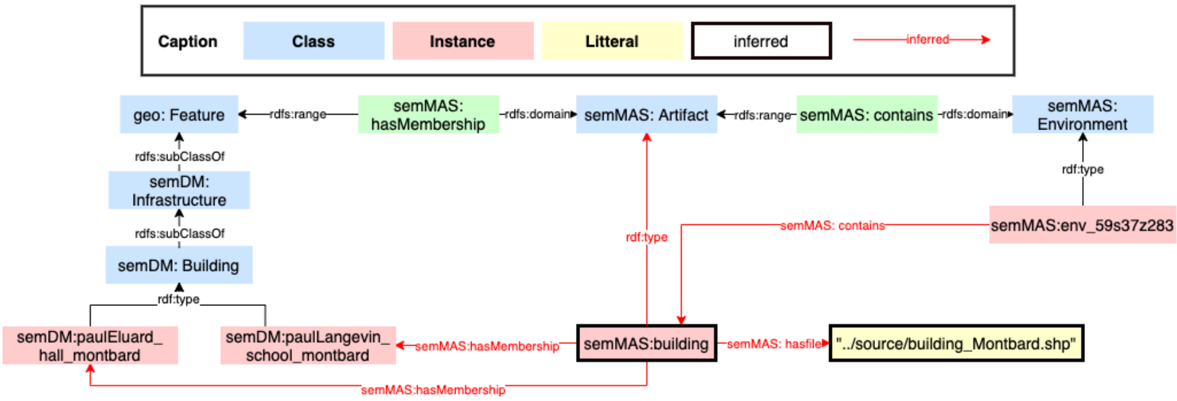


Figure 7.14: Building artifact modeling resulting from rule-based reasoning

It has been explained in section 6.2.1 that a similar rule to the one represented by Code 6.6 exists for roads. Figure 7.15 illustrates the application results of the rule defined for roads to two examples of roads belonging to the integrated data presented section 7.2.1.1: "Rue des Roches" and "Route de Châtillon". The reasoning defines these two roads as members of the artifact *semMAS:road*, contained in the simulation model's environment.

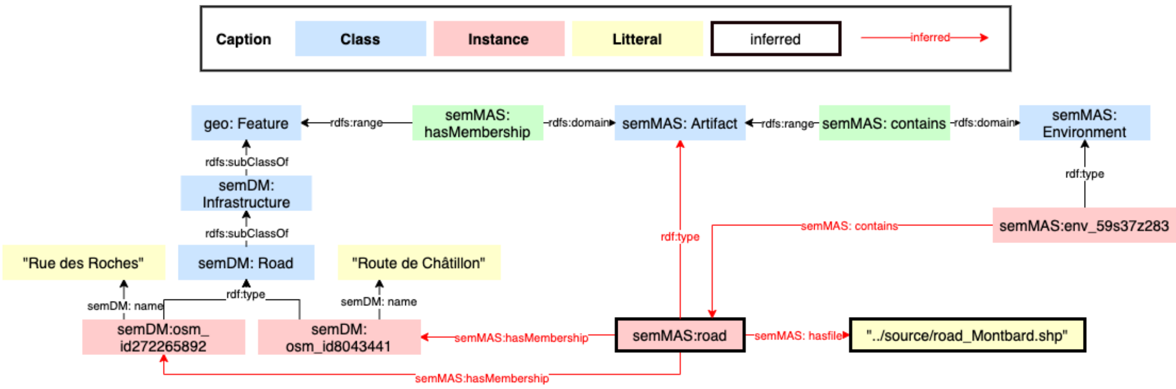


Figure 7.15: Road artifact modeling resulting from rule-based reasoning

Reactive actor agent modeling Actor agents are reactive agent defined through a role that delivers services that follow only protocol. Let us take the example of the role *semDM:fireman* to illustrate the design of an actor agent. The rule presented in Code 6.8 generates an individual of *semMAS:Actor_agent* that represents the role *semDM:fireman*. This individual called *semMAS:agent_fireman* has a newly generated behavior *semMAS: collection_noria_behavior* representing the service *semDM: collection_noria*. This agent is in the number of 28, which is inferred from the quantity of the role *semDM:fireman* owned by *semDM: montbard_SDIS*. It is contained in the environment associated with the simulation model adapted with the disaster linked to the element at risk that serves the service *semDM: collection_noria*. Figure

7.16 illustrates the results of the rule application presented in Code 6.8, on the role of *semDM:fireman*.

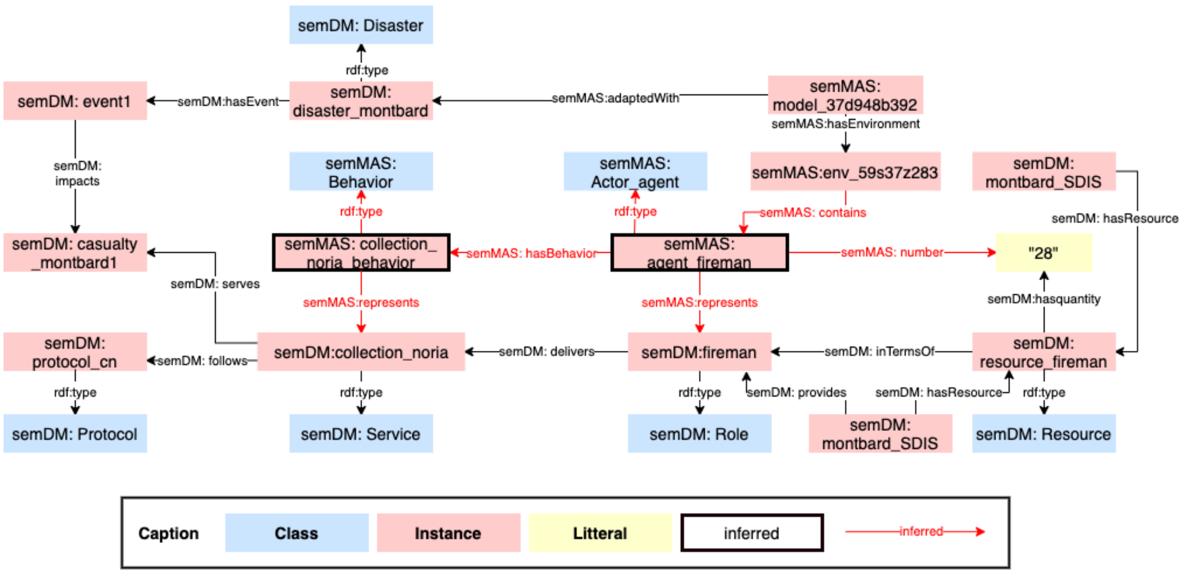


Figure 7.16: Results of the actor agent modeling for the role *semDM:fireman*

The actor agents' behavior have actions that represents the actions belonging to the protocol followed by the service that it represents. Let us take the example of the action *semDM:transport_casualty* that belongs to the *semDM: protocol_cn*. The application of the rule presented in Code 6.8 generates *semMAS: transport_756daed*, an action of the behavior *semMAS: collection_noria_behavior* representing the protocol's action *semDM:transport_casualty*. This rule also defines the values of the properties *semMAS:what*, *semMAS:where*, and *semMAS:requiresObjectives*. The application of the rule presented in Code 6.9 of the previous chapter generates a status that is required by the action *semDM:transport_casualty*. Finally, the fourth priority rule discussed in previous chapter for the case of no successive action, associates the status *semDM:rescued* as a resulting status. This status is the status produced by the service *semDM:collection_noria*.

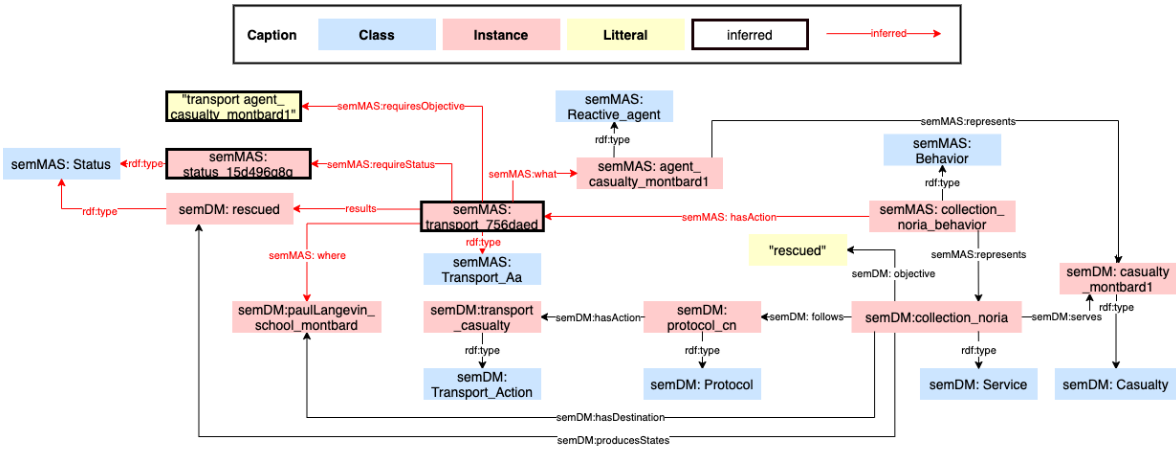


Figure 7.17: Results of the agent's action modeling for the action *semDM:transport_casualty*

Cognitive agent modeling Cognitive BDI agents are defined from a role that delivers services that follow only plans. Let us take the example of the role *semDM:fireman_officer* to illustrate the generation of BDI agents. The rule presented in Code 6.11 generates the BDI agent *semMAS: agent_fireman_officer* that represents *semDM:fireman_officer*. This BDI agent is in the number of 17, which is inferred from the quantity of the role *semDM:fireman_officer* owned by *semDM: montbard_SDIS*. It is contained in the environment associated with the simulation model adapted with the disaster linked to the element at risk that serves the service *semDM: collection_noria_management*. Figure 7.18 illustrates the results of the rule application presented in Code 6.11, on the role of *semDM:fireman_officer*.

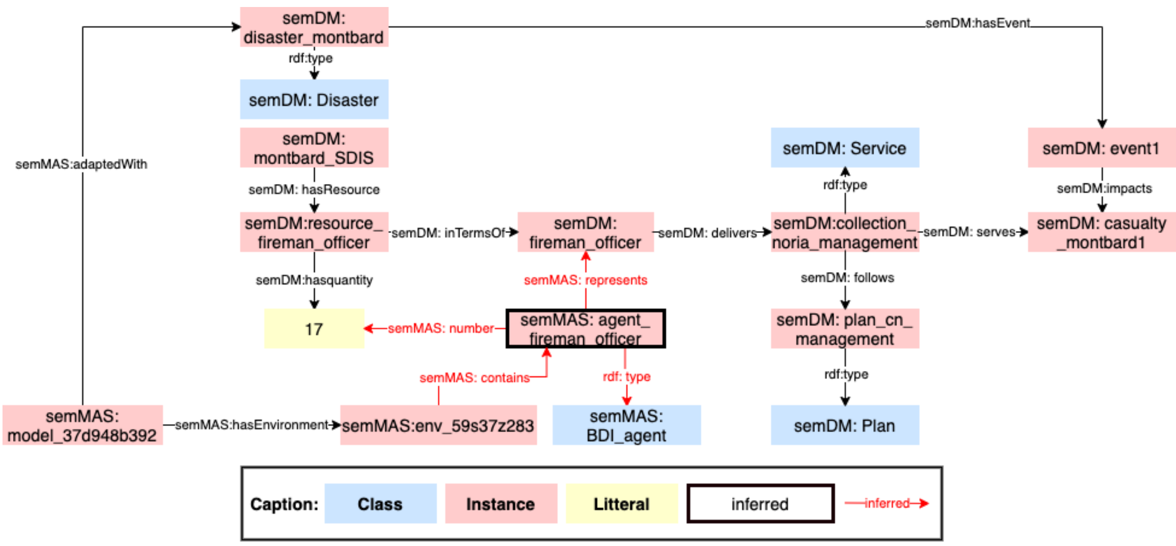


Figure 7.18: Results of the BDI agent modeling for the role *semDM:fireman_officer*

A BDI agent is characterized through desires and plans. Let us continue the example of the BDI agent *semMAS: agent_fireman_officer*. The rule presented in Code

6.12 generates the desire *semMAS: collection_noria_management_desire* that represents the service *semDM:collection_noria_management* and the agent's plan *semMAS: bdi_plan_cn_management* that represents the plan *semDM: plan_cn_management*. These two new individuals are assigned to the agent *semMAS: agent_fireman_officer* through the properties *semMAS:hasDesire* and *semMAS:hasPlan*, respectively. The agent's plan *semMAS: bdi_plan_cn_management* has an intention, which corresponds to the desire *semMAS: collection_noria_management_desire*. Figure 7.19 illustrates the results of desire and plan modeling for the agent *semMAS: agent_fireman_officer*.

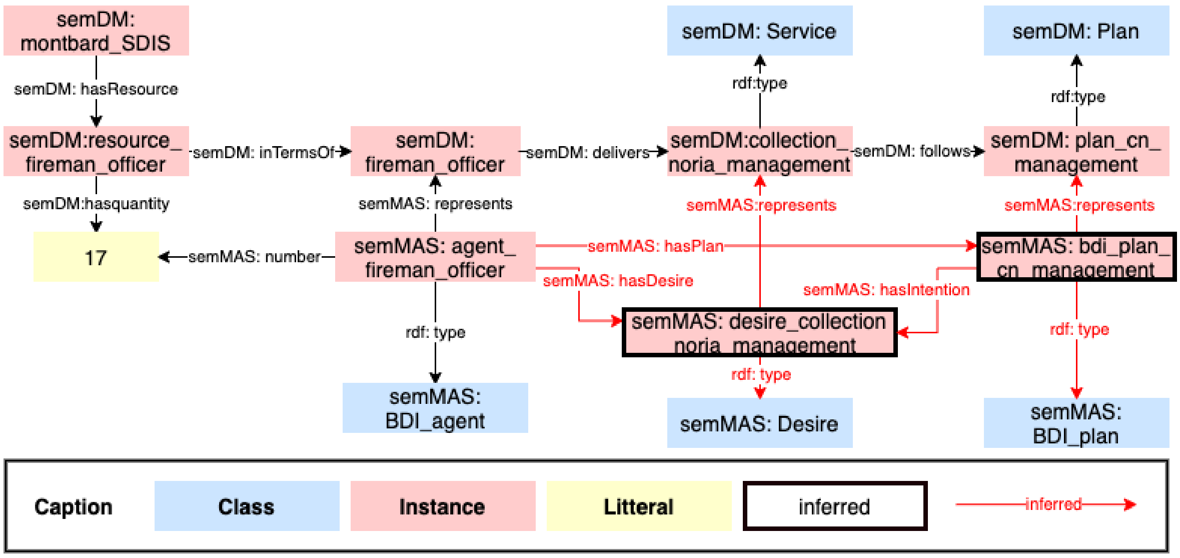


Figure 7.19: Results of desire and plan modeling for the agent *semMAS: agent_fireman_officer*

The rule presented in Code 6.13 generates the belief *semMAS:belief_cn_management* that represents the task *semDM:task_cn_management* and defining the agent *semMAS:agent_fireman_officer* as a *semMAS:Manager_agent*. This new belief is assigned to the agent through the property *semMAS:hasBelief*. Figure 7.20 illustrates the results of belief modeling for the agent *semMAS: agent_fireman_officer*.

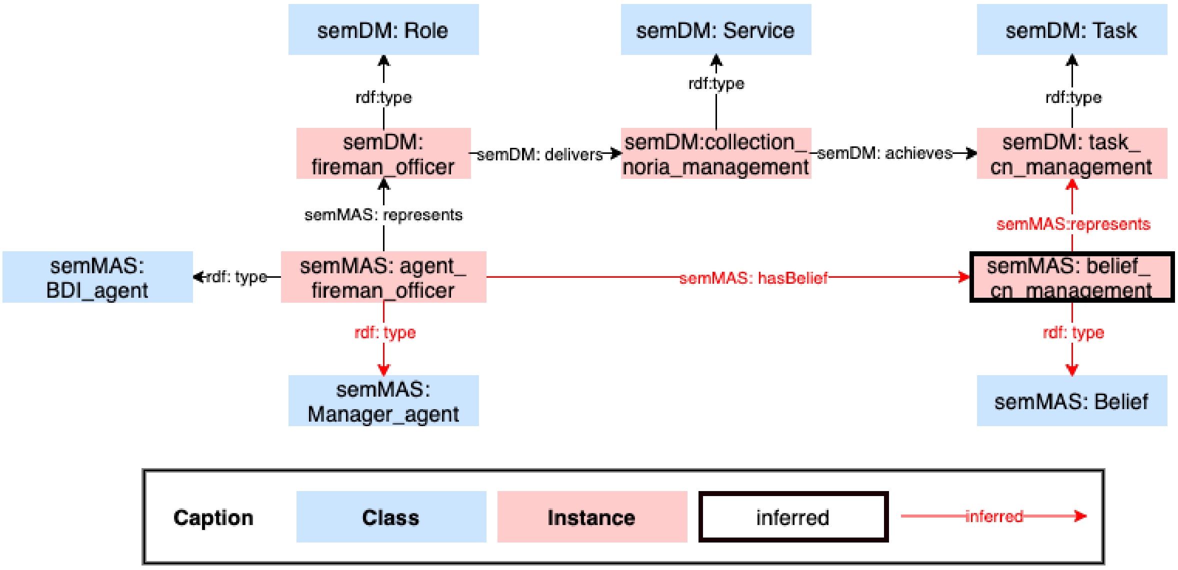


Figure 7.20: Results of belief modeling for the agent *semMAS: agent_fireman_officer*

The rule presented in Code 6.15 allows the identification of agents known by a BDI agent. In the case of the agent *semMAS: agent_fireman_officer*, this rule defines *semMAS: agent_fireman* as known by this agent. This relation is expressed through the property *semMAS:knows*. Figure 7.21 illustrates the results of agents known by the agent *semMAS: agent_fireman_officer*.

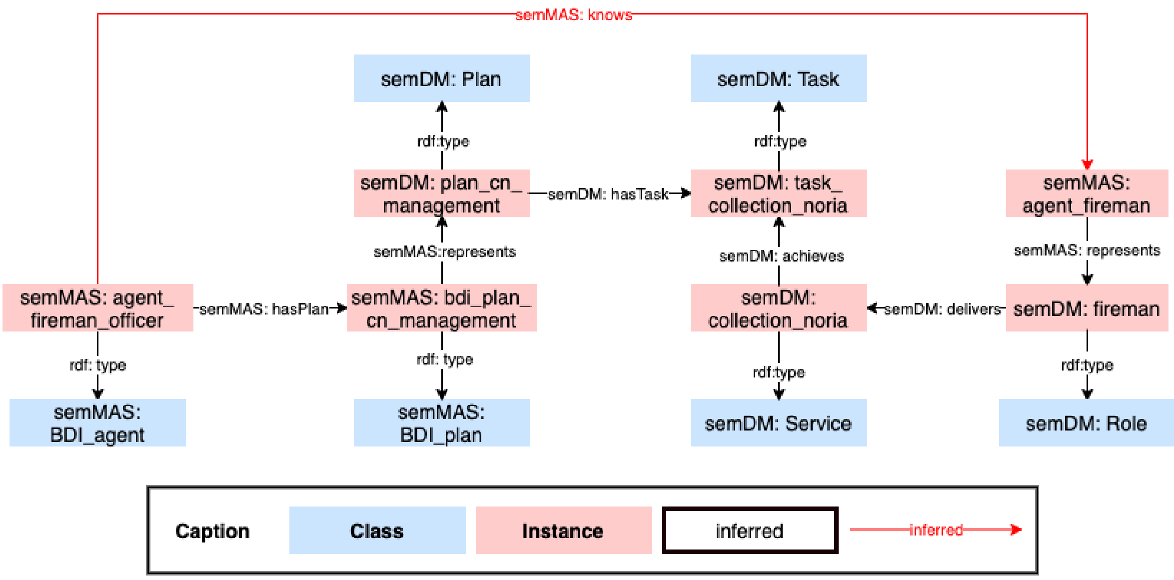


Figure 7.21: Modeling results of agents known by the agent *semMAS: agent_fireman_officer*

7.3.2 Program modeling from the conceptual simulation model

A representation of the programmed model can be generated in the SemMAS ontology, thanks to the definition of concepts specifying the GAML programming according to multi-agent simulation concepts, as presented in section 6.2.2.

The representation of the programmed model is obtained from reasoning on the knowledge base. On the one hand, the reasoning is applied through the inference engine provided by Apache Jena [8] to infer implicit knowledge from explicit knowledge. This reasoning is applied without rules. For example, this first reasoning infers that individuals of *Agent* become *Species* from the *owl:equivalentClass* between these two concepts and that actions composing behaviors of a reactive agent become its reflex from the definition of a sub-property chain of the property *hasReflex*. Figure 7.22 illustrates the result of such inference on the modeling of the agent *semMAS:agent_fireman* and the action *semMAS: transport_756daed*. This action is an individual of the concept *semMAS: Transport_Aa*, whose the OWL specification has been presented in section 5.3.2.2 and the inference of skills associated to its individuals has been explained section 6.2.2.

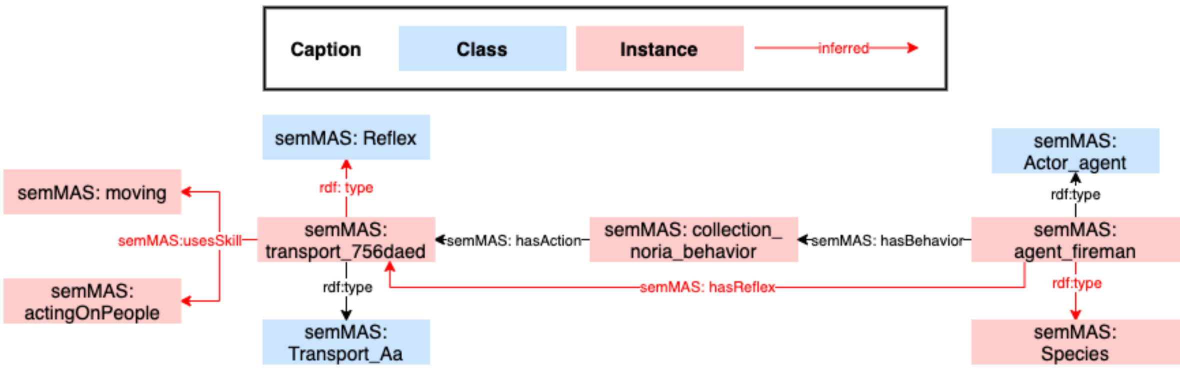


Figure 7.22: Result of axiomatic-based inference on the modeling of the agent *semMAS:agent_fireman* and the action *semMAS: transport_756daed*

On the other hand, second reasoning is applied to infer new knowledge from SHACL rules. It uses the engine reasoner provided by Apache Jena to execute SHACL rules. This second reasoning applies rules as those presented in Code 6.16 and 6.17 of the section 6.2.2. The rule presented in Code 6.16 and applied to the modeling of the agent *semMAS:agent_fireman* and the action *semMAS: transport_756daed*, defines that this agent has the skills *semMAS: moving* and *semMAS: actingOnPeople*. Figure 7.23 illustrates this result.

[8]Apache Jena: `https://jena.apache.org/`

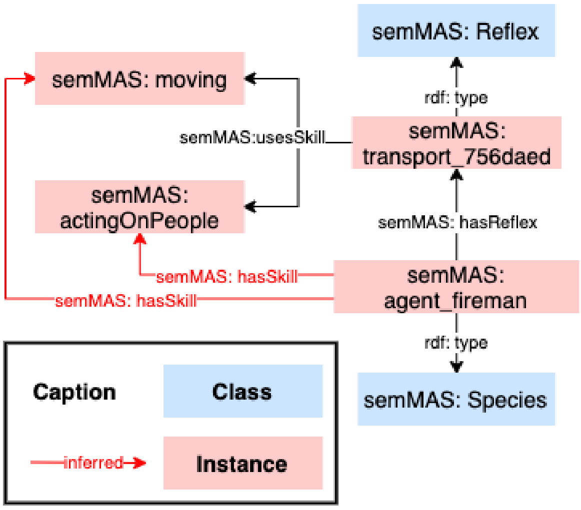


Figure 7.23: Result of rule-base inference on skills for the agent *semMAS:agent_fireman* and the action *semMAS: transport_756daed*

The rule presented in Code 6.17 and applied to the modeling of the agent *semMAS:agent_fireman*, defines that this agent has any location in *semDM: montbard_fire_station*. Figure 7.24 illustrates this result.

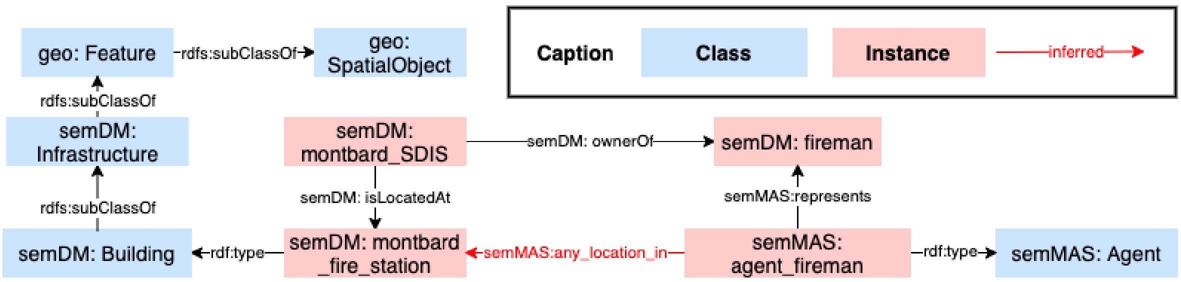


Figure 7.24: Result of rule-base inference on the location of the agent *semMAS:agent_fireman*

In GAML, the *semMAS:SimpleBDI* that represents BDI agents are a part of the programmed model that must be represented. The rule presented in Code 6.19 defines the *semMAS:SimpleBDI* agents. These agents have some specificities as rules, which are generated through the rule presented in Code 6.20. The application of these rules on the agent *semMAS:agent_fireman_officer*, define it as a *semMAS:SimpleBDI* and assign to it a new rule based on its belief. Figure 7.24 illustrates the result of the rule presented in Code 6.19 in blue and the result of the rule presented in Code 6.20 in red.

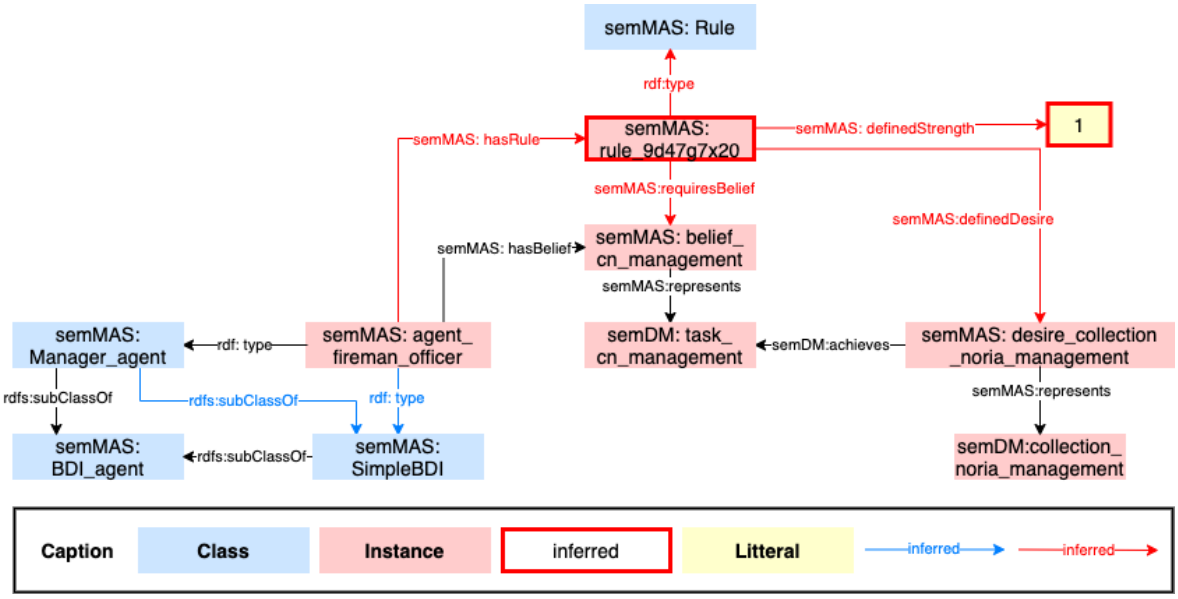


Figure 7.25: Result of rule-base inference on the concept *semMAS:SimpleBDI* specificities

The programmed model in GAML represented into the knowledge base is the base of the generative programming presented in the next section. This generative programming aims at producing the simulation program corresponding to the GAML programming model represented in the knowledge base.

7.4 Simulation design

The third step of the methodology corresponds to the programming and execution of simulations. Figure 7.26 shows this step of processing in the overall system processing.

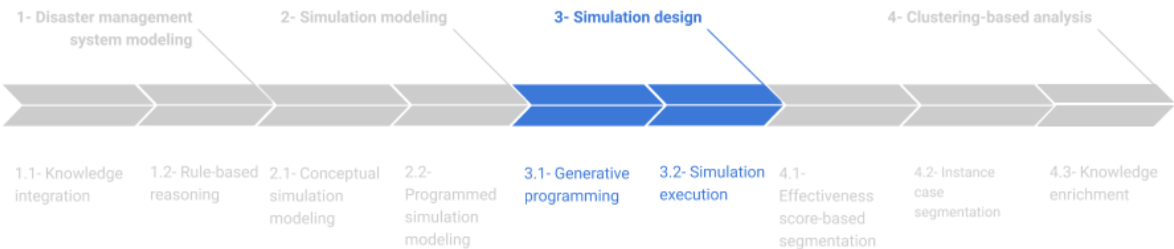


Figure 7.26: Third step of the application: Programming and execution of simulations

7.4.1 Generative programming

The simulation programming consists of the implementation of the simulation model and the implementation of simulation experiments. At a high-level of multi-agent simulation, these implementations correspond to the implementation of two models: the programming model and the experimental model. However, the simulation programming in GAML gathers both the implementation of the simulation model and the experiment model. This section presents the designed simulation model in GAML obtained from its representation into the knowledge base by the generative programming process, presented in section 6.3.1. This GAML model has a structure with five parts, illustrated in Figure 6.10 of the previous chapter:

1. the model definition,
2. the definition of the global species,
3. the definition of regular species representing artifacts,
4. the definition of regular species representing agents, and
5. the definition of experiments.

This section illustrates the result of this process through example, results for each of these parts. For each result of the generative programming, the part of the ontology used as input is presented.

Definition of the simulation model The definition of the model is based on the individual representing the *semMAS: Programmed_model*. The individual in this case study is *semMAS: model_947d83a527*. Figure 7.27 illustrates the results of the first block implementation, corresponding to the model definition.

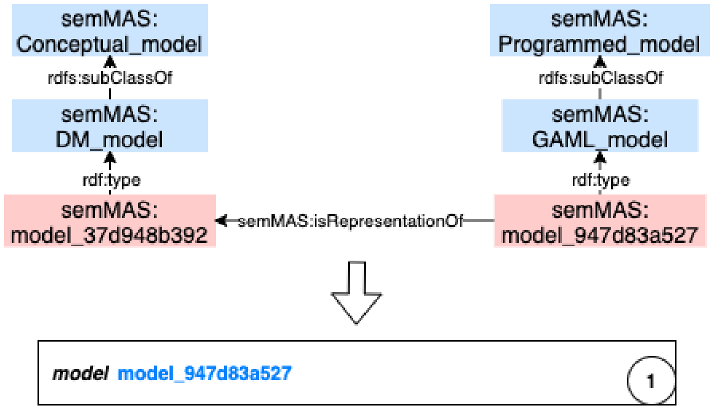


Figure 7.27: Implementation results of the first block of the programmed GAML model

Definition of *global species* As explained in the previous chapter, the global species definition is composed of three parts.

- The first one corresponds to the simulation's variables definition, illustrated in the green frame of Figure 7.29. Let us take the example of the casualty list for the state "in danger". It is represented in SemMAS as illustrated in the green box of Figure 7.28. This representation allows the generative programming to produce a list of casualty, which contains all casualty agents that have the state in danger and which evolve with the casualty state. Then the quantity of casualty in danger is defined according to the size of the list. It exists such a list for each casualties states. Some of them are obtained from an action description. For example, the transport action *semMAS: transport_756daed* described in Figure 7.17, allows the definition of the lists *casualty_montbard1_assessed* and *casualty_montbard1_rescued*. The action *semMAS: transport_756daed* having *semMAS:agent_casualty_montbard1* has object of the property *semMAS: what*, *semMAS:assessed* has object of the property *semMAS:requireStatus*, and *semMAS:rescued* has object of the property *semMAS:results*, means that the states "assessed" and "rescued" are states of a casualty. These states evolve during the simulation and can be observed by other agents or as simulation outputs. Another example of a variable shown in the green box of Figure 7.29 is the graph representing the road network. As illustrated in the light orange box of Figure 7.28, the simulation world is composed of roads. Roads implementation requires defining a graph to allow the move of agents.

- The second part of the global species is the initialization of the components. The light orange box of Figure 7.29 shows the implementation of the road artifact initialization. From the SemMAS ontology part illustrated in the light orange box of 7.28, the artifact road is initialized with its file path. It is then used to initialize the graph. The purple box of Figure 7.29 shows the initialization of two agents : agent_fireman and agent_fireman_officer. These agents are initialized from their modeling in the purple box of Figure 7.29. The agent's number is initialized through the value associated with the property *semMAS:number*. The actor agent is defined with default attribute as target and objective, initialized with the null value. They can have another attribute if the modeling in SemMAS ontology associates values to agents with the property *semMAS:hasAttribute*. Central and manager agent have subordinates, with whom they communicate. In SemMAS ontology, this relation is represented through the property *semMAS:knows*. This relation generates the implementation of a list of the agents known by another.

- The last part corresponds to the definition of the reflex that saves the simulation results. The header of the reflex is completed according to the final condition of the model defined through the property *semMAS:hasFinalCondition*. A condition is represented through three properties: (1) *semMAS:hasOp1*, (2) *semMAS:hasOperator*, and (3) *semMAS:hasOp2* or *semMAS:hasValue*. The third property depends if the condition is based on a value or another individual. The condition can be nested thanks to the property *semMAS:hasOp2* that can link to another condition, allowing composed conditions definition. The dark orange of Figure 7.28 shows a composed condition representation in SemMAS ontology. This representation is used to complete the *when:()* part of the reflex header. Then, the reflex contains a function save taking the list of parameters and observed variables of the simulation. Finally, the process retrieves the file path associated with the model through the property *semMAS:hasOutput* to define the file where saved the results.

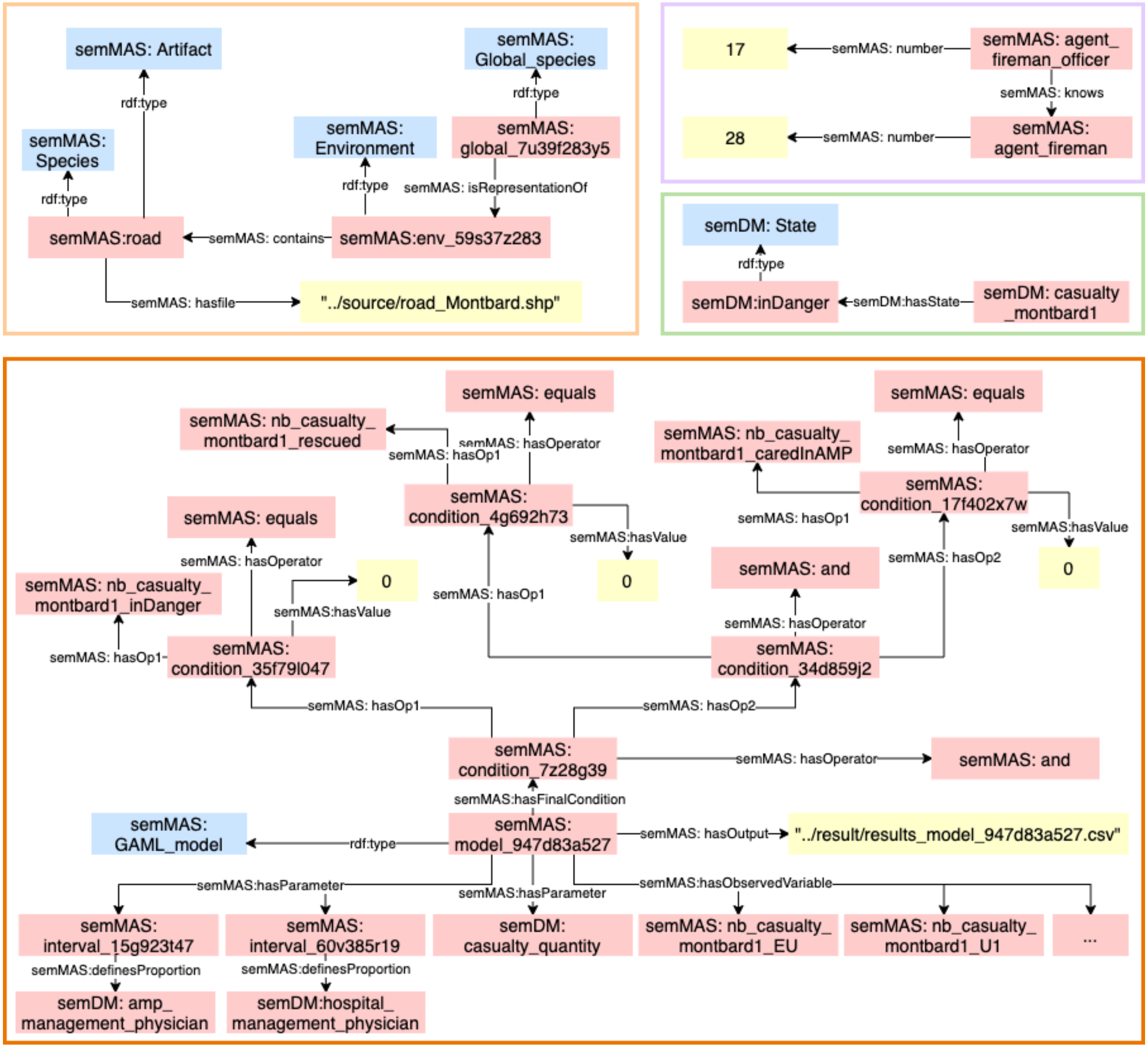


Figure 7.28: Parts of SemMAS ontology for the global implementation

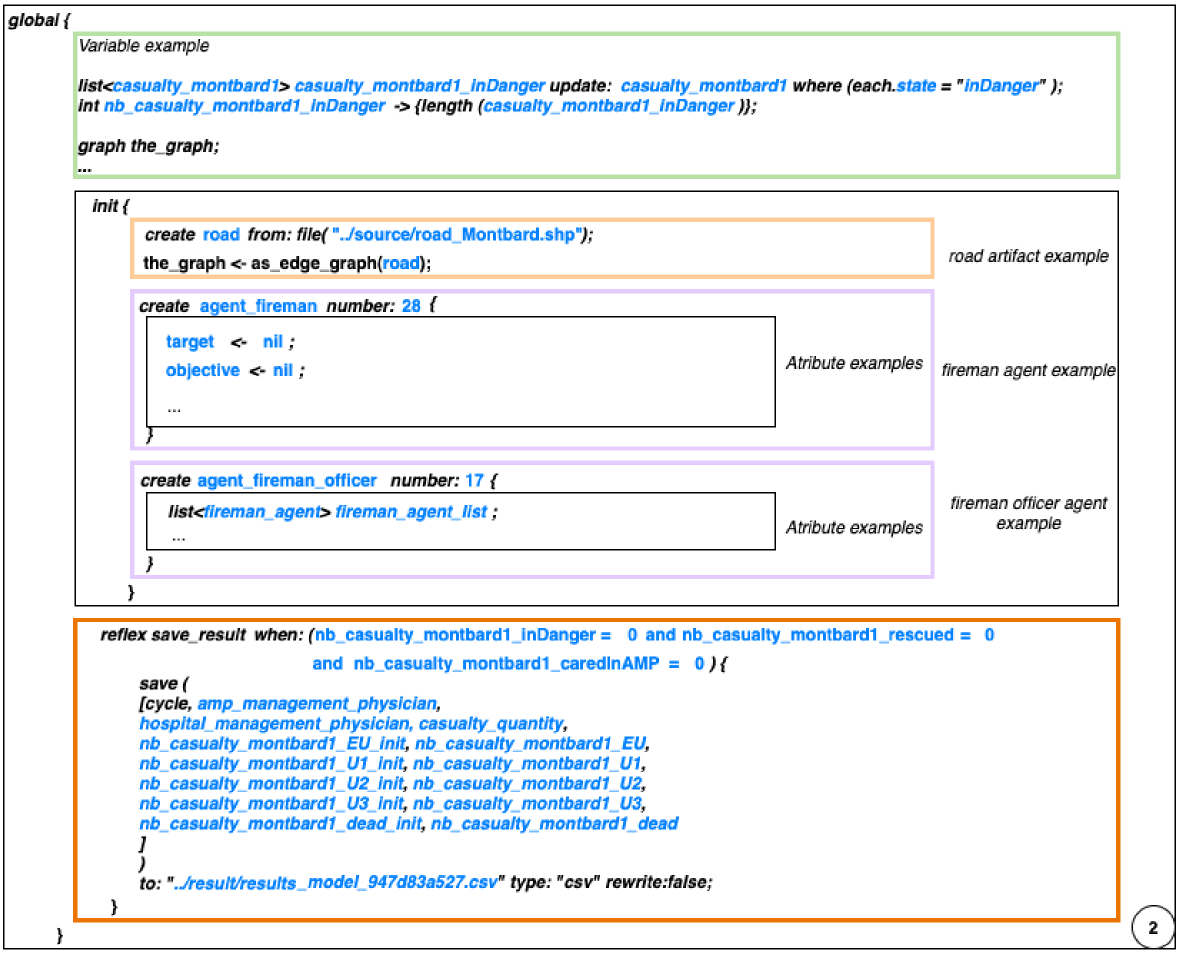


Figure 7.29: Implementation parts result of the second block of the programmed GAML model

Implementation result of *species* representing an artifact The third part of the GAML model corresponds to the definition of species representing the artifacts. The process retrieves all species that are also artifacts to define them. All of them retrieve a fixed set of attributes, which are a type, a label, and a status.

```
species road {
        String type;
        String label;
        String status;
    }                  (3)
```

Figure 7.30: Example of implementation results of the third block type of the programmed GAML model

Implementation result of *species* representing an agent The fourth block consists in defining species representing agents. Let us take an example of an actor agent through the *semMAS:agent_fireman* and a manager agent through the *semMAS:agent_fireman_officer* to illustrate the implementation result of this block type.

- The header of actor agents is defined through their individual name and their skills. In the case of the *semMAS:agent_fireman*, it has three skills *semMAS:moving*, *semMAS:actingOnPeople*, and *semMAS:fipa*. All agents have the skills fipa[9] to allow their interaction through communication. The communication allows central and manager to assign task to their subordonate. All subordonate agents have a reflex to reply by accepting the request of their superior and trigering the message content as its objective. The manager agents add the message content as a belief to trigger the task, whereas actor agents add the message content as its objective. Then, attributes and reflex of actor agents are described according to the SemMAS ontology content. The representation of the transport reflex defines that the *semMAS:agent_fireman* has a casualty (*acm*) and the reflex *transport_756daed*. This reflex is applied when the agent has the objective corresponding to this reflex and a casualty with the status *status_15d496q8g*. The reflex header is thus build from the value of the properties *semMAS:what*, *semMAS:requireStatus*, and *semMAS:requiresObjective*. The value of the property *semMAS:results* defines the casualty status when the goal of the reflex is reached. This implementation is illustrated in the purple box of Figure 7.32 and corresponds to the SemMAS ontology part presented in the purple box of Figure 7.31.

[9]Link to FIPA ACL in GAMA

- The header of central and manager agents are defined through their individual name. They are characterized by the skill *fipa* and by the structure of a simple BDI agent (*control:simple_bdi*). Then, each of their attributes, rules and plans are implemented. The *semMAS:agent_fireman_officer* has a rule (*semMAS:rule_9d47g7x20*), which is defined through the belief *semMAS:belief_cn_management*, the new desire *semMAS:desire_collection_noria_management*, and the strength 1. It has the plan *bdi_plan_cn_management* that realizes the desire *semMAS:desire_collection_noria_management*, when it becomes the agent's intention. This plan assigns the objective "*search agent_casualty_montbard1*" to all *agent_fireman* through a FIPA request[10]. The plan finishes when there is no more casualty in danger (*nb_casualty_montbard1_inDanger=0*), which express its final condition. It finishes by removing the belief that triggers the desire associated to this plan. This implementation is illustrated in the orange box of Figure 7.32 and corresponds to the SemMAS ontology part presented in the orange box of Figure 7.31.

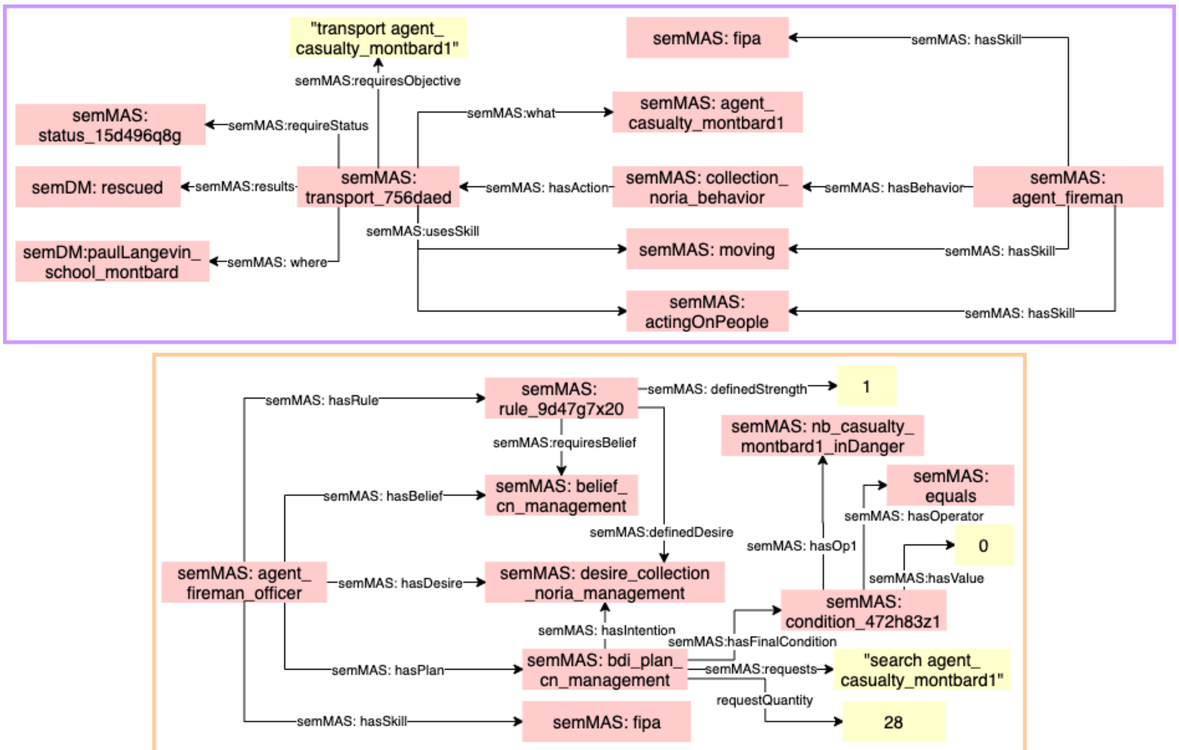


Figure 7.31: Parts of SemMAS ontology for the implementation of agent examples

[10]Link to FIPA Request Interaction Protocol

```
species agent_fireman skills: [ moving, actingOnPeople, fipa] {
agent_casualty_montbard1 acm <- nil;
...
reflex transport_756daed when: objective ="transport_agent_casualty_montbard1" and acm != nil and acm.status = "status_15d496q8g" {
        do goto target: the_target on: the_graph ;

        if the_target = location {
        the_target <- nil ;
        objective <- nil ;
        acm.location <- the_target;
        acm.status <- "rescued";

        }
   }
   ...
}                                                                                    4
```

```
species agent_fireman_officer control: simple_bdi skills: [fipa] {
     ...
     rule belief:belief_cn_management new_desire: desire_collection_noria_management strength: 1;

     plan bdi_plan_cn_management intention: desire_collection_noria_management {
        do start_conversation (to: 28 among list(agent_fireman), protocol: 'fipa-request', performative: 'request', contents: ['search agent_
        casualty_montbard1']);
        if nb_casualty_montbard1_inDanger = 0 {
            do remove_belief(belief_cn_management);
     }
}                                                                                    4
```

Figure 7.32: Examples of implementation result of the fourth block type of the programmed GAML model

Implementation result of *experiment* inside the GAML program The last part of the GAML model corresponds to the experiment definition. The process retrieves individuals of *semMAS:ExperimentalModel* to implement each of them. For each individual, it builds the experiment header with its name, its closest type, the number of repetition and the final condition. The body of experiments is built from the parameter of the model (*semMAS:hasParameter*). In this case study, the experimental model illustrated in Figure 7.33 has three parameters:

1. the casualty quantity represented through a list (*semDM:casualty_quantity*),
2. the proportion of physician assigned to the advanced medical post represented through the interval *semMAS:interval_15g923t47* (that defines the resource *semMAS:amp_management_physician*), and
3. the proportion of physician assigned to the hospital represented through the interval *semMAS:interval_60v385r19* (that defines the resource *semMAS:hospital_management_physician*).

The implementation of these parameters is presented in Figure 7.34. The parameter represented by an interval is implemented with the name of the resource, whose it defines the proportion, its minimal, maximal, and step values, whereas the parameter represented through a list is implemented with its name and a table containing the values of the list. The case of configurations that defines the usage of a resource more than it is available is managed by the agent in charge of these assignments. For example, if a configuration defines 80% of physicians to the advanced medical post and 30% to the hospital, the agent will assign one of this configuration in first and will assign the second one in the limit of the 100% of assignment (e.g., 80% in the advanced medical post and 20% in the hospital or 30% in the hospital and 70% in the advanced medical post).

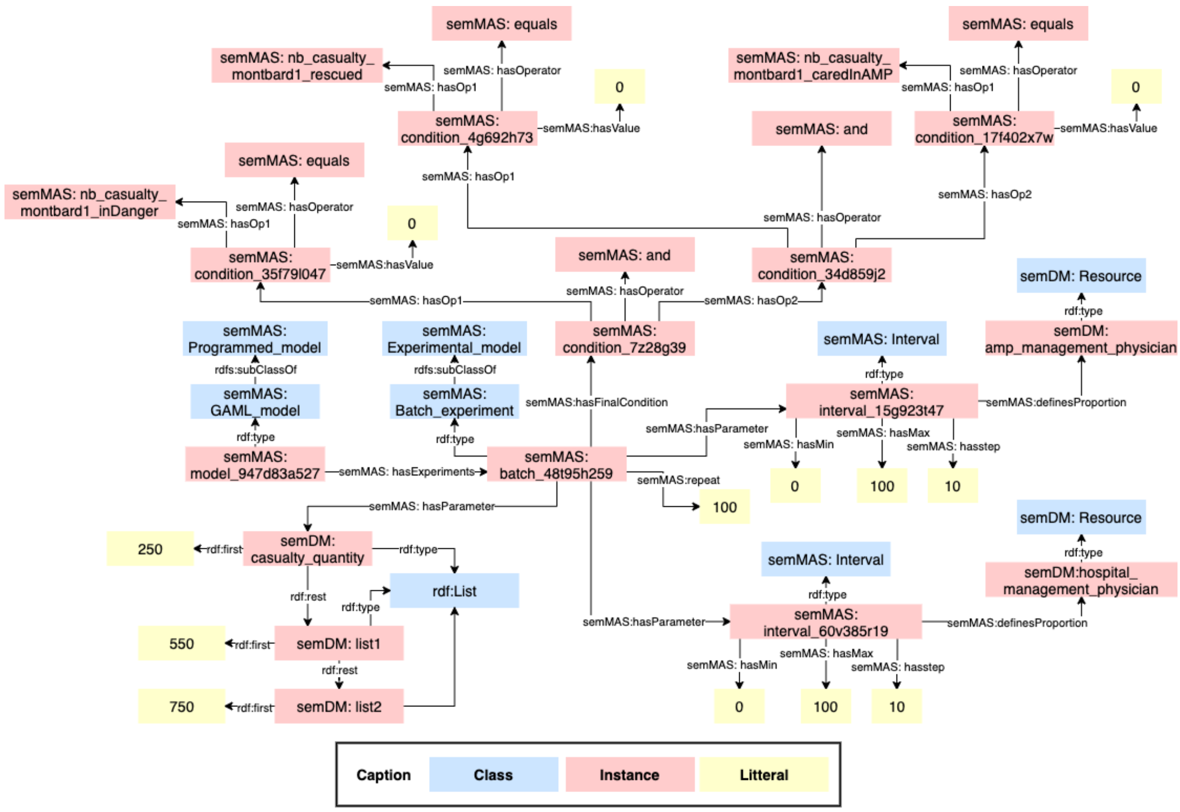


Figure 7.33: Experimental model of the studied case represented in SemMAS ontology

```
experiment batch_48t95h259 type: batch repeat: 100 until: ( nb_casualty_montbard1_inDanger   =  0  and
                                                            nb_casualty_montbard1_rescued    =  0  and
                                                            nb_casualty_montbard1_caredInAMP =  0  ) {

  parameter " amp_management_physician " var: amp_management_physician min: 0  max: 100  step: 10 ;

  parameter " hospital_management_physician " var: hospital_management_physician  min: 0  max: 100  step: 10;

  parameter " casualty_quantity " var: casualty_quantity among: [ 250, 550, 750 ] ;
}                                                                                                   5
```

Figure 7.34: Code generated from the experimental model

The results of the generative programming applied to the case study is the model *model_947d83a527* that contains two artifacts, which are *building* and *road*. This model is composed of three types of actor agents, which are the *agent_fireman*, the *agent_ambulance*, and the *agent_physician*. It is composed of three types of manager agents, which are the *agent_fireman_officer*, the *agent_DSM*, and the *agent_COS*. Finally, the model contains a central agent, which is *agent_DOS* and the experiment model *batch_48t95h259*.

The next section presents the simulation experiments execution.

7.4.2 Simulation execution

Experiment execution After the generative programming, the GAML model of simulation is executed through the GAMA platform. As explained in section 6.3.2 of the previous chapter, the execution of the experiments is done through a bash command. Code 7.1 presents the command used to execute simulation experiments of the case study. It is composed of the bash script, an input XML file and a repository to store results of experiments.

```
bash gama-headless.sh model/experiment/model_947d83a527_exp.xml results/
    outputHeadLess
```

Code 7.1: Bash command to execute experiments of the case study in headless mode

Experiment input XML file This bash command takes an input XML file describing the experiment configurations to execute. Let us illustrate the three parts of the first experiment configuration description of this case study.

1. **Heading.** The heading of an experiment configuration has an id, which is incremented one by one, the path file of the model corresponding to *"./model_947d83a527.gaml"*, the final condition of experiment *"nb_casualty_montbard1_inDanger = 0 and nb_casualty_montbard1_rescued = 0 and nb_casualty_montbard1_caredInAMP = 0"*, and the name of the experiment *"batch_48t95h259"*.

2. **Parameter.** As shown in the experimental model, this case study has two parameters, which are *amp_management_physician* and *casualty_quantity*. For the first configuration, these parameters have the values 0 and 250, respectively.

3. **Output.** Outputs of the experiment are the same for all experiment configurations. They correspond to the observed variable of the model.

Code 7.2 presents the XML input file part corresponding to the description of the first configuration of the case study experiment plan.

```
<?xml version="1.0" encoding="UTF-8"?>
<Experiment_plan>
 <Simulation id="1" sourcePath="./model_947d83a527.gaml" until="nb_casualty_montbard1_inDanger = 0 and nb_casualty_montbard1_rescued = 0 and nb_casualty_montbard1_caredInAMP = 0"
 experiment="batch_48t95h259">
  <Parameters>
    <Parameter name="amp_management_physician" type="INT" value="0" />
    <Parameter name="casualty_quantity" type="INT" value="250" />
  </Parameters>
  <Outputs>
    <Output id="1" name="amp_management_physician" />
    <Output id="2" name="hospital_management_physician"/>
    <Output id="3" name="casualty_quantity"/>
    <Output id="4" name="nb_casualty_montbard1_EU_init"/>
    ...
  </Outputs>
 </Simulation>
 <Simulation id="2" ...>
 ...
 </Simulation>
 ...
</Experiment_plan>
```

Code 7.2: XML input file part corresponding to the description of the first experiment configuration

Simulation results for the case study The simulation results obtained from the experiment plan and the programmed model are represented in CSV files stored into the CSV directory. Figure 7.35 presents a screenshot of one of these experiments. This visualization is obtained through a GUI experiment and shows the simulation components: the environment based on geospatial data of Montbard, the actor agents on the ground, and victims by health state.

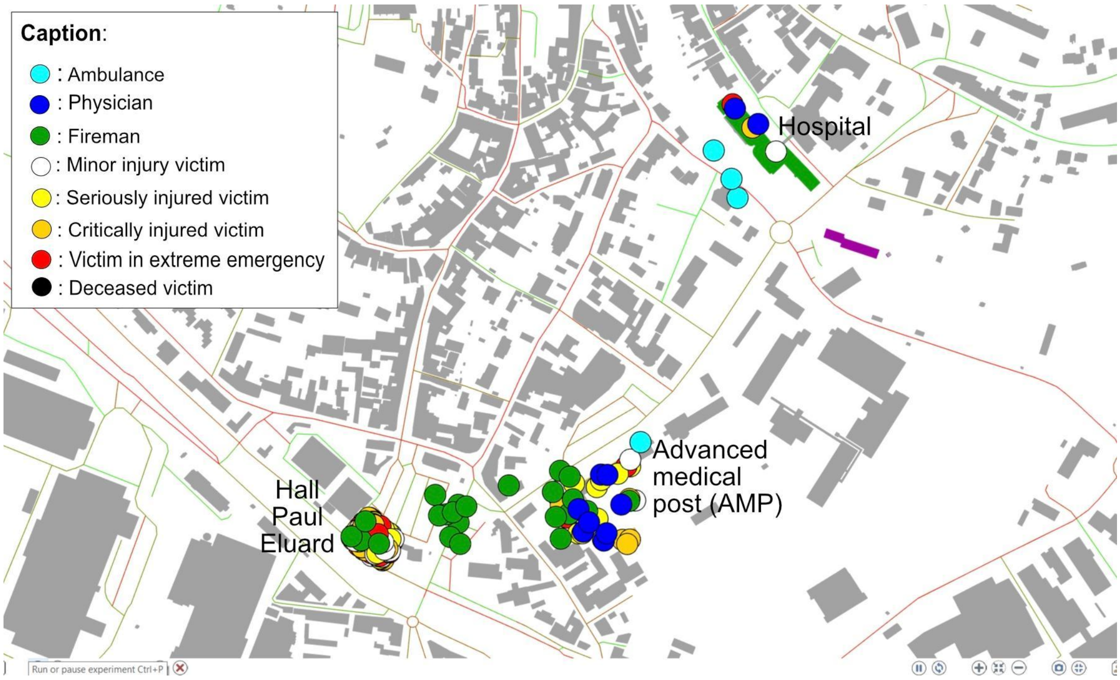


Figure 7.35: Screenshot of the simulation

The results obtained by the simulation experiments are then analyzed to assess plans and enrich the knowledge base. The next section presents the clustering-based analysis applied to these results to enrich the knowledge base.

7.5 Clustering-based analysis

The fourth and last step of the methodology for plan assessment is the clustering-based analysis. Figure 7.36 shows this step of processing in the overall system processing.

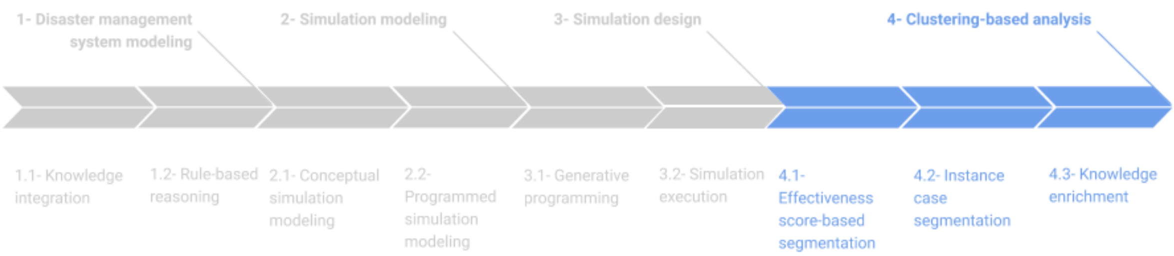


Figure 7.36: Fourth step of the application: Clustering-based analysis

The previous step provides experimentation's results stored in a CSV file. Each experimentation details many variables such as the percentage of physicians assigned to the hospital, the percentage of physicians assigned to the advanced medical post, the total number of casualties, the number of casualty at a specific health state (urgent, heavy, light, healthy), and the number of casualty dead during the plan application. These variables are saved at the beginning and the end of the simulation.

Even if the process considers all of these variables, only three variables are illustrated in the following examples to provide a better view of the plan assessment process. These variables are the percentage of physicians assigned to the advanced medical post (amp_management_physician), the number of casualties (casualty_quantity), and the number of casualties dead (nb_casualty_montbard1_dead) during the plan application.

The experimentation set is composed of 7597 experiments. These experiments are based on the multiple iterations for each variation of casualty number (250, 500, and 750) and each percentage variation of the physicians assigned to the AMP (from 0% to 100% with a step of 10%). The mean experiment is computed for a set of simulation experiment variations to provide a better view of the plan assessment mechanism.

7.5.1 Effectiveness score-based segmentation

The first step consists of computing the effectiveness of each experiment to group them according to their effectiveness score.

Effectiveness computation The objective of the NOVI plan is mainly oriented for limiting the number of dead. Therefore the effectiveness of each experiment is computed through the equation:

$$effectiveness = \frac{casualty\_quantity - nb\_casualty\_montbard1\_dead}{casualty\_quantity}$$

Experiment clustering Experiments are clustered through the use of a hierarchical clustering implementation. The optimal number of categories found is 6. Thus experiments are grouped into six clusters.

Table 7.3 shows the effectiveness value and cluster's number for each mean experiment, computed for each variation of the percentage of physicians assigned to the AMP and the number of casualties.

C	effectiveness	amp_ management_ physician (%)	casualty_ quantity	nb_casualty_ montbard1_ dead
1	0	0	250	250
1	0.00013333333333336364	0	750	750
1	0.00020000000000004546	0	500	500
3	0.4428	10	750	418
3	0.538	10	500	231
4	0.6333333333333333	20	750	275
4	0.6946666666666667	30	750	229
4	0.6964	10	250	76
4	0.6984	90	500	151
4	0.7102	20	500	145
6	0.7268	90	750	205
6	0.7361	70	750	198
6	0.7373	40	750	197
6	0.7454	80	750	191
6	0.7466	50	750	190
6	0.764	60	750	177
6	0.768	40	500	116
5	0.788	30	500	106
5	0.79	50	500	105
5	0.8142	60	500	93
5	0.8162	80	500	92
5	0.8182	70	500	91
5	0.828	20	250	43
2	0.872	40	250	32
2	0.88	90	250	30
2	0.88	30	250	30
2	0.896	70	250	26
2	0.9	60	250	25
2	0.9	50	250	25
2	0.9004	80	250	25

Table 7.3: Result of the clustering based on effectiveness score

Since experiments are clustered in more than one cluster, the plan is classified as specific, and the knowledge is enriched by the following SPARQL insert query where *semDM: plan_novi* corresponds to the individual of the assessed plan.

```
INSERT DATA {semDM:plan_novi semDM:hasApplicability semDM:
    specific_applicability}
```

Code 7.3: SPARQL INSERT query to classify the NOVI plan as specific

7.5.2 Instance case segmentation

As explained in section 6.4.2 of the chapter 6, if the effectiveness-based clustering results in more than one cluster, the analysis continue with segmentation for each variable. The different variables of the case study (representing the columns of the spreadsheet containing the experiments) are segmented into groups by the same strategy than the effectiveness-based clustering. Thus each experiment is linked to different clusters according to the different variables studied (c.f. figure 6.16 of section 6.4.2).

Let us take as an example, the effectiveness-based clusters number 3, number 2, and number 4 that illustrate the different possibilities of cluster configurations based on the other variables (corresponding to the model's parameters). Table 7.4 shows the sample of these three effectiveness-based cluster.

Variable clustering The effectiveness-based cluster number 3 represents a simple type of cluster encountered for the plan assessment. The clustering applied for each variable gathers all experiments contained in the cluster number 3 in one cluster for the "amp_management_physician" variable and cluster each experiment in an individual cluster for the variable "casualty_quantity". Table 7.5 illustrates the clustering for each variable. The letters (a to m) correspond to different clusters.

Cluster filtering Among all variable-based clusters of the effectiveness-based cluster number 3, only the cluster "a" based on the variable "amp_management_physician" (cluster an in table 7.5) is common to all experiments of the effectiveness-based cluster 3. Thus the "amp_management_physician" variable is considered mainly impacting plan's effectiveness for such range. Its value characterizes the effectiveness score represented by the effectiveness-based cluster 3. The other variable, "casualty_quantity", is used to define the situation in which the variable "amp_management_physician" impacts this effectiveness score.

C	effectiveness	amp_ management_ physician (%)	casualty_ quantity	nb_casualty_ montbard1_ dead
3	0.4428	10	750	418
3	0.538	10	500	231
4	0.6333	20	750	275
4	0.69466	30	750	229
4	0.6964	10	250	76
4	0.6984	90	500	151
4	0.7102	20	500	145
2	0.872	40	250	32
2	0.88	90	250	30
2	0.88	30	250	30
2	0.896	70	250	26
2	0.9	60	250	25
2	0.9	50	250	25
2	0.9004	80	250	25

Table 7.4: Clusters 2, 3, and 4 resulting from the effectiveness-based clustering

Its minimal and maximal values are computed to define the situation associated with it.

Similarly, for the clustering of the effectiveness-based cluster number 2, only the cluster "m" based on the variable "casualty_quantity" is common to all experiments. In this case, the value of the variable "casualty_quantity" is identified as impacting and characterizing the effectiveness of the cluster 2. The other variable, "amp_management_physician," is used to define the situation by computing minimal and maximal values.

Contrary to the two previous effectiveness-based clusters, the cluster 4 does not have a variable-based cluster common to all experiments. However, the analysis of this effectiveness-based cluster 4 shows that this cluster can be sub-divided into three sub-clusters having the same "casualty_quantity" variable-based cluster. On the contrary, a subdivision according to the "amp_management_physician" variable-based clusters would produce an over-segmentation by creating a cluster for each experiment (i.e., bk, ck, am, dl, el). Therefore, the analysis subdivides the effectiveness-based cluster 4 into the three sub-clusters illustrated in table 7.6 belonging to the same "casualty_quantity" variable-based cluster. Each of these new sub-clusters (i.e., cluster 4.1, cluster 4.2, cluster 4.3) provides similar conclusions than the effectiveness-based cluster 2. The effectiveness value of these three sub-

C	effectiveness	amp_ management_ physician (%)	casualty_ quantity	nb_casualty_ montbard1_ dead
3	0.4428	a	k	418
3	0.538	a	l	231
4	0.6333	b	k	275
4	0.69466	c	k	229
4	0.6964	a	m	76
4	0.6984	d	l	151
4	0.7102	e	l	145
2	0.872	f	m	32
2	0.88	d	m	30
2	0.88	c	m	30
2	0.896	g	m	26
2	0.9	h	m	25
2	0.9	i	m	25
2	0.9004	j	m	25

Table 7.5: Results of the clustering applied on each variable for the effectiveness-based clusters

clusters are impacted by the value of their associated "casualty_quantity" variable-based cluster and the "amp_management_physician" variable is used to define the situation by computing its minimal and maximal values.

4.1	0.6333	b	k	275
4.1	0.69466	c	k	229
4.2	0.6964	a	m	76
4.3	0.6984	d	l	151
4.3	0.7102	e	l	145

Table 7.6: Result of clustering applied to the effectiveness-based cluster number 4 according to the *casualty_quantity* variable-based clusters

7.5.3 Knowledge enrichment

Let us take as an example, knowledge enrichment through the analysis of the effectiveness-based cluster 3. This cluster constitutes the situation for which the plan has effectiveness around 50%. Among different experiment variables, only the variable "amp_management_physician" is considered as impacting the plan effectiveness with a constant value of 10 in cluster number 3. Its value is extracted to constitute the situation for which the assessed plan has effectiveness around 50%. Information about the cluster is extracted to create a JSON object automatically, as presented in Code 7.4 for the cluster 3.

```
{
   "plan":"plan_novi",
   "e":{
      "values":[0.4428,0.538],
      "minValue":0.4428,
      "maxValue":0.538
      },
   "v":[{
      "name":"amp_management_physician",
      "hasValue":0.1,
      "minValue":null,
      "maxValue":null
      },
      {
         "name":"casualty_quantity",
         "hasValue":null,
         "minValue":500,
         "maxValue":750
      }]
}
```

Code 7.4: JSON Object example corresponding to extracting information of the cluster 3

Thus the SPARQL CONSTRUCT query, presented in Code 7.5, is generated.

```
CONSTRUCT{
   semDM:plan_novi semDM:hasEffectiveness ?e.
   ?e semDM:relatedTo ?s.
   ?e semDM:hasMin 0.4428.
   ?e semDM:hasMax 0.538.
   ?e semDM:hasAverage ?average.
   ?e semDM:hasMedian ?median.
   ?e semDM:hasMeanSquare ?meanS.
   ?s rdf:type semDM:Situation.
   ?s semDM:characterizedBy ?v0.
   ?v0 rdf:type semDM:Variable.
   ?v0 semDM:linkedWith ?init0.
   ?v0 semDM:hasValue 0.1.
   ?s semDM:characterizedBy ?v1.
   ?v1 rdf:type semDM:Variable .
   ?v1 semDM:linkedWith ?init1.
```

```
    ?v1 semDM:hasMax 750.
    ?v1 semDM:hasMin 500.
  } WHERE {
    ?e semTransform:generateURI(semDM, "effectiveness").
    ?s semTransform:generateURI(semDM, "situation").
    ?v0 semTransform:generateURI(semDM, "amp_management_physician").
    ?init0 semTransform:searchIndividual(semDM, "
       amp_management_physician").
    ?v1 semTransform:generateURI(semDM, "casualty_quantity").
    ?init1 semTransform:searchIndividual(semDM, "casualty_quantity").
    ?average semStatistics:GetAverage ( "0.4428", "0.538" ) .
    ?median semStatistics:GetMedian ("0.4428" , "0.538" ) .
    ?meanS semStatistics:GetMeanSquare ("0.4428" , "0.538" ) .
  }
```

Code 7.5: SPARQL CONSTRUCT query to enrich knowledge base with plan assessment

The enrichment is applied for each cluster to provide a plan assessment.

7.5.4 Plan assessment result

In this case study, the NOVI plan assessment is summarized as follows. Figure 7.37 shows the effectiveness of NOVI plan in the SemDM ontology resulting from the effectiveness-based cluster 1 surrounded in orange, 2 surrounded in blue, and 3 surrounded in purple. According to the effectiveness-based cluster 1, if no physicians are assigned to the AMP, the plan is inefficient (0% of effectiveness). According to the effectiveness-based cluster 2, the NOVI plan applied to the Montbard use case has high effectiveness (around 90%) when the number of casualties is 250, and the proportion of physicians assigned to the advanced medical post is comprised between 30% and 90% (included). According to the effectiveness-based cluster 3, the NOVI plan applied to the Montbard use case has average effectiveness (around 50%) when the number of casualties is between 500 and 750, and only 10% of physicians are assigned to the AMP.

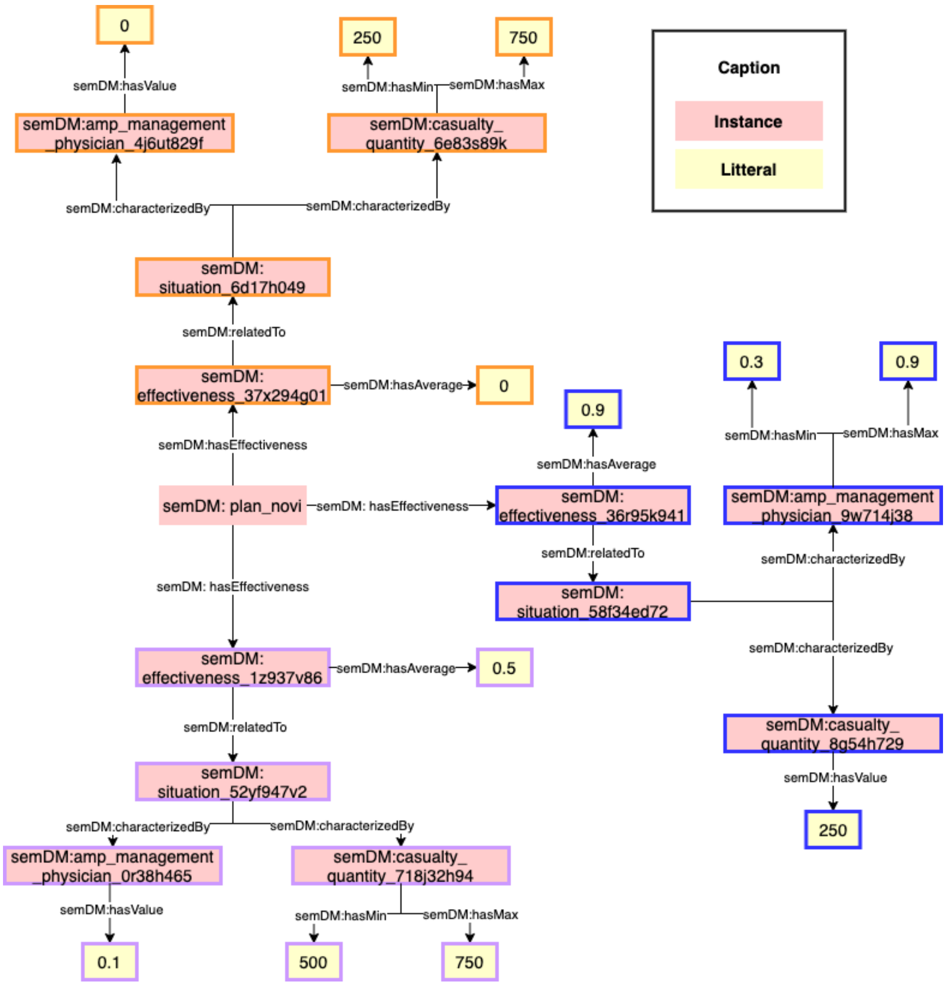


Figure 7.37: Effectiveness resulting from effectiveness-based cluster 1 surrounded in orange, 2 surrounded in blue, and 3 surrounded in purple.

According to the effectiveness-based cluster 4, the NOVI plan applied to the Montbard case study has effectiveness around 70%

- If the casualties number is 750 and physicians assigned to the AMP are between 20% and 30%; or
- If the number of casualties is 250 and only 10% of physicians are assigned to the AMP; or
- If the number of casualties is 500 and 90% or 20% of physicians are assigned to the AMP.

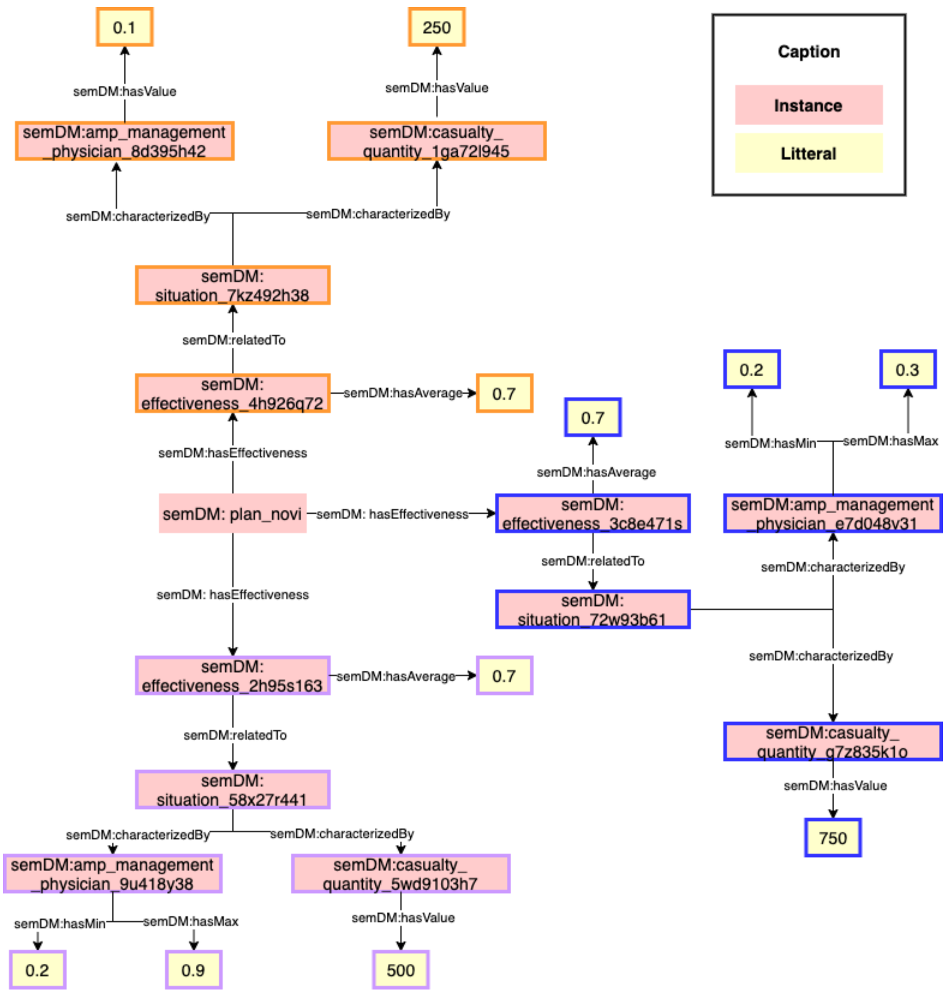


Figure 7.38: Effectiveness resulting from effectiveness-based cluster 4 (surrounded with different colors for the different sub-groups of effectiveness).

According to the effectiveness-based cluster 5, the NOVI plan applied to the Montbard use case has effectiveness around 80%

- If the number of casualties is 500 and between 30% and 80% of physicians are assigned to the AMP; or

- If the number of casualties is 250 and an average of 20% of physicians are assigned to the AMP.

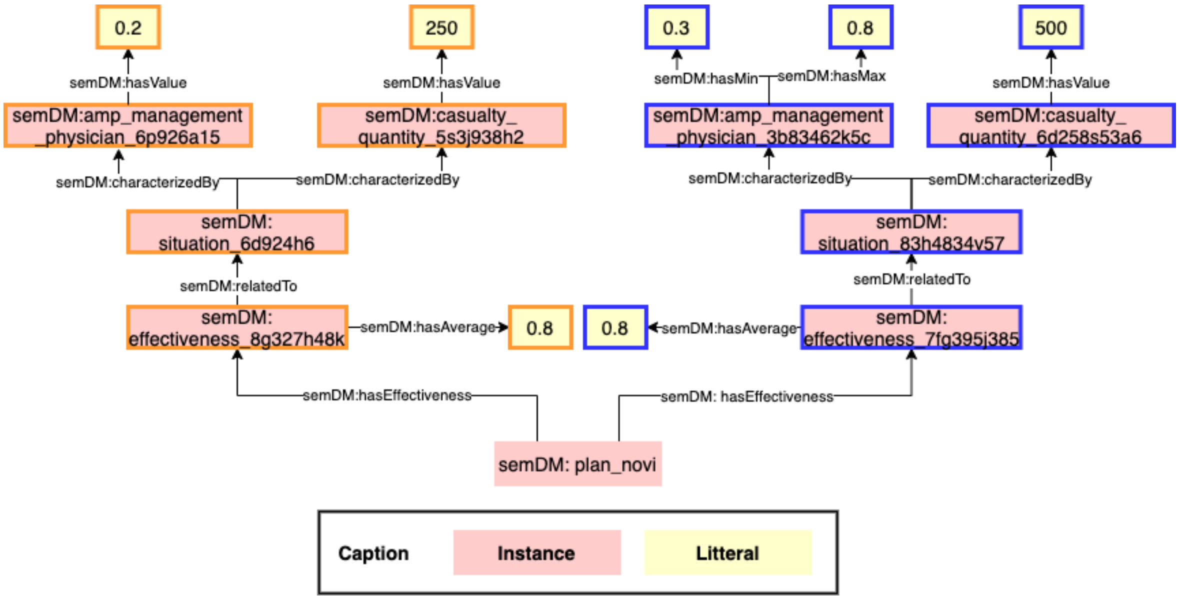


Figure 7.39: Effectiveness resulting from effectiveness-based cluster 5 (surrounded with different colors for the different sub-groups of effectiveness).

According to the effectiveness-based cluster 6, the NOVI plan applied to the Montbard use case has effectiveness around 75%

- If the casualties number is 750 and physicians assigned to the AMP are between 40% and 90% ; or
- If the number of casualties is 500, and 40% of physicians are assigned to the AMP.

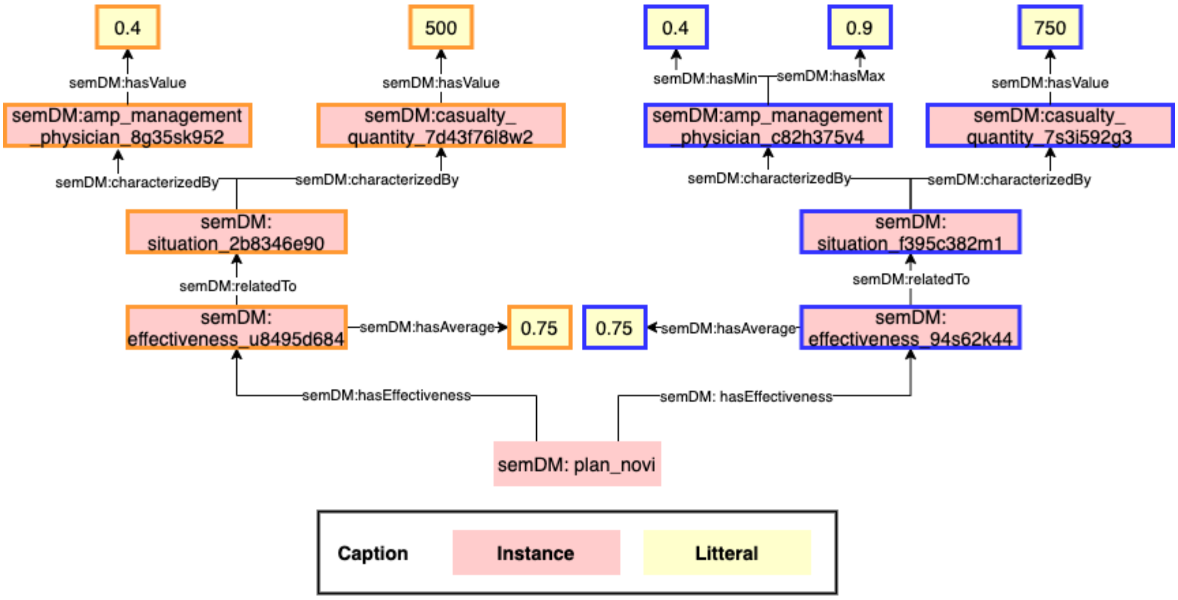


Figure 7.40: Effectiveness resulting from effectiveness-based cluster 6 (surrounded with different colors for the different sub-groups of effectiveness).

Knowledge resulting from the effectiveness assessment This assessment shows that the NOVI plan applied to such a situation is ineffective if no physicians are assigned to the advanced medical post. It obtains the highest effectiveness (90%) in a situation of 250 casualties, where the percentage of physicians assigned to the advanced medical post is comprised between 30% and 90%. This case corresponds to knowledge generated from cluster 2. For 750 casualties, the highest effectiveness score (75%) is obtained when the percentage of physicians assigned to the advanced medical post is comprised between 40% and 90%. This knowledge comes from the cluster 6. In the case of 500 casualties, the highest effectiveness score (80%) is obtained when the percentage of physicians assigned to the advanced medical post is comprised between 30% and 80%. This knowledge comes from cluster 5. A percentage of 40% belongs to this interval and, thus, to this situation. Therefore, we could expect the effectiveness of 80% in the case of 500 casualties and 40% of physicians assigned to the advanced medical post. However, according to cluster 6, this situation characterizes the effectiveness of 75%. Therefore, this knowledge highlights the representation's inaccuracy through an interval delimited through minimal and maximal values to represent the set of experiments describing the situation. This inaccuracy can be corrected by modeling each value representing the experiment types belonging to the cluster through a list rather than the representation of an interval based on only two values, minimal and maximal.

Discussion This case study shows the benefit of the plan assessment provided by the proposed approach. It identifies situations that produce an ineffective plan application and identifies situations to obtain the highest effectiveness of the plan. Besides, the highest plan's effectiveness informs about the effectiveness to expect in the best configuration. This knowledge acquisition is essential in the process of disaster management preparedness to improve plans and make an effective decision when a disaster happens. The first limit observed in this case study is the limit linked to the scenario definition. The knowledge of plan effectiveness is linked to the application scenario defined to assess a plan. In this case study, the knowledge of effectiveness is limited to the three configurations of victims (i.e., 250, 500, and 750 victims) defined as scenarios. For appropriate preparation, it would be required to define a broad diversity of scenarios. The more numerous and diversified the scenarios are, the more knowledge is obtained by the proposed approach. The second limit observed for this case study appears at the level of variable description characterizing a situation. This description is based on an interval through minimal and maximal value representing the cluster values. This representation is an approximation producing inaccuracy that must be corrected by describing the set of values through a *rdf:list*.

7.6 Summary

This chapter has illustrated the application of the proposed approach to a case study. This case study concerns the NOVI plan applied in Montbard for a disaster event affecting the Paul Eluard hall with three victims' configurations. The modeling of this case study provided by a user and his data interpretation have been illustrated in section 7.2. This modeling is composed of the description of sub-procedures, services, tasks, and roles intervening in applying the NOVI plan. The provided data allows the extraction of knowledge related to the roads and buildings of Montbard.

The case study modeling in the SemDM ontology is then used for the simulation conceptualization in the SemMAS ontology, presented in section 7.3. This process uses geospatial information to define the model's environment and artifacts. Then it uses roles, their associated services, procedures, and tasks or actions to define agents and their behaviors. It continues by specifying the simulation modeling for the GAMA platform by defining a GAML model and other specificities of the platform as an agent's skills or SimpleBDI agents. It results in a conceptual simulation model and a representation of the programmed model in GAML.

The programmed model represented in the SemMAS ontology is used as a base for the generative programming process that creates the simulation code. This process follows the structure in five blocks of a GAML model presented in the previous chapter. Each of these blocks has a predefined structure fulfilled according to the SemMAS ontology's content. The ontology's content, more precisely, the experimental model representation, is then used to define the experimental plan. Simulation experimentations results are then stored through CSV files in an output directory.

The results represented into CSV files are used as input of the clustering-based analysis. This process, firstly, computes the experiment's effectiveness. It secondly executes an effectiveness-based clustering on the different experiments to identify if the plan has global or specific applicability. If this clustering produces more than one cluster (i.e., it has specific applicability), it applies a clustering on each simulation parameter. The results of these clustering are used to enrich the plan representation with its assessment.

The knowledge enrichment result describes the NOVI plan as having specific applicability with ten effectiveness definition specific to a situation depending on the percentage of physicians assigned to the advanced medical post and the casualty quantity. This plan assessment provides feedback on the plan's effectiveness in the disaster management community. It results in the knowledge of configurations for which the plan obtains the weakest effectiveness and the knowledge of the highest effectiveness to expect according to the situation. According to the effectiveness' value and the associated situation description, the knowledge of situations in which the plan is ineffective or lowly effective allows its improvement or the creation of a new one effective in these situations. Besides, the proposed approach applied to this case study provides the knowledge of the percentage of physicians assigned to the advanced medical post that results in the best effectiveness according to the casualty quantity. However, this knowledge is currently inaccurate and must be improved by replacing the interval defined by minimal and maximal values representing a cluster's variable through a list that would provide precise knowledge of the situation characterizing effectiveness.

Aside from the effectiveness representation, the efficiency of the proposed approach depends mainly on the generated multi-agent simulation model. Therefore, the next chapter presents its evaluation.

8 Evaluation

Contents

8.1 Evaluation methods 256

8.1.1 Criteria metrics for high-level metrics computation 256

8.1.2 Computation of metrics 260

8.2 Evaluation application 261

8.2.1 Evaluation of environment complexity 262

8.2.2 Evaluation of agent's rationality 264

8.2.3 Evaluation of agent's autonomy 266

8.2.4 Evaluation of agent's reactivity 268

8.2.5 Evaluation of agent's adaptability to the Environment . . . 269

8.3 Discussion 272

8.3.1 Rationality vs. Environment Complexity 272

8.3.2 Autonomy vs. Environment Complexity 273

8.3.3 Reactivity vs. Environment Complexity 274

8.3.4 Adaptability vs. Environment Complexity 274

8.3.5 Conclusion 275

The approaches of related work do not propose an assessment of disaster management plans. It is so, not possible to assess the efficiency of the proposed approach by comparing it with another similar system. The simulation model is the linchpin of the proposed approach. It is both the results of the information integration and the simulation conceptualization, but also the source of plan assessment. Thus, the validation of the multi-agent simulation can validate at least the essential part of the system. Indeed, if the multi-agent system would be approved, that means the information integration and the simulation conceptualization would have well worked. Similarly, if the multi-agent system would be validated, the simulation would be reliable. Therefore, the evaluation and validation of the multi-agent simulation play a significant role to evaluate the proposed approach. The traditional way to assess and validate a simulation model is to compare results obtained by the simulation with real data collected in the same situations. Due to the lack of such existing data and the legal difficulty of sharing, we have no access to such data. For a lack of comparison with results of real data to compare with those obtained by the multi-agent simulation, this thesis has evaluated the multi-agent simulation resulting from the process of simulation conceptualization through a metric plan. The first section explains the metric plan used to assess the multi-agent simulation. The second section presents the application of the metric plan on the application case. Finally, the last section discusses the conclusions of the evaluation.

8.1 Evaluation methods

The methods used to assess the multi-agent simulation model is the metric plan presented in [Di Bitonto et al., 2012]. This metric plan is based on a Goal-Question-Metrics. It provides high-level metrics to measure the agent characteristics such as rationality, autonomy, reactivity, and adaptability to the environment, but also metrics to assess the environment complexity. Thanks to these metrics, this method provides two evaluation perspectives of a multi-agent system: inter-agent and intra-agent. The intra-agent aspect corresponds to the evaluation of the internal structure of agents in terms of capabilities and abilities. In contrast, the inter-agent perspective considers the global and overall multi-agent system corresponding to aspects of communication and cooperation between agents. These two perspectives are the benefit of this approach compared to other methods for MAS evaluation. Indeed, it allows a complete assessment both at the level of the agent and the MAS level. The different approaches focus only on some aspects as the agent performances or the communication between agents.

8.1.1 Criteria metrics for high-level metrics computation

The metric plan proposes the measurement of five high-level metrics: the environment complexity, the rationality, the autonomy, the reactivity, and the adaptability to the environment. High-level metrics are computed from metrics corresponding to values assigned to different criteria (C_i). These criteria characterize the high-level metric. Their value is assigned using a goal-question-methods. This methods allows the assignment of the value *0* if the criteria is low, *0,5* if the criteria is medium, *1* if the criteria is high. When the criterion corresponds to ability, the score assigned is *0* or *1* according to if the ability is present and if it is an advantage to have it or not. Criteria characterizing each metric are presented below.

The environment complexity
The metric representing the environment complexity depends on eight criteria gathers in three categories:

1. Inaccessibility. The inaccessibility represents the difficulty of accessing completely to the environment resources at any instant. The inaccessibility is assessed according to two criteria:
   - *CompInacc*, the inaccessibility of the agent environment components;
   - *ResInacc*, the inaccessibility of the agent environment resources.
2. Instability. The instability represents the evolution way of the environment and how fast. It is expressed through three criteria:
   - *Time*, which can be continuous (V=1) or discrete (V=0);
   - *Dynamic* of the environment, which can change while the agent is thinking or acting (i.e.dynamic with V=1) or not (i.e. static with V=0);
   - *NumEffeAct* represents how unpredictable changes of the environment are. It is assessed according to the effects of the agent's action. If it can have several effects, then it has the value 1, otherwise 0.
3. Complexity of the interaction. The complexity of the interaction represents how complex the interactions between agents are in the MAS. It is expressed through three criteria:
   - *CompGrad* represents the competition degree, which is estimated by the presence of competition between an agent with another (V=1), or not (V=0).
   - *CoopGrad* represents the cooperation degree, which is estimated by the presence of cooperation between an agent with another (V=1), or not (V=0).
   - *Tr&RepMod* represents the need to use trust and reputation models. 1 is assigned if such a model is required, otherwise 0.

The rationnality
The metric representing the rationality depends on five criteria gathers in two categories:

1. Mode of action choice. The mode of action choice represents the degree of rationality in choosing the actions to be performed. It is evaluated according to four criteria:

- *AgType* represents the rationality assigned to different types of agents. Goal-based agents have high rationality (V=1) contrary to simple or stimulus-response agents (V=0);
- *PlaConstr* represents the agent's ability to build plans of action. The presence of this ability implies greater rationality (V=1) than its absence (V=0);
- *LearAb* represents the agent's ability to learn. Similarly, its presence implies greater rationality (V=1) than its absence (V=0);
- *InsMod* represents the agent's possession of an internal model of the actions and intentions of the other MAS agents that make it more rational (V=1) than no property of such an internal model (V=0).

2. Success maximization: Maximization success represents the ability to maximize the action expected result. It is estimated as a normalized difference between the expected results and the obtained results. If the purchased and expected results are equal, it indicates a maximum success, which is not rational. Therefore, in this case, the metric value is 0; otherwise, it is 1.

The autonomy
The metric representing the autonomy depends on six criteria gathers in two categories:

1. Proactivity. The proactivity represents the ability to take the initiative rather than react to the environment. It is assessed according to four criteria:
   - *MoreRol* estimates the ability of an agent to pass to one role to another autonomously. If an agent has more than one role and can change independently from one to another, the assigned value is one, otherwise 0.
   - *NegAg* estimates the agent's ability of negotiation. If an agent has this ability (V=1), he is more autonomous than without this ability (V=0).
   - *DiaErPrAb* estimates the ability to diagnose errors and/or problems during execution of the tasks. If an agent has this diagnostic ability (V=1), he is more proactive than without it (V=0).
   - *ComAutAb* estimates the ability to communicate with other agents. The presence of this ability assigns the value 1, otherwise 0.
2. Autonomy in the Organizational Structure. This category represents the degree of action autonomy of an agent within the MAS organization. It is evaluated according to two criteria:

- *PosStr* represents the autonomy degree of an agent according to the subordinate position in MAS. A subordinate has less autonomy (V=0) than a no-subordinate agent (V=1).
- *SharTask* represents the autonomy degree of an agent according to its ability of task sharing. An agent that has this ability is less autonomous (V=0) than one that does not share tasks (V=1).

The reactivity
The metric representing the reactivity depends on five criteria gathers in two categories:

1. Effectiveness of Perceptions Acquisition. It represents the agent's ability to use the sensors to perceive the environment. If an agent perceives all relevant elements of the environment, the value one is assigned, otherwise 0.
2. Rapidity of Response in a Timely Fashion. This metrics represents how fast is the response to environmental needs. It is assessed through four criteria:
   - *PercQual* represents the reactivity linked to perception processing. If an agent process crude perceptions and has to manage different perceptions or extensive perception sequence, the assigned value is 0 because it reduces the reactivity of an agent that has not this ability and whose the assigned value would be 1.
   - *DefBeh* represents the reactivity linked to the agent's actions, which are predefined by the designer. If the agent's actions are predefined, it would have one as a value because it would be more reactive than an agent that does not have predefined actions and would have 0 as value.
   - *InsMod* represents the reactivity according to the possession of an internal model of actions and intentions of other agents. With such a model, the value would be 0 because it decreases the reactivity. On the contrary to the absence of such a model that conducts to the assignment of the value 1.
   - *ComMin* represents the reactivity linked to the minimization of communication. This metric has value one if the mean number of messages exchanged to achieve a goal of an agent is equal to or less than the expected value; otherwise, the value is 0.

The adaptability
The metric representing the adaptability to the environment depends on five criteria gathers in two categories:

1. Ability to Manage Different Situations. This metric represents the ability to cope with different and unpredictable situations. It is expressed through three criteria.

   - *LearAb* represents the ability of learning. The presence of this ability assigns the value 1 for improving the adaptability, and its absence assigns the value 0;
   - *EurFinAb* represents the agent's ability to finding suitable heuristics for achieving goals or tasks. This metrics is estimated according to the average message send by an agent to obtain useful information. If this average number is higher than expected, the metric has the value 0, because that shows the agent needs other agents to find suitable heuristics; otherwise, the metric has the value 1;
   - *ExcManAb* represents the effectiveness of exceptions management. If the number of exceptions managed by an agent is superior to expected, then the metric has the value 1; otherwise, its value is 0.

2. Ability to Respond to new External Stimuli. This metric represents the ability to respond to changes in the environment. This metrics is estimated through two criteria.

   - *CorrChangReact* represents the correlation between the agent's reactions and changes in the environment. If there is a strong relationship between them, the value of the metric is 1; otherwise, the value is 0.
   - *RightRol* represents the ability to change roles according to changes in the environment. The presence of this ability assigns the value 1 to the metric. Its absence conducts to the assignment of the value 0.

The metric plan begins with the value estimation of each of these criteria and for each MAS agent. These criteria values are used to compute the metric of their category for each agent. Then, the metrics of categories are used to calculate the high-level metrics for each agent. Finally, the category metrics and high-level metrics are computed for the multi-agent system. The computation of these metrics is presented in the next section.

8.1.2 Computation of metrics

The measure of the designed MAS quality begins by the computation of the different high-level metrics both for each agent (M_{aj}) and for the overall multi-agent system (M_{MAS}). Therefore, M_{aj} and M_{MAS} are values that represent the degree of a high-level metric for each agent and for the overall multi-agent system, respectively: a value includes in [0-0,3] represents a low degree, a value includes in [0,3-0,6] represents a medium degree, and a value superior to 0,6 represents a high degree [Di Bitonto et al., 2012]. Finally, the quality of the multi-agent system is assessed according to the adequacy of the high-level metrics (e.g., rationality, autonomy, reactivity, and adaptability) degree with the environment complexity degree.

The metric for each agent (M_{aj}) is computed as the mean value of criteria values V_i characterizing the metric. This computation corresponds to the equation 8.1.

$$M_{aj} = \frac{\sum_{i=1}^{n} V_i}{n} \tag{8.1}$$

Finally, the MAS metric is computed as a mean value from the metric of each agent (M_{aj}) and the quantity of an agent (Q_j). This computation corresponds to the equation 8.2.

$$M_{MAS} = \frac{\sum_{j=1}^{n} (M_{aj} \times Q_j)}{\sum_{j=1}^{n} Q_j} \tag{8.2}$$

The application of this metric computation on the studied case is presented in the next section.

8.2 Evaluation application

The metric plan presented in [Di Bitonto et al., 2012] has been applied to the multi-agent system resulting from the model transformation on the use case. In this use case, three types of actor agents have been created: fireman, physician, and ambulanceman. There are also three types of manager agents: DSM (the Director of Medical Relief), COS (the Commander of Relief Operations), fireman officers, and a central agent, which is DOS (Director of Relief Operations). In addition to the 16 fireman officers discussed in the section 7.1.2, whose some specific officer as the collection officer (c.f. section 7.2.1.3), officer agents include the officer of the advanced medical post, which is not a fireman.

Table 8.1 shows the number of agents for each type of agent.

Agent category	Agent type	Agent quantity
Actor Agent	Fireman	28
	Physician	19
	Ambulanceman	11
Manager Agent	DSM	1
	COS	1
	officer	17
Central Agent	DOS	1
Total amount of Agent		78

Table 8.1: Table representing the number of agents by agent type

8.2.1 Evaluation of environment complexity

As presented previously (c.f. Section 8.1.1), the environment complexity is assessed according to three metrics: the inaccessibility, the instability, and the complexity of interactions.

The metric of inaccessibility depends on the inaccessibility of environment components (*CompInacc*). The actor agents have medium access because they have access only to relevant environment components to achieve their tasks. These relevant components are a manager agent (an officer) for the communication about task achievement and suitable locations as the disaster area (D), the advanced medical post (AMP), or the hospitals (H). On the contrary, the manager agents have access to all other agents and all points of interest (e.g., D, AMP, H). Therefore they have low inaccessibility.
It depends then on the resource inaccessibility (*ResInacc*). The primary resources of this studied case are the victims. All agents have access to the victims and their information.
Table 8.2 gathers values assigned to each criterion of inaccessibility and the result of metrics computed from these criteria. The inaccessibility metric shows that both all agents and the multi-agent system have low inaccessibility (i.e., the value includes in [0-0.3] [Di Bitonto et al., 2012]).

The metric of instability depends on the environment characteristics as the *time* and its *dynamic*. In the application case, the environment time is discrete, and the environment is dynamic. Indeed, at each step of the simulation, whereas agents act and make decisions, the state of the victims evolves at the same time.
Then the instability depends on the effects produced by agents (*NumEffeAct*). The actions of actor agents produce only one effect, whereas the decisions made by manager agents have several implications.
Table 8.3 gathers values assigned to each criterion of instability and the result of

Agent type	***CompInacc***	***ResInacc***	**Agent Inaccessibility**
Fireman	0,5	0	0,25
Physician	0,5	0	0,25
Ambulanceman	0,5	0	0,25
DOS	0	0	0,25
DSM	0	0	0,25
COS	0	0	0,25
Officer	0	0	0,25
Innaccessibility of Multi-Agent system			**0,19 (Low)**

Table 8.2: Table representing the assessment of environment inccessibility

the metrics computed from these criteria. The computation of instability intra-agent shows that actor agents have a medium instability (i.e., value includes in [0.3-0.6] [Di Bitonto et al., 2012]), whereas manager agents have a high instability (i.e., value greater than 0.6 [Di Bitonto et al., 2012]). Manager agents being less than actor agents, the inter-agent instability is medium.

Agent type	***Time***	***Dynamic***	***NumEffeAct***	**Agent Instability**
Fireman	0	1	0	0,33
Physician	0	1	0	0,33
Ambulanceman	0	1	0	0,33
DOS	0	1	1	0,67
DSM	0	1	1	0,67
COS	0	1	1	0,67
Officer	0	1	1	0,67
Instability of Multi-Agent system				**0,42 (Medium)**

Table 8.3: Table representing the assessment of environment instability

The interaction complexity metric depends on the degree of competition and cooperation between agents, but also on the need to use trust and reputation models. In the application case, all agents cooperate with at least one other agent. However, they are not in competition. They do not need to use trust and reputation models. Table 8.4 shows values assigned to each of these criteria and the result of the metrics computed from them. The computation of the interaction complexity metrics shows that both all agents and the multi-agent system have a high interaction complexity.

Based on these three metrics, the high-level metric representing the environment complexity intra and inter-agent has been computed. Table 8.5 presents the results of this high-level metric computation. It shows that both agents and the multi-agent system have a medium environment complexity.

Agent type	*CompGrad*	*CoopGrad*	*Tr&RepMod*	Interaction complexity of Agent
Fireman	0	1	1	0,67
Physician	0	1	1	0,67
Ambulanceman	0	1	1	0,67
DOS	0	1	1	0,67
DSM	0	1	1	0,67
COS	0	1	1	0,67
Officer	0	1	1	0,67
Interaction complexity of Multi-Agent system				**0,67 (High)**

Table 8.4: Table representing the assessment of interaction complexity in the environment

Agent type	Environment complexity of Agent	Environment complexity of Agent x quantity
Fireman	0,42	11,67
Physician	0,42	7,92
Ambulanceman	0,42	4,58
DOS	0,44	0,44
DSM	0,44	0,44
COS	0,44	0,44
Officer	0,44	7,56
Environment complexity of Multi-Agent system		**0,42 (Medium)**

Table 8.5: Table representing the assessment of environment complexity

The next section presents the high-level metric computation of rationality.

8.2.2 Evaluation of agent's rationality

The high-level metric of the rationality is computed from the metric of the action choice mode and the metric of success maximization.

The metric of the action choice mode depends on both the agent type according to the way for making action choice and their internal structure, the agent's ability to learn, and to build an action plan. Actor agents are reactive agents. Therefore they have a stimulus-response and low rationality. On the contrary, manager agents are cognitive and goal-based agents. Therefore they have more rationality. No agent can learn and build an action plan. They also have no internal model of actions and intentions of other agents. Although manager agents can assign tasks to actor agents, this process is managed through agent status and information exchange, but not by an internal model of action and intention of other agents. Table 8.6

presents the values of each criterion intervening in the metric computation of action choice mode.

Agent type	*AgType*	*PlaConstr*	*learAb*	*InsMod*	**ModChAct of Agent**
Fireman	0	0	0	0	0
Physician	0	0	0	0	0
Ambulanceman	0	0	0	0	0
DOS	1	0	0	0	0,25
DSM	1	0	0	0	0,25
COS	1	0	0	0	0,25
Officer	1	0	0	0	0,25
Mode of action choice (ModChAct) of Multi-Agent system (MAS)					**0,06 (Low)**

Table 8.6: Table representing the assessment of the mode of action choice

The metric of success maximization is 0 for all agents because they have a maximal success. This success is due to the no modeling of agents' emotions. In the case of the action plans application, the agents represent professionals who carry out the actions in an optimal way. Thus, not expressing feelings makes it possible to expect realism near to professional behaviors. If emotions were modeled, agents could behave more unpredictably and therefore have less success. However, the degree of authenticity of emotions is difficult to estimate for professionals and would, thus, diverge from reality. It is suitable to do not model emotion than model them badly to obtain the best realism.

Table 8.7 presents the values of each metric intervening in the computation of the high-level metric of rationality. The results of the metrics computation show that the rationality intra and inter-agent is low.

Agent type	**Mode of action choice**	**Sucess maximization**	**Rationality**
Fireman	0	0	0
Physician	0	0	0
Ambulanceman	0	0	0
DOS	0,25	0	0,125
DSM	0,25	0	0,125
COS	0,25	0	0,125
Officer	0,25	0	0,125
Multi-Agent system	**0,06 (Low)**	**0 (Low)**	**0,03 (Low)**

Table 8.7: Table representing the assessment of the rationality

The low rationality of agents is due to the goal of simulation that does not require high-rational agents but to test the rationality of action choices made during the preparedness by stakeholders. That is why, the rationality is low in the multi-agent system, the rationality comes into the simulation through the decision and preparation made by disaster management stakeholders.

8.2.3 Evaluation of agent's autonomy

The high-level metric of autonomy is computed from the proactivity metric and the autonomy metric in the organizational structure.

The proactivity metric depends on the agent's ability to change of role autonomously (*MoreRol*), to negotiate (*NegAg*), to diagnose errors/problems (*DiaErPrAb*), and to communicate independently (*ComAutAb*). All agents have only one role; therefore, they are not able to change of role autonomously. Actor agents do not negotiate with other agents. However, manager agents negotiate between them for resource distribution. The actor agents can detect problems when they do not succeed in a task. They communicate these problems to their manager agent. On the contrary, the manager agents do not know if their decisions are right or not. They only receive feedback on the achievement of their choice through actor agents. All agents can communicate autonomously. Table 8.8 presents the values of each criterion intervening in the metric computation of proactivity.

Agent type	***MoreRol***	***NegAg***	***DiaErPrAb***	***ComAutAb***	**Proactivity of Agent**
Fireman	0	0	1	1	0,5
Physician	0	0	1	1	0,5
Ambulanceman	0	0	1	1	0,5
DOS	0	1	0	1	0,5
DSM	0	1	0	1	0,5
COS	0	1	0	1	0,5
Officer	0	1	0	1	0,5
Proactivity of Multi-Agent system					**0,5 (Medium)**

Table 8.8: Table representing the assessment of the proactivity

The metric of autonomy in the organizational structure depends on task sharing ability (*SharTask*) and the position of a subordinate (*PosStr*). All actor agents are the subordinate of an officer. Officer agents are also subordinates of the COS agent or the DSM agent. Therefore, only the DOS agent, COS agent, and the DSM agent are not at a subordinate position. No agent shares tasks with other agents. Each agent has its own assigned task and makes them independently from the functions of other agents. The absence of task sharing has the advantage to do not need processing time for task distribution that improves the reactivity and autonomy of agents. Table 8.9 presents the values of each criterion intervening in the metric computation of the autonomy in the organizational structure.

Agent type	*PosStr*	*SharTask*	AutoOrg of Agent
Fireman	0	1	0,5
Physician	0	1	0,5
Ambulanceman	0	1	0,5
DOS	1	1	1
DSM	1	1	1
COS	1	1	1
Officer	0	1	0,5
Autonomy in the organizational structure (AutoOrg) of Multi-Agent system			**0,52 (Medium)**

Table 8.9: Table representing the assessment of the autonomy in the organizational structure

Table 8.10 presents the values of each metric intervening in the computation of the high-level metric of autonomy. The results of the metrics computation show that the independence of actor agents and officer agents is medium. In contrast, the independence of the DOS agent, COS agent, and the DSM agent is high. These metrics intra-agent conduct to a medium autonomy of the multi-agent system.

Agent type	Proactivity	Autonomy in the Organizational Structure	Autonomy
Fireman	0,5	0,5	0,5
Physician	0,5	0,5	0,5
Ambulanceman	0,5	0,5	0,5
DOS	0,5	1	0,75
DSM	0,5	1	0,75
COS	0,5	1	0,75
Officer	0,5	0,5	0,5
Multi-Agent system	**0,5 (Medium)**	**0,52 (Medium)**	**0,51 (Medium)**

Table 8.10: Table representing the assessment of the autonomy

8.2.4 Evaluation of agent's reactivity

The high-level metric of the reactivity is computed from two metrics: one that assesses the effectiveness of perceptions acquisition and another that assesses the rapidity of response in a timely fashion.

The perception effectiveness of actor agents is high because they perceive the environment directly around them, whereas manager agents perceive the environment indirectly from the communication with other agents. Therefore, manager agents have low perception effectiveness.

The metric for the rapidity of response in a timely fashion depends on the agent's perception (*PercQual*), behavior (*DefBeh*), internal model (*InsMod*), and communication (*ComMin*). All agents have access to all the information that they need. Therefore they do not need a complex perception process. Even, actor agent that can perceive their surrounding environment uses this perception as checking of some specific aspects, they do not choose the most significant perceptions or aggregate large perception sequences. The goal of the simulation is the application of prepared action plans; the behavior of an agent is predefined before the simulation to correspond to the action plans. As said in the evaluation of the rationality, agents have no internal model of actions and intentions of other agents. All agents have minimal communication. Actor agents and officers exchange messages with their supervisor only, if they fail in a task. The other agents communicate only to assign tasks and negotiate resources. Table 8.11 presents the values of each criterion intervening in the metric computation of response rapidity in a timely fashion.

Agent type	*PercQual*	*DefBeh*	*InsMod*	*ComMin*	Response rapidity of Agent
Fireman	1	1	1	1	1
Physician	1	1	1	1	1
Ambulanceman	1	1	1	1	1
DOS	1	1	1	0	0,75
DSM	1	1	1	0	0,75
COS	1	1	1	0	0,75
Officer	1	1	1	1	1
Response rapidity of Multi-Agent system					**0,99 (High)**

Table 8.11: Table representing the assessment of the response rapidity in a timely fashion

Table 8.12 presents the values of each criterion and metrics intervening in the computation of the high-level metric of the reactivity. The results of metrics computation show that the responsiveness of actor agents is high, whereas the reactivity of the other agents is medium. These metrics intra-agent conduct to the high reactivity of the multi-agent system.

Agent type	Effectiveness of perceptions acquisition	Rapidity of response in a timely fashion	Reactivity
Fireman	1	1	1
Physician	1	1	1
Ambulanceman	1	1	1
DOS	0	0,75	0,375
DSM	0	0,75	0,375
COS	0	0,75	0,375
Officer	0	1	0,5
Multi-Agent system	**0,74 (High)**	**0,99 (High)**	**0,87 (High)**

Table 8.12: Table representing the assessment of the reactivity

8.2.5 Evaluation of agent's adaptability to the Environment

The high-level metric of the adaptability is computed from the metrics that assess the ability to manage different situations and the ability to respond to new external stimuli.

The ability to manage different situations depends on the agent's skills of learning (*LearAb*), finding suitable heuristics for achieving goals or performing tasks (*EurFinAb*) and handling exceptions (*ExcManAb*). All agents have not the ability to learn and of handling exceptions. Agents only ask for information that is strictly necessary for the environment, such as the position of the victims. Thus the number of messages exchanged with the environment is the same as expected (e.g., a target, a request for victim search). Therefore, agents can find suitable heuristics for achieving goals or performing tasks. Table 8.13 presents the values of each criterion intervening in the metric computation of the ability to manage different situations.

Agent type	*LearAb*	*EurFinAb*	*ExcManAb*	Ability of Agent to manage different situations
Fireman	0	1	0	0,33
Physician	0	1	0	0,33
Ambulanceman	0	1	0	0,33
DOS	0	1	0	0,33
DSM	0	1	0	0,33
COS	0	1	0	0,33
Officer	0	1	0	0,33
Ability of Multi-Agent system to manage different situations				**0,33 (Medium)**

Table 8.13: Table representing the assessment of the ability to manage different situations

The ability to respond to new external stimuli depends on the ability of reactions (*CorrChangReact*) and roles change according to the changes in the environment (*RightRol*). The agents have only one role and are not able to change of role. The adaptation of agents to the environment is limited. The purpose of this limitation is to test the plans prepared. It assesses whether the preparation is sufficient to allow adaptation to a variety of situations. If the agents could adapt and thus transgress or transform the prepared plan, the results provided for the simulation would not be representative of the "strict" application of the plan. For this reason, the agent's adaptive capacity depends on the plans prepared, rather than on their capacity. The actor agents have only the ability to identify problems in the achievement of their task to notify the plan fails to its supervisor, which can decide to trigger another plan. Table 8.14 presents the values of each criterion intervening in the metric computation of the ability to respond to new External stimuli.

Agent type	*CorrChangReact*	*RightRol*	Ability of Agent to respond to new External stimuli
Fireman	0,5	0	0,25
Physician	0,5	0	0,25
Ambulanceman	0,5	0	0,25
DOS	0	0	0
DSM	0	0	0
COS	0	0	0
Officer	0	0	0
Ability of Multi-Agent system to respond to new External stimuli			**0,19 (Low)**

Table 8.14: Table representing the assessment of the ability to respond to new External stimuli

Table 8.15 presents the values of each criterion and metrics intervening in the computation of the high-level metric of the adaptability. The results of metrics computation show the agents have a medium ability to manage different situations and a low ability to respond to new external stimuli. Therefore, both the adaptability intra-agent and inter-agent is low.

Agent type	Ability to manage different situations	Ability to response to new external stimuli	Adaptability
Fireman	0,33	0,25	0,29
Physician	0,33	0,25	0,29
Ambulanceman	0,33	0,25	0,29
DOS	0,33	0	0,17
DSM	0,33	0	0,17
COS	0,33	0	0,17
Officer	0,33	0	0,17
Multi-Agent system	**0,33 (Medium)**	**0,19 (Low)**	**0,26 (Low)**

Table 8.15: Table representing the assessment of the adaptability

8.3 Discussion

The application of the metric plan to our use case has allowed measuring the environment complexity of the multi-agent system and several characteristics of the multi-agent system. These measurements provide information about the multi-agent system but do not allow its assessment. Guidelines offered by [Di Bitonto et al., 2012] aims at supporting the comparison between the measurement of the environment complexity and the characteristics of the multi-agent system. This comparison allows the assessment of the multi-agent system design by assessing its adequacy to the environment where it operates. Therefore, the next sub-sections compare each measure of a MAS characteristic with the measurement of the environment's complexity. These comparisons aim at concluding on their adequacy.

8.3.1 Rationality vs. Environment Complexity

The rationality of the MAS has been assessed according to the success maximization and the action choice mode. These two metrics have been compared with the three metrics characterizing the environment complexity.

Guidelines specify that in a stable and accessible environment, the success maximization must be high. A medium level of success maximization could be acceptable only in case of high inaccessibility or instability of the environment. It is due to the difficulty of obtaining a high one. Therefore, the low level of success maximization of our system is not appropriate.

The action choice mode has been analyzed according to the instability and the interaction of the environment's complexity. In case of a high interaction complexity of the environment, a high level of action choice mode is expected; however, our system has a low action choice mode. Guidelines explain that a high action choice mode would be preferable for a high instability of the environment. However, a medium level is acceptable to make a compromise with the response time to the environment. In the case of low instability, the level of action choice mode has no constraints. Our system has a medium stable environment with a low action choice mode; this one is not in the worst configuration (highly unstable environment vs. low action choice mode). Therefore we can consider this relationship as acceptable. Finally, guidelines specify that a medium and low level of action choice mode is acceptable for a medium or low inaccessibility level of the environment. Therefore, our MAS respects this adequacy.

Table 8.16 summarizes the adequacy assessment between the characteristics of the rationality and those of the environment's complexity. To provide a value of adequacy assessment of the rationality, we have given 1 point for comparison respecting the guidelines, 0.5 for acceptable results, and 0 points for the no respect of the instructions. Thus, the adequacy assessment obtains 1.5 on 4 points, corresponding to a percentage of 37.5% of rationality adequacy to the environment. This assessment highlights a lack of rationality in the multi-agent system.

Environment complexity vs. *Characteristics of rationality*				
Environment complexity			***Success maximization***	
Inaccessibility	**Instability**	**Interaction complexity**	**Expected**	**Obtained**
low	medium	-	high	low
Environment complexity			***Action choice mode***	
Inaccessibility	**Instability**	**Interaction complexity**	**Expected**	**Obtained**
low	-	-	low/medium	low
-	medium	-	-	low
-	-	high	high	low
low	medium	high	-	low

Table 8.16: Table representing the adequacy assessment of the MAS rationality to its environment

8.3.2 Autonomy vs. Environment Complexity

Guidelines assess the adequacy of the MAS autonomy to the inaccessibility and interaction complexity of its environment. It is done according to the proactivity metric. In the case of the high complexity of interaction in the environment, the expected proactivity is medium or high. The medium proactivity of the MAS is adequacy to the high interaction complexity of the environment. Guidelines specify that a low level of proactivity is acceptable in case of low inaccessibility. The medium proactivity of the MAS is adequacy to the low inaccessibility of the environment.

Table 8.17 summarizes the assessment of the MAS proactivity adequacy to its environment. This assessment shows the complete adequacy of the MAS autonomy to its environment.

Environment complexity vs. *Characteristic of autonomy*				
Environment complexity			***Proactivity***	
Inaccessibility	**Instability**	**Interaction complexity**	**Expected**	**Obtained**
low	-	-	low is acceptable	medium
-	-	high	medium/high	medium

Table 8.17: Table representing the adequacy assessment of the MAS autonomy to its environment

8.3.3 Reactivity vs. Environment Complexity

The reactivity of the MAS depends on the perception's effectiveness and the response effectiveness in a timely fashion.

Guidelines specify that the perception's effectiveness is related to all aspects of the environment and that whatever their metrics, the perception effectiveness must be high—this is the case of the studied MAS.

The response effectiveness in a timely fashion depends only on the stability of the environment; if this one is high, the MAS must have a rapidity of response. The studied MAS has only a medium instability but has a great answer in a timely fashion that provides functional adequacy to its environment.

Table 8.18 summarizes the MAS reactivity adequacy to its environment. This assessment shows the complete adequacy of the MAS reactivity to its environment.

Environment complexity vs. *Characteristics of reactivity*				
Environment complexity			***Perception effectiveness***	
Inaccessibility	**Instability**	**Interaction complexity**	**Expected**	**Obtained**
-	-	-	high	high
Environment complexity			***Respond in a timely fashion***	
Inaccessibility	**Instability**	**Interaction complexity**	**Expected**	**Obtained**
-	medium	-	medium/high	high

Table 8.18: Table representing the adequacy assessment of the MAS reactivity to its environment

8.3.4 Adaptability vs. Environment Complexity

The adaptability of the MAS depends on its response to new external stimuli and its management of different situations.

The response to new external stimuli depends on the inaccessibility of the environment. Both of them must have the same level. The MAS has low environment inaccessibility and a low response to new external stimuli. Therefore, its response is in adequation to its environment inaccessibility.

The management of different situations is related to all aspects of the environment complexity. For each of these aspects, the management of various situations must have the same level. The environment complexity being defined through low inaccessibility, a medium instability, and high interaction complexity, the level of different situations management cannot satisfy the level of these three aspects in the same time but must have a compromising level. The average of these three aspects providing a medium environment complexity, the best level of management of different situations is a medium level, which is the level of management of the studied MAS. Therefore, the MAS management of different situations is adequacy to its environment.

Table 8.19 summarizes the MAS adaptability adequacy to its environment. This assessment shows the complete adequacy of the MAS adaptability to its environment.

Environment complexity vs. *Characteristics of adaptability*				
Environment complexity			*Respond to new external stimuli*	
Inaccessibility	**Instability**	**Interaction complexity**	**Expected**	**Obtained**
low	-	-	low	low
Environment complexity			*Manage different situations*	
Inaccessibility	**Instability**	**Interaction complexity**	**Expected**	**Obtained**
low	-	-	low	medium
-	medium	-	medium	medium
-	-	high	high	medium
medium			medium	medium

Table 8.19: Table representing the adequacy assessment of the MAS adaptability to its environment

8.3.5 Conclusion

The assessment of each MAS's characteristic according to the environment complexity has shown that three characteristics of the MAS are complete adequacy (100%) to its environment, and one has only 37.5% of suitability to its environment. Therefore, the MAS obtains an average of 84,38% of adequacy to its environment.

This evaluation shows a limit of the MAS at the level of its rationality. However, this limit is compensated by the complete adequacy of the three other MAS characteristics. Thus, the overall MAS has the right level of adequation to its environment.

This proper level of adequation to its environment allows the validation of the multi-agent system's design choice. It also confirms the combination of the model transformation processes and generative programming, which has produced the multi-agent system adequacy to its environment complexity.

9 Conclusion

Contents

9.1 Contributions **278**

9.1.1 Multi-agent simulation adaptation 279

9.1.2 Plan representation 280

9.1.3 Representation of the plan's effectiveness associated with its applicability context 280

9.2 Discussions **281**

9.3 Future work **283**

9.3.1 Short term perspective 284

9.3.2 Medium-term perspective 285

9.3.3 Long term perspective 285

Although many simulations for emergency management exist, each time the disaster management community wants to use simulations for preparing disaster management. They must work actively with a computer expert to define a simulation model corresponding to their goal. Moreover, in addition to the difficulty of understanding between disaster management experts and computer scientists to create the model, they must face interoperability between their information system and the simulation system. Indeed, these interoperability problems arise to provide inputs of the simulation and integrate the results of the simulation into the information system [Balasubramanian et al., 2006]. This thesis has proposed an approach based on knowledge to overcome these issues. The first section summarizes this approach and highlights the main contributions of this thesis. The second section presents the advantages and limits of the approach. Finally, the third section presents future works.

9.1 Contributions

This thesis proposes an approach of knowledge-driven multi-agent simulation engineering for assessing the effectiveness of disaster management plans that answers the problem question presented in Chapter 2.

The SemDM and SemMAS ontologies contained in the knowledge base are interlinked to perform simulation conceptualization through reasoning. This simulation conceptualization aims at generating simulation models from the disaster management plan representation. It produces an independent-platform simulation model called the conceptual simulation model, and a specific-platform simulation model called the programming simulation model. The programming model, specific to the platform *GAMA*, is the input of a generative programming process that implements the simulation program and experiments. The results of simulation experiments are then analyzed to assess plan effectiveness and enrich the knowledge base with the assessment.

The stepwise process of simulation conceptualization, generative programming, and simulation results analysis to enrich the knowledge base by plan assessment is initiated by expert knowledge. The expert knowledge is modeled through a web interface to facilitate their integration into the knowledge base and the automatic integration of knowledge extracted from their data. The knowledge contained in the SemDM ontology can be retrieved through SPARQL queries or visualize through the client interface to support the disaster management community. The proposed approach brings contributions in three topics, whose limits must be overcome to solve the problem question (c.f. section 2.2.1). These three topics and their associated contributions are the following:

1. Multi-agent simulation adaptation
    - SemMAS ontology,
    - Knowledge-driven simulation modeling,
    - Knowledge-driven generative simulation programming, and
    - GAMA Skills extension,
2. Plan representation
    - SemDM ontology,
    - Automatic integration of knowledge extracted from data,

3. Representation of the plan's effectiveness associated to a situation definition that characterized its value
    - Clustering-based analysis for plan's effectiveness representation.

9.1.1 Multi-agent simulation adaptation

SemMAS ontology The SemMAS ontology allows the representation of disaster management simulation modeling and design. This ontology provides high-level concepts describing the domain of multi-agent simulation modeling and design through concepts representing the different models and the experiments' configuration. It also provides low-level concepts, more specific to the application domain, disaster management, and the chosen simulation platform, the GAMA platform. The simulation modeling through ontology facilitates the shareability and reusability of simulation models.

Knowledge-driven simulation modeling The approach of knowledge-driven simulation modeling generates the conceptual and programmed simulation model in the SemMAS ontology. It generates them according to the SemDM ontology content representing disaster management plans. This automatic process provides a uniform method of simulation conceptualization for different disaster management plans. This uniform method has the advantage of avoiding the insertion of different conceptualization biases for different simulations modeling. This approach is managed through reasoning, mainly based on SHACL rules. A set of rule built-ins have been developed to increase the rule-based reasoning capabilities. These built-ins have the advantage of being reusable by other approaches. These built-ins provide the capability to generate URI to create new instances from other existing instances or concepts.

Knowledge-driven generative simulation programming The approach of knowledge-driven simulation programming generates the simulation code to execute it with the GAMA platform. The generative programming uses SPARQL queries to retrieve information about the programming model from the SemMAS ontology in a specific order to write the code step by step. This automatic process provides a uniform method to generate the GAML code from different GAML programmed models represented in the SemMAS ontology. This uniform method has the advantage of avoiding the insertion of different programming biases for different simulation codes.

GAMA Skills extension The knowledge-driven simulation modeling and programming are based on a set of pre-implemented agent skills. These skills are used to program the agent's behavior. An extension of skills specific to disaster management action has been developed. This extended skill allows agent's actions as transport, care, or health assessment. These new skills used by the generative programming are represented into the SemMAS ontology. They have the advantage of being reusable for other simulation implementations in GAML.

9.1.2 Plan representation

SemDM ontology The SemDM ontology allows the representation of disaster management plans and their effectiveness. It is composed of high-level concepts that provide the structure to describe a plan and its associated knowledge, whose effectiveness. It also contains low-level concepts that specify action and resource types achieved and used on the ground. The representation of disaster management plans through an ontology is an advantage for the modeling and understanding the disaster management community. It is also an advantage for the interoperability with other systems. It allows the knowledge and information exchange with other ontology-based systems, which are well developed in this domain (c.f. ontologies presented section 3.3.1) and other systems thanks to uplift and dowlift process.

Automatic integration of knowledge extracted from data The automatic integration of knowledge extracted from data is based on natural language processing, geospatial dimension, and the Semantic Web. This process facilitates the interoperability between systems by interpreting heterogeneous geospatial data. It produces an RDF Graph representing the knowledge extracted from data, which is interlinked with Wikidata concepts. This interlinking with Wikidata offers perspectives of data enrichment by the Semantic Web and enhancement of the Semantic Web by data to the Semantic Web community. Furthermore, this process has shown a capability to extract knowledge from open data, for which no information is available, near to human capabilities (c.f. [Prudhomme et al., 2020a]).

9.1.3 Representation of the plan's effectiveness associated with its applicability context

Clustering-based analysis for plan's effectiveness representation The results of simulation allow computing effectiveness and providing a set of observed variable values. The clustering-based analysis allows segmenting the plan's effectiveness values. In the case of a unique segment, the plan is represented as having global applicability. Otherwise, it is represented as having specific applicability with several effectiveness values related to a situation. In the case of specific applicability, situations related to the value are defined through the variables impacting the plan's effectiveness. This impact is identified through the analysis of different clustering results. The computation of effectiveness value and its definition related to the situation in the case of a value diversity allows providing a detailed assessment of plans.

The approach presented in this thesis results in a system able to support disaster management preparedness thanks to its ability of knowledge enrichment from experiences provided by simulations. These simulation experiences are generated from the disaster management plan representation and analyzed automatically. This automatic approach provides a plan's effectiveness assessment compared with other plan's effectiveness obtained by this approach. This comparison is possible thanks to the uniform method of simulation modeling and design that avoids the insertion of biases between different simulations. This approach brings several contributions and perspectives in different domains (i.e., Disaster management, Semantic Web, and Multi-Agent Simulation). Its advantages and limits are presented in the next section.

9.2 Discussions

To support the disaster management community, the main advantages of the proposed approach are interoperability, reusability, adaptability, and flexibility.

The interoperability The use of Knowledge-based multi-agent simulation facilitates the interoperability between this system and other systems. However, it is mainly the automatic integration process that provides the interoperability of the system. Indeed, it has the advantage of integrating information from heterogeneous geospatial data. These heterogeneous data can have a large variety of information content and can come from different databases, shapefile data of the disaster management community, or open sources. This process, combined with the downlift methods developed in the SemGIS project [Homburg et al., 2018], provides the interoperability of this system with external systems. The combination of the knowledge-based architecture, knowledge-based generative programming, and the simulation analysis for plan assessment also provides the advantage of internal interoperability between the knowledge base and the simulation platform. This architecture and processes allow an exchange from the knowledge base to the simulation platform and from the simulation platform to the knowledge base.

The reusability and sharing The modeling choice of a conceptual model aims at allowing the reusability and the sharing of the multi-agent model. The conceptual model of multi-agent simulation has the advantage of being platform-independent and can thus be reused by others to test other implementation and other platforms. Moreover, the use of an ontology to represent the simulation model allows the reuse of the model for further usages [Benjamin et al., 2006]. It overcomes the limits of model sharing in the simulation community, as highlighted by [Lacy and Gerber, 2004, Miller et al., 2004].

The adaptability to different scenarios According to the interpreted data and expert knowledge, the system's architecture allows integrating different instances of a disaster management model. It then uses the instances of the disaster management model to generate instances of the multi-agent simulation model. The programming and the execution of simulations allow, finally, the enrichment of the disaster management model instances.

The flexibility of extension Traditionally, to simulate another disaster management model, disaster management experts must explain this new model to computer scientists that design a conceptual simulation model in continuous discussion with experts. When both are satisfied by the conceptual model, the computer scientist needs to program the simulation and experiments to execute. Finally, they analyze simulation results to answer questions of disaster management experts. The approach proposed in this thesis reduces this chain of development to the modeling of the new model by disaster management experts. This simplification of the process is possible, thanks to the automatization of simulation modeling, development, and analysis from a disaster management model.

Although semantic modeling requires some specific knowledge, it has the advantage to be easily understandable both by humans and machines. Therefore, a web interface has been developed to guide non-expert in semantic to design a model. This interface provides explicit concept definitions and the structure of modeling to follow thanks to the high-level disaster management concepts, which are common to all disaster management models. An extensive set of actions, representing the most common actions intervening during a crisis and inspired by the emergel ontology, have been provided. These action instances have been linked to the agent's behaviors, whose programming code structure has been implemented. The high and low-level of the SemDM ontology provides a diversified base to design different disaster management models.

The limits of the system are linked to minimum choices of implementation, aiming to prove the concept of the approach, but not to provide an optimal system.

Lack of rationality in the multi-agent simulation The evaluation of the system has highlighted the first limit of the system. This limit is a lack of rationality in the multi-agent system. Indeed, the rationality of agents is linked to their cognitive abilities. In the multi-agent simulation model, the majority of agents (represented by actors agents) are reactive agents rather than cognitive agents. Reactive agents have been preferred to cognitive agents for representing actor agents. This choice aims at providing a high reactivity of the system and allowing large-scale simulation for optimizing not only one plan, but a set of plans from different stakeholders of disaster management. However, even cognitive agents (central and manager agents) have low rationality due to delimited behaviors according to the disaster management model.

Limit of current implementation The system's flexibility is limited to disaster management models using traditional actions of crisis such presented into emergel ontology. Indeed, the definition of a model using other actions would require the extension of the system by adding new crisis actions and their interlinking to the agent's behaviors for allowing the simulation conceptualization and generative programming. The limit of this extension is not accessible for the disaster management community and requires a computer scientist's intervention.

The presented approach offers good adaptability, flexibility, and interoperability to support disaster management preparedness. Its design has been thought to be extended to an application to large-scale with several plans from different stakeholders. Its design also facilitates the reusability and the sharing of the multi-agent model. This approach also has some limits, whose overcoming would increase its benefits. Therefore, the identified limits offer perspectives for the evolution and improvement of the approach. The next section presents these perspectives with future work.

9.3 Future work

The work presented in this thesis offers many perspectives. These perspectives are classified into three categories:

- short term: corresponding to model and system improvement,
- medium-term: corresponding to system extension,
- long term: corresponding to future projects based on this thesis.

9.3.1 Short term perspective

Although the simulation model used in this thesis is sufficient to prove the concept of the proposed knowledge-based approach to evaluate the disaster management plan, two main additional works would be advisable to reinforce the robustness of the approach. The first work concerns the simulation models validation, including the conceptual model with disaster management experts and the programmed model with GAMA platform experts. The second work concerns the improvement of the agent model rationality to reinforce the multi-agent model's weakness through the model evaluation made in chapter 8.

Model validation with disaster management experts The validation of the simulation model will require to compare simulation results with real results based on the same configuration to judge the accuracy of the model and adjust it, if necessary. Such a validation process can be achieved with data obtained from a real exercise or a real intervention of disaster management. It requires an agreement with disaster management experts to obtain access to these data.

Programmed model validation with GAMA platform experts The second validation will be to verify the interlinking between the multi-agent simulation concepts and those of a GAML model with GAMA's expert. Such verification will improve the generative programming process in GAML and thus could offer the possibility to integrate this new capability of knowledge-driven programming to the GAMA platform.

Improvement of agent model rationality The improvement of the cognitive agent's rationality will be to add them to the ability of learning and plan building. These abilities adding will result in a high level of rationality (0,88) for manager agents and can be added in the context of plan failure. Indeed, agents could memorize the cause of the plan fails and thus learn the situation plan is not adapted. Then, the manager could generate a new plan from the gathering of all actions composing its plans. The retrieved set of actions with constraints will then be used as a base for a new planning process, such as a hierarchical task network planning as used by [De La Asunción et al., 2005b]. It will also require some changes at the level of actor agent implementation that will need to have behavior defined according to a variable rather than the element considering in the plan to allow a manager to configure and assign new tasks. This change will indirectly impact the actor agent's rationality, which would have a behavior different from the plan due to the new assignment of tasks by their manager. These changes will increase the global rationality of the multi-agent system. Moreover, the memorization of failure cause and new planning could be integrated into the knowledge base to provide further information and propose new plans for the disaster management community.

9.3.2 Medium-term perspective

The medium-term perspectives correspond to the extension of the current system by increasing the capabilities of the simulation through the integration of other existing models.

Extension with disaster models It would be interesting to extend the system with models to simulate disasters such as floods or bushfires. Such an extension would allow the simulation of disaster management in a more complex scenario. It would enable users to set different types of disaster situations according to estimated risks for a commune, such as a flood from heavy rain, dam break, an industrial disaster with a repartition of toxic clouds, or a pandemic. This extension would require (1) representing these disaster models into the SemMAS ontology, (2) adding rules to allow the configuration of these models from a disaster description represented in the SemDM ontology, and (3) adding the models to the model library of the architecture illustrated in chapter 5.

Extension with other existing models The integration of further models as crowd move or traffic could bring an added value, as for simulating evacuation plans. The integration of further models would require the similar steps than the integration of a disaster model, i.e., (1) representing the model to add into the SemMAS ontology, (2) adding rules to allow the configuration of the model from the SemDM ontology content, and (3) adding the models to the model library.

9.3.3 Long term perspective

The long term perspectives concern two projects that can be built based on the approach proposed in this thesis: a learning support system to train fire officers and a recommendation system based on plans' effectiveness.

Learning support system to train fire officers The work presented in this thesis offers the possibility to provide a system able to support learning for training in fire officer schools through studying the impact of their decision-making. It would be possible to prepare a disaster scenario represented in the SemDM ontology and provided it to the fire officers to train. Based on the scenario, each fire officer can elaborate its plan and represent it into the SemDM ontology to assess the elaborated plans' effectiveness. Such usage would allow fire officer to assess their capabilities in the elaboration of effective plans, but also to compare different plans.

For such a project, it would be interesting to integrate and interpret the fire officer's tactical map that represents the plan elaborated from their situation analysis. The integration of knowledge extracted from the tactical map in the SemDM ontology would facilitate the representation of a plan for a fire officer.

Recommendation system based on the plan's effectiveness The approach presented in this thesis allows creating a knowledge base on the plans' effectiveness. Such a knowledge base would result from a complete preparation, during which the plans would be elaborated and assessed through the approach proposed in this thesis. Based on it, the disaster management community knows the plan's effectiveness according to situation criteria. Such a knowledge base can thus be used to know what plan to apply in what situation. Therefore, it would be interesting to develop a recommendation system based on the assessed plan's effectiveness. Such a system could match a current situation description with a situation description of the highest specific plan's effectiveness to identify the most suited plan to achieve.

Bibliography

[Ackoff, 1989] Ackoff, R. L. (1989). From data to wisdom. *Journal of applied systems analysis*, 16(1):3–9.

[AFKzV, 1999] AFKzV (1999). FwDV 100 - Führung und Leitung im Einsatz. In *Führung und Leitung im Einsatz*. AFKzV - Ausschuss Feuerwehrangelegenheiten Katastrophenschutz und zivile Verteidigung.

[ALT and GODAU, 1995] ALT, H. and GODAU, M. (1995). Computing the Fréchet Distance Between Two Polygonal Curves. *International Journal of Computational Geometry & Applications*, 05(01n02):75–91.

[Anderson, 1999] Anderson, C. V. (1999). The Federal Emergency Management Agency. *Disaster Prevention and Management: An International Journal*, 8(1):499–509.

[Antoniou and Van Harmelen, 2004] Antoniou, G. and Van Harmelen, F. (2004). Web ontology language: Owl. In *Handbook on ontologies*, pages 67–92. Springer.

[Arai et al., 2011] Arai, K., Harsono, T., and Basuki, A. (2011). Cellular Automata for Traffic Modelling and Simulations in a Situation of Evacuation from Disaster Areas. *Cellular Automata - Simplicity Behind Complexity*, page 193.

[Arbeitskreises V, 2003] Arbeitskreises V (2003). Hinweise zur Bildung von Stäben der administrativ – organisatorischen Komponente (Verwaltungsstäbe - VwS). Technical report, Bundesamt für Bevölkerungsschutz und Katastrophenhilfe. Arbeitskreises V: Feuerwehrangelegenheiten, Rettungswesen, Katastrophenschutz Und Zivile Verteidigung. Der Ständigen Konferenz Der Innenminister Und –Senatoren Der Länder (Ak V).

[Arenas et al., 2011] Arenas, M., Bertails, A., Prud'hommeaux, E., and Sequeda, J. F. E. (2011). A Direct Mapping of Relational Data to RDF. *W3C Working Draft 20 September 2011*, 27(September 2012):1–11.

[Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A nucleus for a Web of open data. In *Lecture Notes* *in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4825 LNCS, pages 722–735. Springer.

[Auer et al., 2009] Auer, S., Lehmann, J., and Hellmann, S. (2009). LinkedGeoData: Adding a spatial dimension to the Web of data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5823 LNCS, pages 731–746. Springer.

[Awad and Ghaziri, 2004] Awad, E. and Ghaziri, H. (2004). *Management Information Systems: Managing the Digital Firm 9th edn.* Pearson Education International, Upper Saddle River, NJ.

[Baader and Nutt, 2010] Baader, F. and Nutt, W. (2010). Basic Description Logics. In *The Description Logic Handbook*, pages 47–104.

[Babitski et al., 2011] Babitski, G., Bergweiler, S., Grebner, O., Oberle, D., Paulheim, H., and Probst, F. (2011). SoKNOS - Using semantic technologies in disaster management software. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6643 LNCS(PART 2):183–197.

[Balasubramanian et al., 2006] Balasubramanian, V., Massaguer, D., Mehrotra, S., and Venkatasubramanian, N. (2006). DrillSim: A simulation framework for emergency response drills. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3975 LNCS, pages 237–248. Springer.

[Bansal and Kagemann, 2015] Bansal, S. K. and Kagemann, S. (2015). Integrating Big Data: A Semantic Extract-Transform-Load Framework. *Computer*, 48(3):42–50.

[Barkaoui et al., 2016] Barkaoui, H., Guinet, A., Wang, T., and Meskens, N. (2016). Les plans de gestion de crises dans les pays francophones. In *"8ème édition de la conférence Gestion et Ingénierie des Systèmes Hospitaliers, GISEH", casablanca (du 11/07/2016 au 13/07/2016)*.

[Barros et al., 2015a] Barros, R., Kislansky, P., Salvador, L., Almeida, R., Breyer, M., and Gasparin, L. (2015a). EDXL-RESCUER ontology: Conceptual model for semantic integration. In *ISCRAM 2015 Conference Proceedings - 12th International Conference on Information Systems for Crisis Response and Management*, volume 2015-Janua.

[Barros et al., 2015b] Barros, R., Kislansky, P., Salvador, L., Almeida, R., Breyer, M., Pedraza, L. G., and Vieira, V. (2015b). EDXL-RESCUER ontology: An update based on faceted taxonomy approach. *CEUR Workshop Proceedings*, 1442.

[Battle and Kolas, 2011] Battle, R. and Kolas, D. (2011). Geosparql: enabling a geospatial semantic web. *Semantic Web Journal*, 3(4):355–370.

[Bayram et al., 2012] Bayram, J. D., Kysia, R., and Kirsch, T. D. (2012). Disaster Metrics: A Proposed Quantitative Assessment Tool in Complex Humanitarian Emergencies - The Public Health Impact Severity Scale (PHISS). *PLoS Currents*.

[BBK, 2015] BBK (2015). Grundlagen Krisenmanagement. Technical report, Bundesamt für Bevölkerungsschutz und Katastrophenhilfe.

[Beckett, 2004] Beckett, D. (2004). RDF/XML Syntax Specification. *W3C*, 10(2.3).

[Benaben et al., 2008] Benaben, F., Hanachi, C., Lauras, M., Couget, P., and Chapurlat, V. (2008). A metamodel and its ontology to guide crisis characterization and its collaborative management. In *Proceedings of ISCRAM 2008 - 5th International Conference on Information Systems for Crisis Response and Management*, pages 189–196.

[Beneito-Montagut et al., 2013] Beneito-Montagut, R., Shaw, D., and Brewster, C. (2013). Web 2.0 and Social Media in Disaster Management: Using Web 2.0 applications and Semantic Technologies to strengthen public resilience to disasters. *UK: Aston University*, page 174.

[Benjamin et al., 2006] Benjamin, P., Patki, M., and Mayer, R. (2006). Using ontologies for simulation modeling. In *Proceedings - Winter Simulation Conference*, pages 1151–1159. IEEE.

[Berretti et al., 2000] Berretti, S., Bimbo, A. D., and Pala, P. (2000). Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia*, 2(4):225–239.

[Bharosa et al., 2010] Bharosa, N., Lee, J., and Janssen, M. (2010). Challenges and obstacles in sharing and coordinating information during multi-agency disaster response: Propositions from field exercises. *Information Systems Frontiers*, 12(1):49–65.

[Bizid et al., 2014] Bizid, I., Faiz, S., Boursier, P., and Yusuf, J. C. M. (2014). Integration of heterogeneous spatial databases for disaster management. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence* *and Lecture Notes in Bioinformatics)*, volume 8697, chapter Integratio, pages 77–86. Springer International Publishing, Cham.

[Blatt et al., 2016] Blatt, F., Becker, M., and Szczerbicka, H. (2016). Analysing the cost-efficiency of the multi-agent flood algorithm in search and rescue scenarios. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9872 LNAI, pages 147–154. Springer.

[Boddy et al., 2005] Boddy, D., Boonstra, A., and Kennedy, G. (2005). *Managing information systems: an organisational perspective*. FT Prentice Hall, Harlow.

[Borshchev, 2013] Borshchev, A. (2013). *The big book of simulation modeling: multi-method modeling with AnyLogic 6*. AnyLogic North America.

[Borst, 1997] Borst, W. (1997). *Construction of Engineering Ontologies*. PhD thesis, Institute for Telematica and Information Technology, The Netherlands.

[Boufedji et al., 2018] Boufedji, D., Guessoum, Z., Brandão, A., Ziadi, T., and Mokhtari, A. (2018). Towards a MAS Product Line Engineering Approach. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10738 LNAI, pages 161–179. Springer.

[Bratman, 1987] Bratman, M. (1987). *Intention, plans, and practical reason*, volume 10. Harvard University Press Cambridge, MA.

[Brooks, 1991] Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1-3):139–159.

[Brown and Robinson, 2005] Brown, D. E. and Robinson, C. D. (2005). Development of metrics to evaluate effectiveness of emergency response operations. Technical report, VIRGINIA UNIV CHARLOTTESVILLE.

[Buford et al., 2006] Buford, J., Jakobson, G., Lewis, L., Parameswaran, N., and Ray, P. (2006). D-AESOP: A Situation-AWARE BDI Agent System for Disaster Situation Management. In *Proceedings In AAMAS Workshop on Agent Technology for Disaster Management, Japan*, pages 1–23.

[Bundesministerium für Inneres, 2015] Bundesministerium für Inneres (2015). System des Krisenmanagements in Deutschland www.bevoelkerungsschutz-portal.de. Technical report, Bundesministerium des Innern.

[Burel et al., 2017] Burel, G., Piccolo, L. S., Meesters, K., and Alani, H. (2017). DoRES-A three-tier ontology for modelling crises in the digital age. In *Proceedings of the International ISCRAM Conference*, volume 2017-May, pages 834–845.

[Burggraf, 2006] Burggraf, D. S. (2006). Geography markup language. *Data Science Journal*, 5:178–204.

[Buscaldi and Rosso, 2008] Buscaldi, D. and Rosso, P. (2008). Geo-WordNet: Automatic georeferencing of WordNet. In *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*, pages 1255–1258.

[Casado et al., 2015] Casado, R., Rubiera, E., Sacristan, M., Schütte, F., and Peters, R. (2015). Data interoperability software solution for emergency reaction in the Europe Union. *Natural Hazards and Earth System Sciences*, 15(7):1563–1576.

[Castaner, 2020] Castaner, C. (2020). Planification et exercices de Sécurité civile - L'organisation de la réponse de sécurité civile (ORSEC). Technical report, Ministère de l'intérieur.

[Chaffey and White, 2010] Chaffey, D. and White, G. (2010). *Business Information Management: Improving Performance Using Information Systems*. FT Prentice Hall, Harlow.

[Choi et al., 2006] Choi, N., Song, I.-Y., and Han, H. (2006). A survey on ontology mapping. *ACM Sigmod Record*, 35(3):34–41.

[Christensen and Sasaki, 2008] Christensen, K. and Sasaki, Y. (2008). Agent-Based emergency evacuation simulation with individuals with disabilities in the population. *Jasss*, 11(3):9.

[Christley et al., 2004] Christley, S., Xiang, X., and Madey, G. (2004). An ontology for agent-based modeling and simulation. In *Proceedings of the agent 2004 conference*.

[Chu et al., 2009] Chu, T. Q., Drogoul, A., Boucher, A., and Zucker, J. D. (2009). Interactive learning of independent experts' criteria for rescue simulations. *Journal of Universal Computer Science*, 15(13):2719–2743.

[Cointe, 2005] Cointe, P. (2005). Towards generative programming. In Banâtre, J.-P., Fradet, P., Giavitto, J.-L., and Michel, O., editors, *Unconventional Programming Paradigms*, pages 315–325, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Coppola, 2011] Coppola, D. P. (2011). *Introduction to International Disaster Management*. Elsevier.

[Corman et al., 2018] Corman, J., Reutter, J. L., and Savković, O. (2018). Semantics and validation of recursive SHACL. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11136 LNCS, pages 318–336. Springer.

[Cregan et al., 2005] Cregan, A., Mochol, M., Vrandecic, D., and Bechhofer, S. (2005). Pushing the limits of OWL, rules and protege a simple example. *CEUR Workshop Proceedings*, 188.

[Cruz and Xiao, 2005] Cruz, I. F. and Xiao, H. (2005). The role of ontologies in data integration. *Engineering Intelligent Systems*, 13(4):245–252.

[Das et al., 2012] Das, S., Sundara, S., and Cyganiak, R. (2012). R2RML: RDB to RDF Mapping Language. *W3C Recommendation*, pages 1–34.

[Davenport et al., 1998] Davenport, T. H., Prusak, L., and Others (1998). Working knowledge: how organizations manage what they know. *Choice Reviews Online*, 35(09):35–5167–35–5167.

[DDSC, 2005] DDSC (2005). Plan Communal de Sauvegarde : Guide pratique d'élaboration. Technical report, Direction de la Défense et de la Sécurité Civiles.

[de Farias et al., 2016] de Farias, T. M., Roxin, A., and Nicolle, C. (2016). SWRL rule-selection methodology for ontology interoperability. *Data and Knowledge Engineering*, 105:53–72.

[De La Asunción et al., 2005a] De La Asunción, M., Castillo, L., Fdez-Olivares, J., García-Pérez, O., González, A., and Palao, F. (2005a). SIADEX: An interactive knowledge-based planner for decision support in forest fire fighting. *AI Communications*, 18(4):257–268.

[De La Asunción et al., 2005b] De La Asunción, M., Castillo, L., Fdez-Olivares, J., Garcıa-Pérez, O., González, A., and Palao, F. (2005b). Knowledge and plan execution management in planning fire fighting operations. *Planning, Scheduling And Constraint Satisfaction: From Theory To Practice*, 117:149.

[Debruyne et al., 2017] Debruyne, C., McGlinn, K., McNerney, L., and O'Sullivan, D. (2017). A lightweight approach to explore, enrich and use data with a geospatial dimension with semantic web technologies. In *GeoRich 2017 - 4th International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data, in Conjunction with SIGMOD 2017*, pages 1–6. ACM.

[Dentler et al., 2011] Dentler, K., Cornet, R., Ten Teije, A., and De Keizer, N. (2011). Comparison of reasoners for large ontologies in the owl 2 el profile. *Semantic Web*, 2(2):71–87.

[Départementale-métropolitaine des jeunes sapeurs pompiers, 2017] Départementale-métropolitaine des jeunes sapeurs pompiers, A. (2017). SMV-NOVI.

[Description, 2008] Description, L. D. (2008). Logiques de Description.

[Di Bitonto et al., 2012] Di Bitonto, P., Laterza, M., Roselli, T., and Rossano, V. (2012). Evaluation of multi-agent systems: Proposal and validation of a metric plan. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7270 LNCS, pages 198–221. Springer.

[Do and Rahm, 2002] Do, H.-H. and Rahm, E. (2002). COMA — A system for flexible combination of schema matching approaches. In *VLDB '02: Proceedings of the 28th International Conference on Very Large Databases*, pages 610–621. VLDB Endowment.

[D'Orazio et al., 2014] D'Orazio, M., Quagliarini, E., Bernardini, G., and Spalazzi, L. (2014). EPES - Earthquake pedestrians' evacuation simulator: A tool for predicting earthquake pedestrians' evacuation in urban outdoor scenarios. *International Journal of Disaster Risk Reduction*, 10(PA):153–177.

[Dunn, 1973] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

[Durak and Oren, 2016] Durak, U. and Oren, T. (2016). Towards an ontology for simulation systems engineering. *Simulation Series*, 48(2):92–99.

[Durif, 2014] Durif, T. (2014). *Environnement informé sémantiquement enrichi pour la simulation multi-agents: application à la simulation en environnement virtuel 3D*. PhD thesis, Dijon.

[Environmental Systems Research Institute (ESRI), 1998] Environmental Systems Research Institute (ESRI) (1998). ESRI Shapefile Technical Description. *Computational Statistics*, 16(July):370–371.

[Fan and Zlatanova, 2010] Fan, Z. and Zlatanova, S. (2010). Exploring ontology potential in emergency management. In *GI4DM 2010 Conference - Geomatics for Crisis Management*, pages 132–137. Politechno di Torino.

[Farias et al., 2015] Farias, T. M., Roxin, A., and Nicolle, C. (2015). FOWLA, A federated architecture for ontologies. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9202, pages 97–111. Springer.

[Fazal, 2017] Fazal, S. (2017). *GIS Basics*. New Age International.

[Federal Emergency Management Agency, 2010] Federal Emergency Management Agency (2010). Developing and maintaining emergency operations plans. Comprehensive Preparedness Guide 101.

[Ferber, 1997] Ferber, J. (1997). *Les systèmes multi-agents: vers une intelligence collective*. InterEditions.

[Fortin, 2013] Fortin, P. (2013). Organisation de la Réponse de SEcurité civile. Technical report, Direction de la Sécurité Civile.

[Frankish, 2005] Frankish, K. (2005). Non-monotonic inference. *The Encyclopedia of Language and Linguistics*.

[Gangemi et al., 2002] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). Sweetening ontologies with DOLCE. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 2473, pages 166–181. Springer.

[Gaur et al., 2019] Gaur, M., Shekarpour, S., Gyrard, A., and Sheth, A. (2019). Empathi: An Ontology for Emergency Managing and Planning about Hazard Crisis. In *Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019*, pages 396–403. IEEE.

[Gearon et al., 2013] Gearon, P., Passant, A., and Polleres, A. (2013). SPARQL 1.1 Update.

[Genesereth, Michael R. and Nilsson, N. J., 1987] Genesereth, Michael R. and Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, Los Altos, California, USA.

[GeoPii, Integrasys, 2014] GeoPii, Integrasys, T. (2014). D2.2 Definition of Modelling Hierarchy. Technical report, COBACORE - European Commission.

[Ghawi and Cullot, 2007] Ghawi, R. and Cullot, N. (2007). Database-to-Ontology Mapping Generation for Semantic Interoperability. In *VDBL'07 conference, VLDB Endowment ACM*, volume 91, pages 1—-8.

[Grange et al., 2014] Grange, J. J. L., Lehmann, J., Athanasiou, S., Rojas, A. G., Giannopoulos, G., Hladky, D., Isele, R., Ngonga Ngomo, A.-C., Sherif, M. A., Stadler, C., and Wauer, M. (2014). The GeoKnow Generator: Managing Geospatial Data in the Linked Data Web. *Proceedings of the Linking Geospatial Data Workshop*.

[Grignard et al., 2013] Grignard, A., Taillandier, P., Gaudou, B., Vo, D. A., Huynh, N. Q., and Drogoul, A. (2013). Gama 1.6: Advancing the art of complex agent-based modeling and simulation. In *International conference on principles and practice of multi-agent systems*, pages 117–131. Springer.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

[Gruber, 1995] Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human - Computer Studies*, 43(5-6):907–928.

[Guarino, 1995] Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human - Computer Studies*, 43(5-6):625–640.

[Guarino et al., 2009] Guarino, N., Oberle, D., and Staab, S. (2009). Handbook on Ontologies. In *Handbook on Ontologies*, chapter What is an, pages 1–17. Springer Science & Business Media.

[Guha et al., 2000] Guha, S., Rastogi, R., and Shim, K. (2000). Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366.

[Guha et al., 2001] Guha, S., Rastogi, R., and Shim, K. (2001). Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26(1):35 – 58.

[Guha-Sapir and Below, 2002] Guha-Sapir, D. and Below, R. (2002). The quality and accuracy of disaster data: A comparative analyse of 3 global data sets. *Disaster Management facility, World Bank, Working paper ID*.

[Gulati et al., 2012] Gulati, R., Wohlgezogen, F., and Zhelyazkov, P. (2012). The Two Facets of Collaboration: Cooperation and Coordination in Strategic Alliances. *Academy of Management Annals*, 6(1):531–583.

[Guo and Huang, 2008] Guo, R. Y. and Huang, H. J. (2008). A modified floor field cellular automata model for pedestrian evacuation simulation. *Journal of Physics A: Mathematical and Theoretical*, 41(38):385104.

[Hacherouf et al., 2015] Hacherouf, M., Bahloul, S. N., and Cruz, C. (2015). Transforming XML documents to OWL ontologies: A survey. *Journal of Information Science*, 41(2):242–259.

[Han et al., 2010] Han, L., Potter, S., Beckett, G., Pringle, G., Welch, S., Koo, S. H., Wickler, G., Usmani, A., Torero, J. L., and Tate, A. (2010). FireGrid: An e-infrastructure for next-generation emergency response support. *Journal of Parallel and Distributed Computing*, 70(11):1128–1141.

[Hawe et al., 2012] Hawe, G. I., Coates, G., Wilson, D. T., and Crouch, R. S. (2012). Agent-based simulation for large-scale emergency response: A survey of usage and implementation. *ACM Computing Surveys*, 45(1):8.

[Hawe et al., 2015] Hawe, G. I., Coates, G., Wilson, D. T., and Crouch, R. S. (2015). Agent-based simulation of emergency response to plan the allocation of resources for a hypothetical two-site major incident. *Engineering Applications of Artificial Intelligence*, 46:336–345.

[Helsinger and Wright, 2005] Helsinger, A. and Wright, T. (2005). Cougaar: A robust configurable multi agent platform. In *2005 IEEE Aerospace Conference*, pages 1–10. IEEE.

[Hillner and Ngomo, 2011] Hillner, S. and Ngomo, A. C. N. (2011). Parallelizing LIMES for large-scale link discovery. In *ACM International Conference Proceeding Series*, pages 9–16. ACM.

[Hitzler et al., 2009a] Hitzler, P., Krötzsch, M., and Rudolph, S. (2009a). *Foundations of Semantic Web Technologies*. Chapman and Hall/CRC.

[Hitzler et al., 2009b] Hitzler, P., Parsia, B., Patel-schneider, P. F., and Rudolph, S. (2009b). OWL 2 Web Ontology Language Primer. *W3C Recommendation*, 27(December):1–123.

[Homburg et al., 2018] Homburg, T., Prudhomme, C., and Boochs, F. (2018). Semantische Geoinformationssysteme: Integration und Management von heterogener Geodaten.

[Homburg et al., 2017] Homburg, T., Prudhomme, C., Boochs, F., Roxin, A., and Cruz, C. (2017). Integration, quality assurance and usage of geospatial data with semantic tools. *gis.Science - Die Zeitschrift fur Geoinformatik*, 3:91–96.

[Homburg et al., 2016] Homburg, T., Prudhomme, C., Würriehausen, F., Karmacharya, A., Boochs, F., Roxin, A., and Cruz, C. (2016). Interpreting heterogeneous geospatial data using semantic web technologies. In *Computational Science and Its Applications – ICCSA 2016*, pages 240–255. Springer International Publishing.

[Hooshangi and Alesheikh, 2017] Hooshangi, N. and Alesheikh, A. A. (2017). Agent-based task allocation under uncertainties in disaster environments: An approach to interval uncertainty. *International Journal of Disaster Risk Reduction*, 24:160–171.

[Horrocks et al., 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member submission*, 21(79):1–31.

[Huttenlocher et al., 1992] Huttenlocher, D. P., Rucklidge, W. J., and Klanderman, G. A. (1992). Comparing images using the Hausdorff distance under translation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992-June(9):654–656.

[IEEE, 2011] IEEE (2011). IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP).

[Information, 2014] Information, E. P.-e. (2014). D3 . 1 - Description of method and data sets for the pan-European inventory of disasters. Technical report, EPISECC project, grant agreement [607078].

[Jashapara, 2004] Jashapara, A. (2004). *Knowledge management: An integrated approach*. Pearson Education.

[Jessup and Valacich, 2003] Jessup, L. and Valacich, J. (2003). *Information systems today*. Prentice Hall Upper Saddle River, NJ.

[Jimènez-Ruiz et al., 2015] Jimènez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., Horrocks, I., Pinkel, C., Skjæveland, M. G., Thorstensen, E., and Mora, J. (2015). BOOTOX: Practical mapping of RDBs to OWL 2. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9367, pages 113–132. Springer.

[Jones, 2013] Jones, E. (2013). Emergency data exchange language (edxl) distribution element version 2.0 - committee specification 02.

[Jonkman, 2005] Jonkman, S. N. (2005). Global perspectives on loss of human life caused by floods. *Natural Hazards*, 34(2):151–175.

[Kalemi and Martiri, 2011] Kalemi, E. and Martiri, E. (2011). FOAF-Academic ontology: A Vocabulary for the academic community. In *Proceedings - 3rd IEEE International Conference on Intelligent Networking and Collaborative Systems, INCoS 2011*, pages 440–445. IEEE.

[Karmacharya et al., 2015] Karmacharya, A., Boochs, F., and Tietz, B. (2015). Knowledge guided object detection and identification in 3D point clouds. In *Videometrics, Range Imaging, and Applications XIII*, volume 9528, page 952804. International Society for Optics and Photonics.

[Karypis et al., 1999] Karypis, G., Eui-Hong Han, and Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75.

[Keßler and Hendrix, 2015] Keßler, C. and Hendrix, C. (2015). The Humanitarian eXchange Language: Coordinating disaster response with semantic web technologies. *Semantic Web*, 6(1):5–21.

[Klügl, 2009] Klügl, F. (2009). Sesam: Visual programming and participatory simulation for agent-based models.

[Knoblock et al., 2012] Knoblock, C. A., Szekely, P., Ambite, J. L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., and Mallick, P. (2012). Semi-automatically mapping structured sources into the semantic web. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7295 LNCS, pages 375–390. Springer.

[Knublauch et al., 2011] Knublauch, H., Hendler, J. A., and Idehen, K. (2011). SPIN-overview and motivation. *W3C Member Submission*, 22:W3C.

[Knublauch and Kontokostas, 2017] Knublauch, H. and Kontokostas, D. (2017). Shapes Constraint Language (SHACL) W3C Working Draft. *W3C Recommendation*, 20(March):1–107.

[Kontopoulos et al., 2018] Kontopoulos, E., Moßgraber, J., Van Der Schaaf, H., Lombardo, F., Ferri, M., Vrochidis, S., Mitzias, P., Hertweck, P., Hilbring, D., Norbiato, D., Karakostas, A., and Kompatsiaris, I. (2018). Ontology-based representation of crisis management procedures for climate events. In *Proceedings of the International ISCRAM Conference*, volume 2018-May, pages 1051–1062.

[Kravari and Bassiliades, 2015] Kravari, K. and Bassiliades, N. (2015). A survey of agent platforms. *Jasss*, 18(1):11.

[Kruchten et al., 2007] Kruchten, P., Woo, C., Monu, K., and Sotoodeh, M. (2007). A human-centered conceptual model of disasters affecting critical infrastructures. In *Intelligent Human Computer Systems for Crisis Response and Management, ISCRAM 2007 Academic Proceedings Papers*, pages 327–344.

[Labba et al., 2017] Labba, C., Salah, N. B., and Bellamine Ben Saoud, N. (2017). An agent-based meta-model for response organization structures. In *Lecture Notes in Business Information Processing*, volume 301, pages 154–167. Springer.

[Lacy and Gerber, 2004] Lacy, L. and Gerber, W. (2004). Potential modeling and simulation applications of the Web Ontology Language-owl. In *Proceedings - Winter Simulation Conference*, volume 1, pages 265–270. IEEE.

[Lange and Zeugrnann, 1993] Lange, S. and Zeugrnann, T. (1993). Monotonic versus non-monotonic language learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 659 LNAI, pages 254–269. Springer.

[Larsson, 2008] Larsson, A. (2008). A Framework for Evaluating Emergency Preparedness Plans and Response Strategies. *IIASA Interim Report*.

[Laudon and Laudon, 2006] Laudon, K. and Laudon, J. (2006). *Management Information Systems: Managing the Digital Firm 9th edn.* Pearson Prentice Hall, Upper Saddle River, NJ.

[Lauras et al., 2015] Lauras, M., Truptil, S., and Bénaben, F. (2015). Towards a better management of complex emergencies through crisis management metamodelling. *Disasters*, 39(4):687–714.

[Levenshtein, 1966] Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707 – 710.

[Liu et al., 2013] Liu, S., Brewster, C., and Shaw, D. (2013). Ontologies for crisis management: A review of state of the art in ontology design and usability. In *ISCRAM 2013 Conference Proceedings - 10th International Conference on Information Systems for Crisis Response and Management*, pages 349–359.

[MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297. Oakland, CA, USA.

[Magdelaine, 2009] Magdelaine, C. (2009). Statistiques sur les catastrophes naturelles. http://www.notre-planete.info/geographie/risques_naturels/catastrophes_naturelles.php.

[Malone and Crowston, 1990] Malone, T. W. and Crowston, K. (1990). What is coordination theory and how can it help design cooperative work systems? In *Proceedings of the 1990 ACM Conference on Computer-Supported Cooperative Work, CSCW 1990*, pages 357–370. ACM.

[Manning et al., 2015] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2015). The Stanford CoreNLP Natural Language Processing Toolkit. In *52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

[Marecki et al., 2005] Marecki, J., Schurr, N., and Tambe, M. (2005). *Agent-Based Simulations for Disaster Rescue Using the DEFACTO Coordination System*, chapter 14, pages 281–297. John Wiley & Sons, Ltd.

[Mas et al., 2015] Mas, E., Koshimura, S., Imamura, F., Suppasri, A., Muhari, A., and Adriano, B. (2015). Recent Advances in Agent-Based Tsunami Evacuation Simulations: Case Studies in Indonesia, Thailand, Japan and Peru. *Pure and Applied Geophysics*, 172(12):3409–3424.

[McBride, 2004] McBride, B. (2004). The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS. In *Handbook on Ontologies*, pages 51–65. Springer.

[McConnell and Drennan, 2006] McConnell, A. and Drennan, L. (2006). Mission impossible? Planning and preparing for crisis. *Journal of Contingencies and Crisis Management*, 14(2):59–70.

[McGinnis et al., 2011] McGinnis, L., Huang, E., Kwon, K. S., and Ustun, V. (2011). Ontologies and simulation: A practical approach. *Journal of Simulation*, 5(3):190–201.

[McGuinness et al., 2004] McGuinness, D. L., Van Harmelen, F., et al. (2004). Owl web ontology language overview. *W3C recommendation*, 10(10):2004.

[Melnik et al., 2002] Melnik, S., Garcia-Molina, H., and Rahm, E. (2002). Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings - International Conference on Data Engineering*, pages 117–128. IEEE.

[Mendes Calo et al., 2012] Mendes Calo, K., Cenci, K. M., Fillottrani, P. R., and Estévez, E. C. (2012). Information sharing. *Journal of Computer Science & Technology*, 12.

[Miller et al., 2004] Miller, J. A., Baramidze, G. T., Sheth, A. P., and Fishwick, P. A. (2004). Investigating ontologies for simulation modeling. In *Proceedings of the IEEE Annual Simulation Symposium*, pages 55–63. IEEE Computer Society.

[Mishra et al., 2019] Mishra, D., Kumar, S., and Hassini, E. (2019). Current trends in disaster management simulation modelling research. *Annals of Operations Research*, 283(1-2):1387–1411.

[Nagarajan et al., 2012] Nagarajan, M., Shaw, D., and Albores, P. (2012). Disseminating a warning message to evacuate: A simulation study of the behaviour of neighbours. *European Journal of Operational Research*, 220(3):810–819.

[Nance, 1994] Nance, R. E. (1994). The Conical Methodology and the evolution of simulation model development. *Annals of Operations Research*, 53(1):1–45.

[Narzisi et al., 2007] Narzisi, G., Mincer, J. S., Smith, S., and Mishra, B. (2007). Resilience in the face of disaster: Accounting for varying disaster magnitudes, resource topologies, and (sub)population distributions in the PLAN C emergency planning tool. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4659 LNAI, pages 433–446. Springer.

[Nentwig et al., 2017] Nentwig, M., Hartung, M., Ngonga Ngomo, A. C., and Rahm, E. (2017). A survey of current Link Discovery frameworks. *Semantic Web*, 8(3):419–436.

[Neteler and Mitasova, 2008] Neteler, M. and Mitasova, H. (2008). *Open source GIS: A GRASS GIS approach*, volume 689. Springer Science & Business Media.

[Ng and Han, 2002] Ng, R. T. and Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016.

[Ngomo and Auer, 2011] Ngomo, A. C. N. and Auer, S. (2011). LIMES - A time-efficient approach for large-scale link discovery on the web of data. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 2312–2317.

[Niu et al., 2011] Niu, X., Rong, S., Zhang, Y., and Wang, H. (2011). Zhishi.links results for OAEI 2011. *CEUR Workshop Proceedings*, 814:220–227.

[Niwattanakul et al., 2013] Niwattanakul, S., Singthongchai, J., Naenudorn, E., and Wanapu, S. (2013). Using of jaccard coefficient for keywords similarity. *Lecture Notes in Engineering and Computer Science*, 2202(6):380–384.

[Noy and McGuinness, 2001] Noy, N. F. and McGuinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Technical report, Stanford University, Stanford, CA, 94305.

[Otero-Cerdeira et al., 2015] Otero-Cerdeira, L., Rodríguez-Martínez, F. J., and Gómez-Rodríguez, A. (2015). Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949 – 971.

[Othman and Beydoun, 2013] Othman, S. H. and Beydoun, G. (2013). Model-driven disaster management. *Information & Management*, 50(5):218–228.

[Othman and Beydoun, 2016] Othman, S. H. and Beydoun, G. (2016). A metamodel-based knowledge sharing system for disaster management. *Expert Systems with Applications*, 63:49–65.

[Othman et al., 2014] Othman, S. H., Beydoun, G., and Sugumaran, V. (2014). Development and validation of a Disaster Management Metamodel (DMM). *Information Processing and Management*, 50(2):235–271.

[Pan and Space, 2016] Pan, E. and Space, E. I. (2016). D4 . 4 . – Ontology model for the EPISECC use case. Technical report, EPISECC project, grant agreement [607078].

[Pan, 2009] Pan, J. Z. (2009). Resource Description Framework. In *Handbook on Ontologies*, pages 71–90. Springer.

[Patel-Schneider, 2017] Patel-Schneider, P. F. (2017). ASHACL: Alternative Shapes Constraint Language. *arXiv preprint*, pages 1–18.

[Patroumpas et al., 2014] Patroumpas, K., Alexakis, M., Giannopoulos, G., and Athanasiou, S. (2014). TripleGeo: An ETL tool for transforming geospatial data into RDF triples. In *CEUR Workshop Proceedings*, volume 1133, pages 275–278.

[Paulheim et al., 2009] Paulheim, H., Döweling, S., Tso-Sutter, K., Probst, F., and Ziegert, T. (2009). Improving usability of integrated emergency response systems: the SoKNOS approach. *INFORMATIK 2009 - Im Focus das Leben, Beitrage der 39. Jahrestagung der Gesellschaft fur Informatik e.V. (GI)*, 154:1435–1449.

[Perry and Herring, 2012] Perry, M. and Herring, J. (2012). OGC GeoSPARQL-A geographic query language for RDF data.

[Pinkel et al., 2017] Pinkel, C., Binnig, C., Jimenez-Ruiz, E., Kharlamov, E., Nikolov, A., Schwarte, A., Heupel, C., and Kraska, T. (2017). INCMap: A Journey towards ontology-based data integration. *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*, 265:145–164.

[Ponciano et al., 2019a] Ponciano, J.-J., Karmacharya, A., Wefers, S., Atorf, P., and Boochs, F. (2019a). *Connected Semantic Concepts as a Base for Optimal Recording and Computer-Based Modelling of Cultural Heritage Objects*, volume 18. Springer International Publishing.

[Ponciano et al., 2019b] Ponciano, J.-J., Trémeau, A., and Boochs, F. (2019b). Automatic detection of objects in 3d point clouds based on exclusively semantic guided processes. *ISPRS International Journal of Geo-Information*, 8(10).

[Poveda et al., 2015] Poveda, G., Serrano, E., and Garijo, M. (2015). Designing emergency management services by ontology driven social simulation. *IT CoNvergence PRActice (INPRA)*, 3(1):17–32.

[Praiwattana and El Rhalibi, 2016] Praiwattana, P. and El Rhalibi, A. (2016). Survey: Development and analysis of a games-based crisis scenario generation system. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9654, pages 85–100. Springer.

[Prudhomme et al., 2019a] Prudhomme, C., Cruz, C., and Boochs, F. (2019a). Modélisation sémantique et logique pour une simulation multi-agent dans le contexte de gestion de catastrophe. In *International Francophone Conference SAGEO (Spatial Analysis and Geomatics)*, Clermont Ferrand, France.

[Prudhomme et al., 2019b] Prudhomme, C., Cruz, C., and Boochs, F. (2019b). Semantic and Logic Modeling of Disaster Simulation for Multi-agent Systems. *International Journal of Modeling and Optimization*, 9(4):198–204.

[Prudhomme et al., 2018] Prudhomme, C., Cruz, C., Roxin, A., and Boochs, F. (2018). A Framework to Improve the Disaster Response Through a Knowledge-Based Multi-Agent System. *International Journal of Information Systems for Crisis Response and Management*, 9(3):96–109.

[Prudhomme et al., 2020a] Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Cruz, C., and Roxin, A.-M. (2020a). Interpretation and automatic integration of geospatial data into the semantic web. *Computing*, 102(2):365–391.

[Prudhomme et al., 2017a] Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Roxin, A., and Cruz, C. (2017a). Automatic integration of spatial data into the semantic web. In *WebIST 2017* , Porto, Portugal.

[Prudhomme et al., 2017b] Prudhomme, C., Homburg, T., Ponciano, J. J., Boochs, F., Roxin, A., and Cruz, C. (2017b). Automatic integration of spatial data into the semantic web. In *WEBIST 2017 - Proceedings of the 13th International Conference on Web Information Systems and Technologies*, pages 107–115.

[Prudhomme et al., 2016] Prudhomme, C., Roxin, A., Cruz, C., and Boochs, F. (2016). Semantic Catalogue to Manage Data Sources in Disaster Management System. In *International Conference on Informatics in Economy, Ie 2016: Education, Research & Business Technologies*, pages 225–230.

[Prudhomme et al., 2017c] Prudhomme, C., Roxin, A., Cruz, C., and Boochs, F. (2017c). Towards the design of respond action in disaster management using knowledge modeling. In *Lecture Notes in Business Information Processing*, volume 301, pages 168–174. Springer.

[Prudhomme et al., 2020b] Prudhomme, C., Roxin, A., Cruz, C., and Boochs, F. (2020b). Modélisation sémantique et programmation générative pour une simulation multi-agent dans le contexte de gestion de catastrophe. *Revue Internationale de Géomatique*.

[Prud'hommeaux and Seaborne, 2008] Prud'hommeaux, E. and Seaborne, A. (2008). SPARQL Query Language for RDF. *W3C Recommendation*.

[Rahm, 2011] Rahm, E. (2011). Towards large-scale schema and ontology matching. In *Schema matching and mapping*, pages 3–27. Springer.

[Rahm and Bernstein, 2001] Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350.

[Rainer and Turban, 2008] Rainer, R. K. and Turban, E. (2008). *Introduction to Information Systems: Supporting and Transforming Business*. John Wiley & Sons.

[Rein et al., 2007] Rein, G., Abecassis Empis, C., and Carvel, R. (2007). *The Dalmarnock Fire Tests: Experiments and Modelling*. School of Engineering and Electronics, University of Edinburgh.

[Repici, 2006] Repici, D. J. (2006). The comma separated value (csv) file format. *Creativyst Software*.

[Rijgersberg et al., 2013] Rijgersberg, H., Van Assem, M., and Top, J. (2013). Ontology of units of measure and related concepts. *Semantic Web*, 4(1):3–13.

[Rodrigues Da Silva, 2015] Rodrigues Da Silva, A. (2015). Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems and Structures*, 43:139–155.

[Rodríguez-Muro and Rezk, 2015] Rodríguez-Muro, M. and Rezk, M. (2015). Efficient SPARQL-to-SQL with R2RML mappings. *Journal of Web Semantics*, 33:141–169.

[Rowley, 2007] Rowley, J. (2007). The wisdom hierarchy: Representations of the DIKW hierarchy. *Journal of Information Science*, 33(2):163–180.

[Saoud et al., 2006] Saoud, N. B.-B., Mena, T. B., Dugdale, J., Pavard, B., and Ahmed, M. B. (2006). Assessing large scale emergency rescue plans: an agent based approach. *The International Journal of Intelligent Control and Systems*, 11(4):260–271.

[Saxena et al., 2017] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., and Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267:664 – 681.

[Scharffe et al., 2012] Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., Hamdi, F., Bihanic, L., Képéklian, G., Cotton, F., Euzenat, J., Fan, Z., Vandenbussche, P. Y., and Vatant, B. (2012). Enabling linked data publication with the datalift platform. In *AAAI Workshop - Technical Report*, volume WS-12-13, pages 25–30.

[Schmidt-Schauß and Smolka, 1991] Schmidt-Schauß, M. and Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26.

[Schulte-Zurhausen, 2005] Schulte-Zurhausen, M. (2005). *Organisation*. Vahlen Franz GmbH; 6. Auflage.

[Seifert, 2008] Seifert, M. (2008). INfrastructure for SPatial InfoRmation in Europe INSPIRE.

[Shafiq et al., 2012] Shafiq, B., Chun, S. A., Atluri, V., Vaidya, J., and Nabi, G. (2012). Resource sharing using UICDS™ framework for incident management. *Transforming Government: People, Process and Policy*, 6(1):41–61.

[Shaw et al., 2009] Shaw, R., Troncy, R., and Hardman, L. (2009). Lode: Linking open descriptions of events. In Gómez-Pérez, A., Yu, Y., and Ding, Y., editors, *The Semantic Web*, pages 153–167, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Shearer et al., 2008] Shearer, R., Motik, B., and Horrocks, I. (2008). HermiT: A Highly-Efficient OWL Reasoner. In *OWLED*, volume 432, page 91.

[Siddhartha et al., 2009] Siddhartha, H., Sarika, R., and Karlapalem, K. (2009). Score Vector : A New Evaluation Scheme for RoboCup Rescue Simuation Competition 2009. *Evaluation*.

[Siebra and Tate, 2003] Siebra, C. and Tate, A. (2003). I-Rescue: A Coalition Based System to Support Disaster Relief Operations. *Proceedings of the Third International Conference on Artificial Intelligence and Applications*.

[Silver et al., 2007] Silver, G. A., Al-Haj Hassan, O., and Miller, J. A. (2007). From domain ontologies to modeling ontologies to executable simulation models. In *Proceedings - Winter Simulation Conference*, pages 1108–1117. IEEE Press.

[Silver et al., 2009] Silver, G. A., Bellipady, K. R., Miller, J. A., Kochut, K. J., and York, W. (2009). Supporting interoperability using the discrete-event modeling ontology (DeMO). In *Proceedings - Winter Simulation Conference*, pages 1399–1410. IEEE.

[Sirin et al., 2007] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53.

[Šišlák et al., 2006] Šišlák, D., Rehák, M., Pěchouček, M., and Pavlíček, D. (2006). Deployment of a-globe multi-agent platform. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1447–1448.

[Smart et al., 2007a] Smart, P. R., Russell, A., and Shadbolt, N. R. (2007a). AKTiveSA: Supporting civil-military information integration in Military Operations Other Than War. In *2007 International Conference on Integration of Knowledge Intensive Multi-Agent Systems, KIMAS 2007*, pages 434–439. IEEE.

[Smart et al., 2007b] Smart, P. R., Russell, A., Shadbolt, N. R., Shraefel, M. C., and Carr, L. A. (2007b). AKTiveSA: A technical demonstrator system for enhanced situation awareness. *Computer Journal*, 50(6):703–716.

[Stadler et al., 2013] Stadler, C., Unbehauen, J., Lehmann, J., and Auer, S. (2013). Connecting Crowd-sourced Spatial Information to the Data Web with Sparqlify.pdf. Technical report, University of Leipzig.

[Studer et al., 1998] Studer, R., Benjamins, R., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2):161–198.

[Taillandier et al., 2019] Taillandier, P., Gaudou, B., Grignard, A., Huynh, Q. N., Marilleau, N., Caillou, P., Philippon, D., and Drogoul, A. (2019). Building, composing and experimenting complex spatial models with the GAMA platform. *GeoInformatica*, 23(2):299–322.

[Takahashi, 2003] Takahashi, T. (2003). RoboCupRescue simulation league. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, volume 2752, pages 477–481. Springer.

[Tate, 2003] Tate, A. (2003). <{INCA}>: an ontology for mixed-initiative synthesis tasks. In *Proceedings of the workshop on mixed-initiative intelligent systems ({MIIS}) at the international joint conference on artificial intelligence ({IJCAI-03}), Acapulco, Mexico*, pages 125–130.

[The Sphere Project, 2011] The Sphere Project (2011). *Humanitarian Charter and Minimum Standards in Humanitarian Response*. Practical Action.

[Touraine, 2014] Touraine, M. (2014). Guide méthodologique: Aide à l'organisation de l'offre de soins en situations sanitaires exceptionnelles. Technical report, Ministère des affaires sociales, de la santé et des droits des femmes.

[Tsarkov and Horrocks, 2006] Tsarkov, D. and Horrocks, I. (2006). FaCT++ description logic reasoner: System description. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4130 LNAI, pages 292–297. Springer.

[Turban et al., 2005] Turban, E., Rainer, R. K., and Potter, R. E. (2005). *Introduction to Information Technology, 3rd edn*. New York, Wiley.

[Uschold and Gruninger, 1996] Uschold, M. and Gruninger, M. (1996). Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11(2):93–136.

[Veltkamp, 2001] Veltkamp, R. C. (2001). Shape matching: Similarity measures and algorithms. In *Proceedings - International Conference on Shape Modeling and Applications, SMI 2001*, pages 188–197. IEEE.

[Volz et al., 2009] Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. (2009). Silk - A Link Discovery Framework for the Web of Data. *CEUR Workshop Proceedings*, 538.

[W3C OWL Working Group, 2012] W3C OWL Working Group (2012). OWL 2 Web Ontology Language Document Overview. *OWL 2 Web Ontology Language*, pages 1–7.

[Walter et al., 2016] Walter, H., Gering, F., Arnold, K., Gerich, B., Heinrich, G., and Welte, U. (2016). Rodos-based simulation of potential accident scenarios for emergency response management in the vicinity of nuclear power plants. *BfS-SCHR-60/16. urn: nbn: de*, pages 0221–2016091214084.

[Wickler et al., 2006] Wickler, G., Tate, A., and Potter, S. (2006). Using the< INCA> Constraint Model as a Shared Representation of Intentions for Emergency Response. In *Proceedings of the First International Workshop on Agent Technology for Disaster Management (ATDM), at the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Future University, Hakodate, Japan*, pages 2–9. Fifth International Joint Conference on Autonomous Agents and Multiagent Systems.

[Wooldridge, 2002] Wooldridge, M. (2002). *An introduction to multiagent systems*. John Wiley & Sons.

[Zhang et al., 2005] Zhang, J., Dawes, S. S., and Sarkis, J. (2005). Exploring stakeholders' expectations of the benefits and barriers of e-government knowledge sharing. *Journal of Enterprise Information Management*, 18(5):548–567.

[Zhang et al., 1996] Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114.

[Zhou et al., 2012] Zhou, T., Wang, D., Zong, D., and He, X. (2012). The Study of the responsibility space regionalization of emergency shelters in the urban center and the evaluation of evacuation road based on microscope computer simulation - A case of Chongqing three gorges square area. *Disaster Advances*, 5(4):230–236.

List of Figures

3.1 Cycle of disaster management 49

3.2 Cycle of preparedness according to [Federal Emergency Management Agency, 2010] 52

4.1 Ontology for multi-agent simulation proposed by [Christley et al., 2004] 114

4.2 Hierarchy of agents 115

4.3 Main concepts from GAML used by the GAMA platform interlinked to concepts of multi-agent simulation [Taillandier et al., 2019] 116

4.4 The DM Metamodel proposed by [Othman and Beydoun, 2013] . . . 119

4.5 Overview of Emergel ontology through its highest concepts 120

4.6 Tasks represented in Emergel ontology 121

4.7 GeoSPARQL vocabulary 122

4.8 Conceptualization overview of SemDM ontology 123

4.9 Method overview 125

4.10 Processing workflow of the method 126

4.11 Overview of simulation explicit conceptualization process from disaster management model 128

4.12 Overview of program explicit conceptualization process from the conceptual multi-agent simulation model 129

5.1 Deployment diagram of the proposed method 135

5.2 Overview of server's processes intervening in the method achievement136

5.3 Component 4 of the architecture: the simulation platform 137

5.4 Usage example of the new implemented skill: rescue behavior . . . 140

5.5 Class diagram of the root structure of the simulation 140

5.6 Component 3 of the architecture: the knowledge base 141
5.7 Overview of knowledge base 142
5.8 Modeling related to the concept *semDM:Service* representing disaster action service 143
5.9 Disaster event modeling 144
5.10 Procedure modeling 144
5.11 Plan's effectiveness modeling 145
5.12 Modeling related to the concept *semDM:Resource* representing disaster management resources 145
5.13 Geospatial modeling 146
5.14 Multi-agent simulation modeling 147
5.15 Representation of the conceptual simulation model 148
5.16 Agent modeling 149
5.17 Agent BDI modeling 149
5.18 Action modeling 150
5.19 Representation of the programmed simulation model 151
5.20 Species modeling 152
5.21 Skill modeling 153
5.22 Component 1 of the architecture: the Client 154
5.23 Component 2 of the architecture: Processing server 156
5.24 Overview of the proposed sequence 157

6.1 Disaster management modeling in processing timeline. 162
6.2 Screenshot of the user interface for the knowledge modeling. 164
6.3 Screenshot of the user interface for the integration of knowledge extracted from data. 165
6.4 User interface to visualize knowledge base's content using WebVOWL 1.1.7. 166
6.5 Methodology overview applied to an example [Prudhomme et al., 2017b] 169
6.6 Multi-agent simulation modeling in processing timeline 173
6.7 Overview of simulation modeling process 174

6.8 Multi-agent simulation design and execution in processing timeline 187

6.9 Main steps of the generative programming process, using retrieving actions through SPARQL queries highlighted in grey 188

6.10 Program structure resulting from the generative programming process, where variables depending on the simulation model are represented in blue. 189

6.11 Simulation execution in deployement diagram 193

6.12 Clustering-based analysis in processing timeline 194

6.13 Clustering-based analysis overview. 194

6.14 Plan Assessment workflow. 195

6.15 Segmentation workflow 197

6.16 Illustration of Segmentation 197

6.17 Illustration of Knowledge enrichment for global Plan 198

6.18 Illustration of Knowledge enrichment for specific Plan (with different shades of blue box to represent the different effectiveness clusters) . 199

7.1 NOVI Plan schema. 205

7.2 First step of the application: Design of the disaster management model corresponding to the case study 207

7.3 Illustration of the concept hierarchy resulting from the interpretation process of the data sample presented in Table 7.1 and the defined mapping sample is shown in Table 7.2 210

7.4 Illustration of some individuals resulting from the interpretation process of the data sample presented in Table 7.1 210

7.5 Disaster modeling composed of an event with three configurations of casualties quantity 211

7.6 Modeling of organizations and their resources in terms of role. . . . 213

7.7 Modeling of roles and their associated services 214

7.8 Modeling of services and their associated procedures. 215

7.9 Modeling of NOVI plan and its sub-procedures. 217

7.10 Modeling of the service collection noria and its associated protocol. 218

7.11 Result illustration of reasoning on an event representation. 219

7.12 Second step of the application: Simulation modeling processes . . . 220
7.13 Conceptual simulation modeling resulting from rule-based reasoning 221
7.14 Building artifact modeling resulting from rule-based reasoning . . . 222
7.15 Road artifact modeling resulting from rule-based reasoning 222
7.16 Results of the actor agent modeling for the role *semDM:fireman* . . . 223
7.17 Results of the agent's action modeling for the action *semDM:transport_casualty* 224
7.18 Results of the BDI agent modeling for the role *semDM:fireman_officer* 224
7.19 Results of desire and plan modeling for the agent *semMAS: agent_fireman_officer* 225
7.20 Results of belief modeling for the agent *semMAS: agent_fireman_officer* 226
7.21 Modeling results of agents known by the agent *semMAS: agent_fireman_officer* 226
7.22 Result of axiomatic-based inference on the modeling of the agent *semMAS:agent_fireman* and the action *semMAS: transport_756daed* . . 227
7.23 Result of rule-base inference on skills for the agent *semMAS:agent_fireman* and the action *semMAS: transport_756daed* 228
7.24 Result of rule-base inference on the location of the agent *semMAS:agent_fireman* 228
7.25 Result of rule-base inference on the concept *semMAS:SimpleBDI* specificities 229
7.26 Third step of the application: Programming and execution of simulations 229
7.27 Implementation results of the first block of the programmed GAML model 231
7.28 Parts of SemMAS ontology for the global implementation 233
7.29 Implementation parts result of the second block of the programmed GAML model 234
7.30 Example of implementation results of the third block type of the programmed GAML model 235
7.31 Parts of SemMAS ontology for the implementation of agent examples 236
7.32 Examples of implementation result of the fourth block type of the programmed GAML model 237

7.33 Experimental model of the studied case represented in SemMAS ontology . . . 238
7.34 Code generated from the experimental model . . . 238
7.35 Screenshot of the simulation . . . 241
7.36 Fourth step of the application: Clustering-based analysis . . . 241
7.37 Effectiveness resulting from effectiveness-based cluster 1 surrounded in orange, 2 surrounded in blue, and 3 surrounded in purple. . . . 249
7.38 Effectiveness resulting from effectiveness-based cluster 4 (surrounded with different colors for the different sub-groups of effectiveness). . . . 250
7.39 Effectiveness resulting from effectiveness-based cluster 5 (surrounded with different colors for the different sub-groups of effectiveness). . . . 251
7.40 Effectiveness resulting from effectiveness-based cluster 6 (surrounded with different colors for the different sub-groups of effectiveness). . . . 252

A.1 The semiotic triangle . . . 324

B.1 Geometry and data set specification [Prudhomme et al., 2017a] . . . 352
B.2 Identification by value analysis [Prudhomme et al., 2017a] . . . 354
B.3 Process of linking with Semantic Web resource [Prudhomme et al., 2017a] . . . 356
B.4 Example of SHACL rule corresponding to the part presented in Code 6.16 . . . 357

List of Tables

3.1 Limits of existing multi-agent simulations for plan assessment . . . 62

3.2 Overview of existing ontologies in relation to the main multi-agent simulation concepts 65

3.3 Platforms comparison according to application domains related to the disaster management domain 68

3.4 Observed variables in disaster management simulation (IP: Impacted population, NbC: Number of Casualty, NbD: Number of Dead, IB: Impacted building, IA: Impacted area, TP: Time performance, SQ: Success quantity, HM: Human means, RM: Resource means) 70

3.5 Platforms comparison according to application domains related to the disaster management domain 74

3.6 Existing ontologies according to required terms (1) 91

3.7 Existing ontologies according to required terms (2) 92

3.8 Relevant approaches for a knowledge-driven plan assessment through simulation and limits to overcome 100

4.1 Overview of requirements to overcome the specific limits of approaches tackling the limits observed in plan assessment 103

4.2 Overview of proposed approaches to solve requirements identified for automating the plan assessment 108

7.1 Sample of the road dataset to process 209

7.2 Sample of mapping between concepts from SemDM and Wikidata . 209

7.3 Result of the clustering based on effectiveness score 243

7.4 Clusters 2, 3, and 4 resulting from the effectiveness-based clustering 245

7.5 Results of the clustering applied on each variable for the effectiveness-based clusters 246

7.6 Result of clustering applied to the effectiveness-based cluster number 4 according to the *casualty_quantity* variable-based clusters 246

8.1 Table representing the number of agents by agent type 262
8.2 Table representing the assessment of environment inccessibility . . . 263
8.3 Table representing the assessment of environment instability 263
8.4 Table representing the assessment of interaction complexity in the environment 264
8.5 Table representing the assessment of environment complexity 264
8.6 Table representing the assessment of the mode of action choice . . . 265
8.7 Table representing the assessment of the rationality 265
8.8 Table representing the assessment of the proactivity 266
8.9 Table representing the assessment of the autonomy in the organizational structure 267
8.10 Table representing the assessment of the autonomy 267
8.11 Table representing the assessment of the response rapidity in a timely fashion 268
8.12 Table representing the assessment of the reactivity 269
8.13 Table representing the assessment of the ability to manage different situations 270
8.14 Table representing the assessment of the ability to respond to new External stimuli 271
8.15 Table representing the assessment of the adaptability 271
8.16 Table representing the adequacy assessment of the MAS rationality to its environment 273
8.17 Table representing the adequacy assessment of the MAS autonomy to its environment 274
8.18 Table representing the adequacy assessment of the MAS reactivity to its environment 274
8.19 Table representing the adequacy assessment of the MAS adaptability to its environment 275

A.1 Different levels of knowledge representation 320

A.2 Comparison of vector and raster data 339
A.3 Organizational structure of disaster management in France according to ORSEC strategy 341
A.4 Comparison between disaster management in France and Germany according to strategic, operational and tactical plans 347
A.5 Overview of disaster preparedness concepts according to the strategic, operational and tactical aspects 348

A Related work appendix

A.1 Knowledge engineering

The goal of knowledge engineering is "*turning the process of constructing [Knowledge-based systems] from an art into an engineering discipline*" [Studer et al., 1998]. The overall consensus in 1998, states that knowledge engineering can be seen as a modeling activity [Studer et al., 1998]. Knowledge modeling consists in to create a model able to gather all knowledge and information. This model is the root of a knowledge base, which contains all information and knowledge and allows their management. Knowledge bases are at the heart of the knowledge-based systems. That is why this section firstly explains the background of knowledge representation and secondly describes the knowledge base content and working.

A.1.1 Knowledge representation

It exists several levels of knowledge representation (KR) that have been presented by Guarino in [Guarino, 1995]. The table A.1 presents these different levels of knowledge representation.

Level	Primitives	Interpretation	Main features	Language
Logical	Predicates, functions	Arbitrary	Formalization	Predicate logics
Epistemiological	Structuring relations	Arbitrary	Structure	KL-ONE
Ontological	Ontological relations	Constrained	Meaning	OWL
Conceptual	Conceptual relations	Subjective	Conceptualization	
Linguistic	Linguistic terms	Subjective	Language dependency	Natural language

Table A.1: Different levels of knowledge representation

Logical level The logical level is the first level of KR formalism. It corresponds to the first-order logic, which respects to ontological choices. Its description logic is based on predicates and functions, allowing a neutral and very general representation to be done. The ontological neutrality is not very adapted to represent the knowledge because it is necessary to take a point of view to describe a domain. This point of view is one of the domains.

Epistemological level The term epistemology represents a philosophical field where nature is used as a source of knowledge. For the epistemology, the process of inference is essential. The epistemological level uses the language KL-ONE , which is an overlayer of predicate logic because it adds a knowledge-structuring mechanism that is favorited for the inference process [Description, 2008, Guarino, 1995]. The weakness of this level is its lack of meaning in its KR.

Ontological level The work of two previous levels has two major issues. The first is their lack of meaning and the fact that they do not consider the ontological commitment. The second is they have an arbitrary interpretation, which does not allow the intended models to be made explicit. Unlike epistemology, an ontology is a study of the world organization and nature independently to the form of our knowledge. This level adds so two elements. The first element is the meaning according to a specific ontological commitment. The second element is the constraint, which allows restricting the primitives' semantics to exclude non-intended models and thus facilitate large-scale knowledge integration. This level enables doing a reused KR.

Conceptual level The primitives of the conceptual level represent language-independent concepts like necessary actions or thematic roles. This level can represent a "standard" for a domain, so it corresponds too to a specific type of ontology. However, it is possible to use this level to create an ontology, only if principles (allowing the definition of the basic ontological categories) are based on a well-defined ontological level. The well-funded principles are fundamental mostly to specialize in logical relations into categories like parts, qualities, properties, states (cf. [Guarino, 1995]).

Linguistic level

The linguistic level is simply a natural language. Like the other levels, it uses signs (or here words), which refers to something and invokes a concept (cf. the semiotic triangle, figure 2.1). It is a system of KR, but it is not formal, and it cannot be used for the sharing between machines.

Among these different levels, the ontological level is the knowledge representation level used to represent the knowledge explicitly through its semantic. It aims at creating a knowledge base that defines concepts with meaning understandable both by humans and machines. The ontological level is chosen for its right balance between the linguistic level of humans and the logical level of computers. The ontological representation of knowledge is done through an ontology. An ontology comes from the Greek etymology of two words (onto: "being" and logia: "logical discourse") and means the "study of being". The first subsection presents the role of an ontology in computer science, and the second subsection explains how extracts knowledge to design ontology.

A.1.1.1 Ontology

An ontology has three aims. The first aim is to analyze the knowledge on a domain to transform what is implicit on the domain in an explicit form. The second aim is to distinguish the knowledge on a domain and the operational knowledge. Finally, the third aim of an ontology is to reuse the knowledge on a domain in sharing its understanding between people and machines.

Several definitions have been given to ontology over time. Initially, it is in 1993 that Grubber defined the notion of ontology as an *"explicit specification of a conceptualization"* [Gruber, 1993]. Then, in 1997, Borst brought out the notion of sharing in the ontology, by defining this one as a *"formal specification of a shared conceptualization"* [Borst, 1997]. Finally, in 1998, Studer et al. merged these two previous definitions to define the ontology as *"a formal, explicit specification of a shared conceptualization"* [Studer et al., 1998]. Nicola Guarino, Daniel Oberle, and Steffen Staab have decomposed this definition and explain an ontology through three major points [Guarino et al., 2009]:

- Defining a conceptualization.
- Explaining a formal, explicit specification.
- Presenting the importance of the sharing for an ontology.

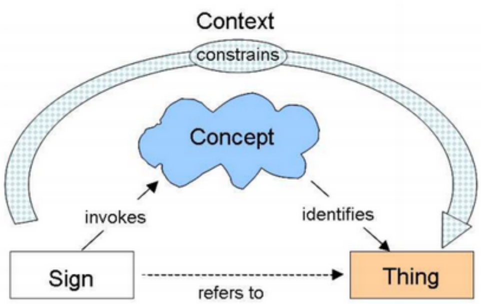
Conceptualization A conceptualization is an abstract model of a real phenomenon. It is a set of relevant concepts that characterize a domain according to the point of view (a limited view of the world). The conceptualization has been explained as an extensional relational structure by Genesereth and Nilsson: "An extensional relational the structure is a tuple (D, R) where D is a set called the universe of discourse and R is a set of relations on D" [Genesereth, Michael R. and Nilsson, N. J., 1987]. The authors of [Guarino et al., 2009] have highlighted the main issue of this definition, which is R depends on a specific world. Considering to define a conceptualization as an extensional relation would require for representing all world states to take into account all cases. These authors claim that a conceptualization being based on concepts must be focused on them to have a general representation independently of a single world state. When a detail of the representation changes, the conceptualization must not change but to adapt. To manage this evolution without change of conceptualization, the notion of the world and one of the world states have been introduced. Based on the definitions of a world and an intensional relational structure in [Guarino et al., 2009], Guarino has defined a conceptualization as follows: "An intensional relational structure is a triple $C = (D, W, R)$ with D a universe of discourse, W a set of possible worlds, and R a set of conceptual relations on the domain space $< D, W >$." The definition of world state shows that a world state is a way to see the world and to link the universe of discourse D and relations R. Thus, by adding world W, which is a set of world state we can add a world state in W to evolve the ontology without changing the basis of ontology, which are D and R. In simple words, a conceptualization is a description of the world (according to the point of view) through concepts.

Formal, explicit specification The second major point in the definition of an ontology is a "explicit formal specification". Behind these words, two fundamental notions are hidden:

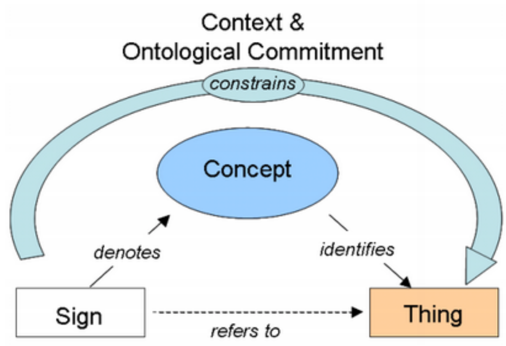
1. *Explicit specification.* A conceptualization is the representation of a world that evolves with the time according to a domain and relations of this domain related to the world. The universe of discourse D is a set of terms which concern a specific domain. In the beginning, when these terms have been determined, they are an implicit representation in the mind of people. But the meaning of a term for a person is not always the same meaning for all people. So, it needs to specify our conceptualization explicitly. This explicit specification consists of defining these terms and their explicit relations.

2. *Formal.* The explicit specification needs a form to express it. The use of the word "formal" implies that the form to represent the specification must be readable by a machine that excludes the natural language.

Sharing knowledge The notion of sharing in an ontology is not mandatory; that is why the first definitions do not take into account this notion. But the use of an ontology without the notion of sharing is minimal. The sharing of knowledge is what that have always grown the world in all fields. The notion of **shared** ontology is so essential because it represents the sharing of knowledge and information in the field of computer science. The sharing of knowledge and information is essential in several domains (e.g., the domain of supply chain [Mendes Calo et al., 2012] and for government [Mendes Calo et al., 2012, Zhang et al., 2005]) because it allows the efficiency improvement. For example, in the domain of supply chain, it provides more efficient supply chains and more effective organizations [Mendes Calo et al., 2012], and for government, it offers strategic advantages to improve decision making [Zhang et al., 2005]. This sharing is done through information technology [Zhang et al., 2005], where ontology is often used to represent knowledge (e.g., [GeoPii, Integrasys, 2014, Shafiq et al., 2012, Beneito-Montagut et al., 2013, Babitski et al., 2011]). In default of obtaining an ontology as complete and formal as the theory desires it, an ontology must at least provide the interoperability by *well-founding* and *well-chosing* the sense of basic primitives without forgot to axiomatize them for the general understanding. The semiotic triangle of Ogden and Richard (see Figure A.1(a)) represents the "thinking" system in the sharing of knowledge.



(a) The semiotic triangle of Ogden and Richard



(b) The semiotic triangle adapted with a shared ontology according to [Guarino et al., 2009]

Figure A.1: The semiotic triangle

This semiotic uses three linked notions that depend on the context. The sharing of knowledge is based on the signs (or words), which refers to something. When an agent receives a sign, this sign is "treated" by his mind to identify the thing which it refers to. This treatment consists of invoking a concept corresponding to this sign according to a specific context. The problem is that a concept for a specific context is not the same for all minds. In the same way that the system of "thinking" is not the same for all people, a word or a sign can have two different meanings for two people that lead to incomprehension or an inadequate understanding between them. That is why it limits the sharing between human people and between machines. When a community of people wants to share elements on a domain, they define a set of standards to all people can "talk the same language" and share these elements of this domain. The role of a common ontology for all agents is the same as shared standards by a community. Because *"the usage of signs are implied by the logical theory specifying the ontology"* [Guarino et al., 2009], a common ontology for all agents implies they "think" in the same way since they use the same logical theory to reference something. It is what is illustrated by figure A.1(b) of the semiotic triangle revisited with a shared ontology by [Guarino et al., 2009].

The role of an ontology is to represent formally and explicitly knowledge of a world through concepts to share this knowledge with machines and thus endow the machine with more capabilities. The next subsection discusses how to capture knowledge from experts and how to design ontologies.

A.1.1.2 Ontology Design Methodology

The authors of [Uschold and Gruninger, 1996] were in the first to propose a global methodology to build an ontology. Their methodology is composed of four phases, which all must be done according to a set of criteria essential to the ontology building. The four phases presented in this methodology are:

1. Identify Purpose and Scope;
2. Building the Ontology: ontology capture, ontology coding, integrating existing ontologies;
3. Evaluation;
4. Documentation.

Essential criteria for ontology design decisions The essential criteria for ontology design decisions aim at orienting choices of design toward aspects of the sharing and reuse of knowledge. Grubber has presented these criteria in [Gruber, 1995]) as follows:

1. Clarity: it aims to go toward the most of "*complete definition*" and the most of definition is "*stated in logical axioms*".
2. Coherence: it aims to verify that all defining axioms are consistent. For that, it needs to verify if "*a sentence that can be inferred from the axioms contradicts a definition*". As long as you find contradictions, you must change and upgrade your ontology.
3. Extendibility: it aims to be able to define new terms with the existing vocabulary without the need to change the current definitions.

4. Minimal encoding bias: An encoding bias is used when you want to specialize a detail, which corresponds to the aspect of encoding and not an element of the knowledge. But it needs to minimalize this encoding bias to share knowledge because agents who use the knowledge can be implemented in different systems.

5. Minimal ontological commitments: It corresponds to define the knowledge, the vocabulary of a field with sufficient elements to be used but without so many details which could prevent the sharing of knowledge.

During the overall design process, it is necessary to keep these criteria in mind. Moreover, it is required to verify that the ontology design respects all criteria after each version of the plan. Sometimes, it needs to do some trade-offs between all criteria, but these trade-offs are not on the choice of one criterion rather than another. The best trade-off between criteria is to obtain for each criterion the highest level, without this level invalidate another criterion.

Ontology Design After the identification of the purpose and scope of the ontology, the base of its building is the capture of knowledge. Capture the knowledge of a domain such as a disaster management require to capture and collect knowledge from different experts, sometimes having a very different point of view of a disaster. This diverse knowledge and point of view must be gathered to represent disaster management preparation globally. In such a context, the authors of [Uschold and Gruninger, 1996] propose to capture ontology through four steps. The first step consists in defining the scope through collaborative brainstorming to identify relevant terms and phrases to the group then them into a subgroup of work areas. The second step consists in producing definitions of terms and reaching agreement on terms overlapping different works areas. The third step is the review of the definitions. The last step consists in designing a meta-ontology by identifying the main terms of the domain and using the natural language definitions as an implicit requirement specification. The term "meta-ontology" is used in [Uschold and Gruninger, 1996] to speak about "the basic terms that will be used to specify the ontology (e.g., class, entity, relation); this is often called a 'meta-ontology' because it is in essence, the [underlying] ontology of representational terms that will be used to express the main ontology" [Uschold and Gruninger, 1996]. After the capture of ontology, a representation language must be chosen to code the ontology. The capture and coding of an ontology must consider existing ontologies by evaluating if and how these ontologies can be integrated.

Later, Natalya F. Noy and Deborah L. McGuinness in [Noy and McGuinness, 2001] propose a more detailed guide to identifying components (e.g., class, the entity, relation) of an ontology. These authors present seven steps of ontology design, which looks like steps of the oriented object model. However, the big difference between them, is that the design of an oriented object model is "*based on the operational properties of a class, whereas* [the design of an ontology is] *based on the structural properties of a class*". This method aims to define concepts with classes, slots, which are the properties, and facets which have the role of restriction.

1. The first step of this guide is to determine the domain and scope of the ontology. For that, it needs to answer questions as to what is the domain, the aim of the ontology? Who will be the users? Then, it is necessary to find all types of questions, which the ontology will answer to.

2. The second step is to consider ontologies that exist and translate them if it is necessary to reuse them (c.f. Ontolingua, T. R. Gruber [Gruber, 1993]).

3. The third step is to determine all the most relevant terms in the ontology. The questions which we need to ask are: "*What are the terms would we like to talk about it? What properties do those terms have? What would we like to say about those terms?*"

4. The fourth step is to define the classes and the class hierarchy. To do that, there are three approaches: the process top-down which starts with a general view and goes toward a specific picture, the process bottom-up which is the reverse of top-down and which begins with a specific view to go toward a general perspective, and the last is a combination of both of the previous process. The choice of the approach depends on the personal view of the domain.

5. The fifth step is for defining class properties. With all terms and classes which have been defined previously, it must determine among the remained terms which class they characterize.

6. The sixth step is to define the facets of the slots. Three main types of aspects are necessary to explain. These facets are the slots cardinality, the slot-value type (e.g., String, number, boolean), the domain and range of a slot (for example, if we take a niche which is *produces*, its range is *Wine* and its domain is *Winery*).

7. The last step is to create instances of the classes. For that, it needs to choose a class and fill its slots-value to define a specific type of this class.

This guide makes an intervene distinction between the components of an ontology (e.g., class, instances, slots more commonly called properties, and facets more commonly called restrictions). An ontology aims at gathering knowledge in a single place; in other terms, it aims at constituting a knowledge base. Therefore, the next section explains the components and mechanisms of a knowledge base.

A.1.2 Knowledge base

In the DL[1] context, a knowledge base is composed of a Terminological Box (TBox) and an Assertional Box (ABox). The TBox is a controlled vocabulary that allows the ontology's specification. It corresponds to the meta-ontology discussed in [Uschold and Gruninger, 1996]. The controlled vocabulary is expressed through classes representing the concepts and properties representing the relationships between the concepts. Each class is described through description logic to represent its definition in natural language. The ABox represents facts, and information expressed through the controlled vocabulary of the TBox. The Abox is composed of instances, which are specific individuals of a concept. The facts and data are represented by linking individual between then or with value through properties.

In the Semantic Web context, a knowledge base is represented by a semantic graph through **R**essource **D**escription **F**ramework (RDF) [Beckett, 2004, Pan, 2009]. The graph represents a set of enunciations. An enunciated is a triple (S, P, O) which corresponds to **S**ubject, **P**redicate (a property), and **O**bject (the value of the property for this subject). These triple are RDF links that can describe relations, identity, and vocabulary in the same graph. **R**essource **D**escription **F**ramework **S**chema (RDFS) [McBride, 2004] allows describing more details than a triple of RDF. It allows modeling classes, instance, properties, and hierarchical relations. The hierarchical relationships create the taxonomy of classes and properties. RDFS being a very light description of classes, the ontologies need a language to be defined completely. This language is the **W**eb **O**ntology **L**anguage (OWL) [McGuinness et al., 2004] initiated in 2004. Its aim is to constraint the cardinality, define classes with a specification of constraint on properties, identify disjoint class, define a class with a boolean combination, and characterize some properties (transitivity, functions, reverse). OWL adds a vocabulary to RDFS with a formal semantic well defined. It is based on a logical description, and it exists several types of OWL: OWL DL, OWL Lite, and OWL Full. Each one of these types has different constraints. Later, OWL has evolved into OWL2 [Hitzler et al., 2009b]. According to the power of lan-

[1]DL: Description logic

guage, the complexity of inference increases. The two next subsections present the description logics and the inference.

A.1.2.1 Description Logic

The description logic provides different expressiveness of ontologies. All expressiveness is based on the minimal expressiveness called attributive language (AL) [Schmidt-Schauß and Smolka, 1991]. According to [Baader and Nutt, 2010], the attributive language is based on:

C, D (concept descriptions) $\rightarrow A$ (atomic concept) $|\top$ (universal concept) $|\ \bot$ (bottom concept $\emptyset$) $|\neg A$ (atomic negation) $|C \sqcap D$ (intersection) $|\forall R.C$ (value restriction) $|\exists R.$ (limited existential quantification).

On this basis, further expressiveness has been defined:

- ALC = AL + union ($C \sqcup D$, where C, D are concepts, e.g. $Animal \equiv Vertebrate \sqcup Invertebrate$)
  + full existential quantification ($\exists R.C, where \text{C} a concept and \text{R} a role$)
- ALCN = ALC + number restrictions ($\leq nR| \geq nR$, where n a number and R a role)
- S = ALC + transitivity (e.g. $hasAncestor \circ hasAncestor \sqsubseteq hasAncestor$)
- R = role chains (e.g. $hasParent \circ hasBrother \sqsubseteq hasUncle$)
- H = role hierarchies (R, S are roles, $R \sqsubseteq S$)
- O = nominals (C is a concept and a, b, c are individuals, $C \equiv \{a, b, c\}$)
- I = inverse roles (R, S are roles, $R \equiv \neg S$)
- Q = qualified cardinality restrictions ($= nR| \leq nR| \geq nR$, where n a number and R a role)
- F = role functionality, a particular case of cardinality ($\top \sqsubseteq \leq 1.\top$, e.g. the role marriedWith)
- D = datatypes (e.g. $hasAge(jack, 13^{\wedge\wedge}xsd : integer)$)

The expressiveness of an ontology is identified according to the definitions contained in its TBox.

TBox description logic The terminological axioms are mostly in the form of inclusions ($C \sqsubseteq D$, where C, D are concepts or $R \sqsubseteq S$, where R, S are roles) and equalities ($C \equiv D$, where C, D are concepts or $R \equiv S$, where R, S are roles) [Baader and Nutt, 2010]. An atomic concept followed by equality aims at defining this concept. Let's take the example of the terminology with Concepts about animal world relationships to illustrate concept definitions:

$$Vertebrate \equiv Animal \sqcap \exists hasCharacteristic.Backbone$$

$$Invertebrate \equiv \neg Vertebrate$$

$$Human \sqsubseteq Mammal \sqsubseteq Vertebrate$$

ABox description logic The ABox of a knowledge base corresponds to the description of a world state. Assertions of the ABox introduce individuals by assigning them names through concepts (i.e. $C(a)$, where C a concept and a an individual) and properties through roles (i.e. $R(a, b)$, where R a role and a, b individuals) [Baader and Nutt, 2010]. Let us take an example of assertions for the domain of the animal world, whose TBox examples are given previously:

$$Human(jack)$$

$$Backbone(human\_backbone)$$

$$hasCharacteristic(jack, human\_backbone)$$

$$hasVertebrae(human\_backbone, 33^{\wedge\wedge}xsd : integer)$$

In the Semantic Web, the different types of OWL do not provide the same expressiveness. Indeed, OWL-Lite has a SHIF(D) expressiveness, OWL-DL has a SHOIN(D) expressiveness, and OWL2-DL has an SROIQ(D) expressiveness, whereas OWL-FULL and OWL2-FULL are not description logic ([Hitzler et al., 2009a], p.167).

The expressiveness of an ontology has an impact on the complexity of its inference. According to [Hitzler et al., 2009a] (p.207), the complexity of ALC, SHIQ, SHOQ, SHIO expressiveness is ExpTime, whereas the complexity of SHOIQ is NExpTime, and the complexity of SROIQ is N2ExpTime. The next section presents the principles of inference.

A.1.2.2 Inference

The inference is a reasoning on a description logic system [Baader and Nutt, 2010]. The inference has two leading roles in a knowledge base. The first role is to check the consistency of the knowledge base. Let us take the example of a TBox within $Woman \equiv \neg Man$ to define that concepts *Woman* and *Man* are disjoint and an ABox within $Woman(Dominique)$ and $Man(Dominique)$ to define that the individual *dominique* is both a woman and a man. In such an example, the inference highlights the inconsistency between definitions from the TBox and the assertions of the ABox. The second role of the inference is to add axioms in the TBox and assertions in the ABox. There are two processes of inference: the inference based on the description logic and the inference based on rules.

Description logic-based inference The inference based on the description logic analyzes both the TBox and the ABox. In the TBox, this inference process analyzes four properties of a concept [Baader and Nutt, 2010]:

- The satisfiability, which corresponds to verify that each concept satisfies the TBox;
- The subsumption, which corresponds to identify every concept C that is subsumed by another concept D concerning the TBox, to add the axiom $C \sqsubseteq D$.
- The equivalence, which corresponds to identify every concept C that is equivalent to another concept D concerning the TBox, to add the axiom $C \equiv D$.
- The disjointness, which corresponds to identify every concept C that is disjoint from another concept D concerning the TBox, to add the axiom $C \equiv \neg D$.

In the ABox, this inference process deduces new assertions concerning the TBox. Assertions can be $C(a)$ or $R(a, b)$, where C a concept, R a role, a an individual and b an individual or a data (e.g. string, number). For example, from the TBox within $Motorcycle \equiv Vehicle \sqcap \forall hasWheel.2$, and the inference can deduce the subsumption between *Motorcycle* and *Vehicle*, and thus add $Motorcycle \sqsubseteq Vehicle$. By adding into the ABox, the assertion $Motorcycle(m)$, then the inference deduces $Vehicle(m)$ and hasWheel(m,2).

Rule-based inference The inference based on rules aims at going through the knowledge base with rules to add new assertions. A rule is composed of two parts: premises and conclusions. For each rule, the inference process checks the knowledge base with the premises of a rule. If the premises are satisfied, then the inference process adds assertions corresponding to the conclusion part of the rule.

The premises part is introduced by IF, whereas the conclusion part is introduced by THEN. Let us take the following example:

- an ABox within $Plan(p)$, $protects(p, koeln)$, $appliedFor(p, 845)$ and $hasWaterLevel(koeln, 845)$;
- the following rule: IF $protects(?x, ?y)$ AND $appliedFor(?x, ?v)$ AND $hasWaterLevel(?y, ?v)$ THEN $isActivated(?x, true)$

In such an example, the inference process adds the new assertion $isActivated(p, true)$.

Inference engine An inference engine achieves the inference process, also called a reasoner, and depends thus on the capabilities of the reasoner engine. For example the reasoner engine Fact++ [Tsarkov and Horrocks, 2006] allows reasoning on SHOIQ description logic (adapted to OWL-DL and OWL2-DL), whereas the reasoner engine Pellet [Sirin et al., 2007] allows hybrid reasoning on OWL-DL and rules. It exists different syntaxes of rules as the Semantic Web Rule Language (SWRL) used by the Pellet reasoner or the Shapes Constraint Language (SHACL) [Knublauch and Kontokostas, 2017], a W3C recommendation. In addition to these two types of inference, the choice of reasoner engine depends on the characteristics of the inference process that they propose, for example, monotonic versus non-monotonic inference or open versus close world inference.

Monotonic versus non-monotonic inference An inference results in new assertions. There are two approaches to manage these new assertions. The first approach is to save the new facts in another place than the knowledge base. This approach is monotonic inference. It guarantees the preservation of the consistency of the knowledge base. The second approach is non-monotonic inference. It consists of adding new facts in the knowledge base. The risk is to obtain an inconsistent knowledge base. However, in [Frankish, 2005], it is possible to find ways to manage this problem of non-monotonic inference and to benefit, thus, of the advantage of a non-monotonic inference. The advantage of the non-monotonic the inference is that it doesn't restrict the inference power contrary to the monotonic inference [Lange and Zeugrnann, 1993]. The description logic-based inference (presented previously) is monotonic, whereas rule-based inference is a non-monotonic inference.

Open world versus Close world inference The difference between open and close world inference is that a close world inference assumes that something, which is not defined in the knowledge base does not exist. On the contrary, the open-world inference assumes a lack of knowledge. It considers thus that it is not because something is not defined in the knowledge base that it does not exist.

A.1.2.3 Knowledge querying

Simple Protocol And RDF Query Language (SPARQL) is a language which allows retrieving and manipulating knowledge expressed according to the RDF format [Prud'hommeaux and Seaborne, 2008]. With SPARQL, it is possible to search, add, modify, and remove data in a knowledge base. SPARQL is used to query a database specially designed to store RDF data. This specific database is called triplestore because it stores a unique type of data, which are RDF triples. Initially, SPARQL query can have four forms [Prud'hommeaux and Seaborne, 2008]:

- SELECT query used to retrieve raw values from a SPARQL endpoint that is returned in a table format.
- CONSTRUCT query used to create a valid RDF graph from information extracted from the SPARQL endpoint.
- ASK query used to check assertions into a SPARQL endpoint by answering with a boolean (True or False) to the query.
- DESCRIBE query used to retrieve an RDF graph from the SPARQL endpoint, containing all information (e.g., relationships, concepts) about variables targeted by the query.

Later, SPARQL has been extended to allow the update of the RDF graph by adding new query forms [Gearon et al., 2013]:

- INSERT query used to add information on a SPARQL endpoint,
- DELETE query used to remove information from the SPARQL endpoint.

A.2 Spatial data models

A spatial data is a georeferenced data, representing geographically objects or a phenomenon. Each element of spatial data is described through a spatial (geometrical or graphical) component and an attribute component. The spatial component describes the location of objects or the spatial distribution of a geographic phenomenon [Neteler and Mitasova, 2008]. The attribute component describes the properties of an element [Neteler and Mitasova, 2008]. There are three types of attributes: the attribute value of an element, the relations between the data elements, and the quality of the data [Fazal, 2017]. The attribute value describes the properties of the data elements.

The relation attributes are relations information that cannot be calculated from the coordinate or the structure of the data elements. Finally, the quality includes firstly, quality information as the graphical accuracy, the updating information (when and how), and the resolution [Fazal, 2017]. It includes, secondly, quality assessment according to the spatial component and the attribute components. The quality assessment depends on the extent of geographical coverage, the logical consistency between geometry and attributes, discrete versus continuous representation, and the relevance of a data [Fazal, 2017]. These four elements determining the data quality are related to the data model used to represent objects or a phenomenon. It exists two different data models that depend on the graphical representation of their spatial components. The first data model is vector data. Its graphical representation corresponds to a geometry defined as a point, a line, or an area given by their coordinates [Neteler and Mitasova, 2008]. The second model is the raster data. Its graphical representation corresponds to a field representation. A field representation is characterized by regularly distributed points or an area element (pixel) in the space having an assigned value (a number or no-data) [Neteler and Mitasova, 2008]. The two next subsections further explain these two models.

A.2.0.1 Vector data

Vector data represents a set of features. Each of these features has a shape represented by a geometry and attributes that describe the feature properties. That is why vector data are the most adapted model to represent objects whose shape can be represented geometrically (e.g., a road by a line, a building by a polygon).

A geometry as a spatial representation. The most straightforward geometry is a point. Coordinates X, Y, and optionally Z, which represents the height above sea level, describe a point. The second geometry is a line or polyline when several line segments describe the geometry. A line is a segment defined by two extremity points, and a polyline is a sequence of joined points. The last geometry is a polygon representing an area. A polygon is also a sequence of points, contrary to a polyline, the first point, and the last point is the same. Each geometry is a composition of a simplest one, until the point which is based on coordinates. Thus, each geometry is based on coordinates, and these coordinates depend on a Coordinate Reference System (CRS).

Coordinate reference system. It exists two types of coordinate reference system: a geographic coordinate reference system and a projected coordinate reference system. The most commonly referenced system used is the geographic coordinate reference system based on longitude and latitude.

A geographic coordinate system uses degrees of latitude and longitude and sometimes a high value. The most popular geographic coordinate system is called WGS 84. The reference line for latitude is the equator. Each line of latitude runs parallel to the equator and are equally spaced from each other, from North to South (or South to North). Each hemisphere is divided into ninety sections. The reference line of longitude is the meridian that runs from the North Pole to the South Pole through Greenwich in England. Each longitude line is perpendicular to the equator and converges at the poles. Longitude lines are measured from 0 to 180 degrees from East to West according to the reference line of longitude. Values from West to East are negative measures from 0 to 180 degrees. Each degree is divided into minutes and seconds to provide an acceptable level of accuracy. A degree represents sixty minutes but also 3600 seconds.

A projected coordinate reference system is based on a two-dimensional coordinate reference system defined by two axes at right angles: X (horizontal axis) and Y (vertical axes) or by a three-dimensional coordinate reference system, where the third dimension is added through a third axis z at right angles to the X and Y-axis. For both hemispheres, the projected coordinate reference system has its origin on the equator at a specific longitude. However, in the southern hemisphere, the Y-values increase southwards, and the X-values increase to the West, whereas in the Northern hemisphere, the Y- values increase northwards, and the X-values increase to the East. An example of a projected coordinate system is the Universal Transverse Mercator (UTM). UTM CRS is generally used all over the world. The UTM is a cylindrical projection in two dimensions, where the axis of the cylinder lies in the equatorial plane, and the line of tangency is the Central meridian. The world is divided into 60 equal Zones corresponding to a wide of 6 degrees in longitude from East to West. A position of coordinate is defined by a UTM zone (1 to 60) following by an "s" if it belongs to the southern hemisphere, a northing value (y), and an easting value (x). The northing value is the distance from the equator, and the easting value is the distance from the central meridian. The northing value of a location in The computation of the southern hemisphere adds a false northing value of 100.000.000 on its original negative northing value. Such a process avoids negative values. Similarly, for a negative easting value, the computation adds a false easting value of 500.000 m.

Data attributes. The coordinate reference system is one of the data attributes that impact its quality, and more precisely, the accuracy of its spatial representation. The vector data is the richest model in attribute quantity. In addition to the quality attributes that can be provided through metadata, a vector data is composed of an attribute values table and can be accompanied by a file describing relationship attributes (also called the topology). The table structure of attribute values allows gathering all attribute values of a feature in a row thanks to representing an attribute type by column. Rules can represent the topology attributes, also called relations attributes. These attributes allow the detection and correction of digitizing errors.

A.2.0.2 Raster data

In raster data, the real world is represented through uniform and regular cells. That is why raster data is the most adapted model to represent information continuously across an area. Among raster data, satellite and aerial images are often used to depict the surface of the real world (e.g., forest density) or a phenomena repartition (e.g., flooded area). Raster data also represent more abstract ideas resulting from a computation (e.g., digital elevation model, population repartition, or a level of flood risk).

Cell as a spatial representation. These cells are more usually square or rectangle, but can also be triangular or hexagonal. A raster-based on rectangular or square cells is also called a grid model. The resolution of the data depends on the size of cells.

Data attributes. As vector data, raster data also has quality attributes as the resolution previously announced. However, the quantity of attribute values is much more limited than in vector data, since each Cell contains a unique attribute value. This value is mostly a numerical value, but can also be a text value. There are three types of raster representations according to the value contained in a cell:

- Binary representation: The cell value equals one if there is the feature at this cell position or 0 if it is not the case. This binary representation of raster brings only the information of a feature location, whose type is implicitly represented by the data. Let us take the example of a binary raster representing the forest, cells expressing the location of forest have the value 1, whereas the others have the value 0.

- Enumeration representation: The cell value is a numeric value or a text value to describe the type of feature located at the corresponding Cell. The enumeration representation of a raster brings the feature type explicitly with its location. In case of a text value, the cell value can directly provide the name or an abbreviation of the feature type, whereas, in the case of a numeric value, each numeric value is associated to a feature type. Let us take the example of a raster representing land cover with the following values: 1 for the forest, 2 for the crop, 3 for water, and 4 for an artificial area as urban. Each cell location of such a raster map is classified in one of these four categories, and the value associated with the classification type is then assigned to the Cell.
- Numeric representation: The cell value is an integer or a float describing or recording the value of a phenomenon. Let us take the example of a raster representing the risk of an earthquake; an integer value can be assigned to each level of risk: 1 for very high risk, 2 for high risk, 3 for low risk, and 4 for inexisting risk. Thus, one of these four values is assigned to each Cell. The value assignment depends on the risk estimation at the cell location. Another example of a numeric raster is a raster representing the level of water resulting from a flood simulation of substantial rainfall. In this example, the float value assigned to each Cell corresponds to the level of water at the location cell in the case of a strong rainfall flood. The numeric representation is typically used to represent continuous data (also called a field) as topographic maps.

As a cell is limited to a unique value, the representation of different objects or phenomena that can be located at the same place must be done through different raster layers, one for each object type.

Georeferencing. Georeferencing is the process to know what part of the earth surface the raster represents. It uses a coordinate representing the raster top-left pixel, the pixel size in the X direction, the pixel size in the y-direction, and the amount (if any) by which the image is rotated.

A.2.0.3 Discussion

The description of the two data models highlights the difference in spatial information representation and attribute representation. Table A.2 summarizes the difference between the two spatial data models. The geometrical representation of vector data is very adapted to represent discrete data. Discrete data correspond to objects or features delimited by a distinct boundary or a discrete limit. They are useful to show the location, length, and perimeter of an object. A raster can also represent discrete data. However, according to the type of object (e.g., street or building), raster data can be less accurate than vector data. This accuracy depends on the object size and the raster resolution. If the raster represents "small objects" and has a low resolution, a cell would represent a large area, more significant than the object size. Therefore, the object representation in such a raster would be approximated compared to a vector data that could define precisely through coordinate the object location through a precise boundary. Nevertheless, the possibility of representing objects both through vector and raster data allows the adaptation of the representation model according to the user's computational needs. The vector data have the advantage of providing a large variety of attribute values associated with each object. The raster data have the advantage to represent continuous data, which has no boundary or has no well-defined boundary. They can represent broad area classification or describe phenomena through smooth transition values from one cell to another. The choice of the adapted data model and coordinate reference system to represent spatial information has a substantial impact on the data accuracy and, thus, on the data quality.

Characteristic	Vector data	Raster data
Feature representation	Geometry (Point, Line, Polygon)	Pixels/Cells
The most adapted representation type	Discrete data	Continuous data
Spatial information	Coordinates based on Coordinate Reference System	Georeferencing based on: - Coordinate of top left pixel - Cell size in X - Cell size in Y - Rotation(s) of image
Feature attributes	- A geometry spatially referenced - Various attribute values, whose at least an identifier and type code - Relationships/Topologies - Data quality	- A cell/area spatially referenced - From 0 explicit attribute value, (in binary representation), to 1 value, which can represent a type or another attribute value. - Data quality

Table A.2: Comparison of vector and raster data

A.3 Practice from different countries

A common point in disaster management is planning according to the three previously presented aspects: strategic, operational, and tactical. However, each country and sometimes sub-jurisdiction of a country has its strategy of disaster management. Allowing the modeling of a diversity of strategy requires to identify common points that are the base of the model, but also differences to verify that the model is flexible enough to represent diverse strategy. This subsection presents the disaster management strategy in France and Germany to identify these common points and differences.

A.3.1 Disaster management in France

This subsection presents the French model of disaster management firstly. Secondly, it offers plans related to this model and its usage.

French disaster management model The disaster management strategy in France is guided through the strategic plan ORSEC defined by the Ministry of the Interior [Castaner, 2020]. This strategic plan defines an organizational structure for disaster management that is consistent at all French administrative levels [Fortin, 2013]. Each of these levels has a command center, an administrative officer designated as director of operations, and a guideline for creating a plan adapted to the administrative level (see Table A.3). At the national level, the minister of the interior is responsible for disaster management. In case of disaster, he applies the ORSEC plan by leading from the operational center for interministerial crisis management (COGIC [2]). At the zone level, the prefect of a zone is in charge of disaster management from the zone command center (COZ[3]) and follows the region ORSEC plan. At the departmental level, the administrative responsibility is the prefect and has the function of director of rescue operations (DOS[4]). He follows the departmental ORSEC plan from the departmental operational center ((COD[5]). In the context of the ORSEC plan, each prefect draws up a specific intervention plan aimed at managing technological risks (e.g., dams, industry). This plan is based on the identification of the technological risks of the territory. It imposes on the manager at the origin of the risk to elaborate emergency plan and on the affected municipalities

[2]COGIC: Centre Opérationel de Gestion Interministériel de Crises
[3]COZ: Centre Opérationnel de Zone
[4]DOS: Directeur des Opérations de Secours
[5]COD: Centre Opérationnel Departemental

to carry out a communal safeguard plan [DDSC, 2005]. In case of a disaster at a higher level than a municipality, the municipality mayor activates the communal safeguard plan. He coordinates operations from the operational command post (PCO [6]) according to prefect directives. In case of a communal disaster that can be managed by the municipality, the role of rescue operation director is played by the mayor from the communal command post (PCC [7]), from where he also applies the communal safeguard plan. The communal command post is composed of communication responsible, an evaluation and securitizing responsible, a logistic and intervention responsible, population support responsible, a representative of the police, a representative of the fire brigade, and secretaries to document decisions and actions made.

Administrative level	Center of management	Responsible of operation	Plan
National	COGIC	Minister of the Interior	ORSEC
Zonal	COZ	Prefect of zone	Zonal ORSEC plan
Departmental	COD	DOS = Prefect	Departmental ORSEC plan
Communal	PCC	DOS = Mayor	Communal safeguard plan

Table A.3: Organizational structure of disaster management in France according to ORSEC strategy

Plans usage inside the french model The entire ORSEC plan is a strategic plan that defines the organizational structure, the plans required at different administrative levels, and case-specific plans. Implementing a plan defined in the ORSEC plan is mostly an operational plan (e.g., communal safeguard plan) designed according to an estimation of risk-adapted to the administrative level. The risk estimation determines potential disaster and their severity that can impact the population, the critical infrastructures, and the vulnerable areas. According to these risks, each operational plan contains a set of action sheets presenting the tasks, the members, the responsibility for tasks, and sometimes the required resources. The operational plans also contain a description of the command center organization and an inventory of resources available for the administrative level.

When a disaster impacts a large area, the higher administrative level affected becomes in charge of the management and implements its plan. He must coordinate

[6]PCO: Poste de Commandement Opérationel
[7]PCC: Poste de Commandement Communal

and support the lower organizational levels that also activate their plans to manage disasters locally. The role of the administrative responsibility is firstly to promote and protect the population by providing and equipping the administrative area by necessary structures (e.g., provide shelters, organize the evacuation of danger area), secondly to support and coordinate rescue staff, medical staff, and other stakeholders. Among the operational plans, action sheets are aiming at describing how to achieve the tasks directly managed by the command post. When these are well defined, that means they describe actions, people assigned to these actions, resources, and conditions of application, the action sheet becomes a tactical plan. Thus, it is possible to find tactical plans as a sub-part of operational plans.

In case of a significant number of casualties, France has an operational plan described by the NOVI plan, that makes intervenes rescue and medical staff. The rescue staff generally corresponds to fire brigades, and the medical team is mainly composed of the French ambulance and emergency service (SAMU[8]). The NOVI plan describes operations based on three main tasks: firstly, the evacuation of casualties from danger area to an advanced medical post, secondly the management of an advanced medical post, and thirdly, the evacuation to hospital. The rescue operation commander (COS[9]) manages the rescue staff and is responsible for the first task of the NOVI plan. This role is generally played by fire brigade chief. The medical staff is lead by the director of medical rescue (DSM[10]), who is responsible for the two last tasks of the NOVI plan. A medical doctor having the aptitude for managing the medical staff (for example, the director of the French ambulance and emergency service) plays this role. Actions applied during a disaster are the result of tactical planning of rescue operation commander and director of medical rescue. These tactical planning can result from their decision-making of resource management directly on the ground or from protocols of intervention that are tactical plans.

Each hospital also has specific management for an exceptional flux of casualty, which is described by an operational plan, called "white plan". A "white plan" follows the strategic plan ORSAN that corresponds to the organization of the health system's response to exceptional health situations [Touraine, 2014].

[8]SAMU: Service d'Aide Médicale d'Urgence
[9]COS: Commandant des Opérations de Secours
[10]DSM: Directeur des Secours Médicaux

A.3.2 Disaster management in Germany

This subsection presents the German model of disaster management firstly. Secondly, it offers plans related to the two central staffs of disaster management.

German disaster management model Germany is composed of federal states, which have their laws. After the terrorist attacks of 11th September 2001 in the United States, Germany decided, in agreement with its federal states, to standardize crisis management throughout Germany [Bundesministerium für Inneres, 2015]. This choice of uniformity was first reflected in the creation of the Federal Office for Civil Protection and Disaster Relief in 2004. The objective of this federal office is to support federal states in their disaster management through information and technical support services. Then, the federal states agreed on the concept of federal disaster assistance. The purpose of this agreement is to provide external support when the forces and resources available in a federal state are not sufficient. For example, a federal-state may request the police forces of other federal states, the authorities and institutions of other administrations as well as the federal police and the armed forces[11]. This collaboration defined by the "*concept for nationwide disaster relief agreed between the federal states*"[12] is tested about every two years by the Federal Office for Population Protection and Disaster Relief of Germany through the program Lükex[13]. Finally, they also agreed on a uniform disaster management system based on two staffs led by the person having overall political responsibility (e.g., district administrator or mayor). Despite this agreement, the system of disaster management of a federal state depends on its laws. Therefore, the agreed model of disaster management is only used if the law of the federal state forces it. Due to the agreement, only this German model based on two staffs is further described in this subsection. The first staff of this model is an administrative-organizational staff respecting the guideline [Arbeitskreises V, 2003]. It can be a crisis committee of administration for an administrative level from a federal-state scale to district and city level or an exceptional events staff, which is an organizational structure for disaster control for a district community level. The second one is an operative-tactical staff (operations management) respecting the guideline [AFKzV, 1999]. Each guideline (e.g., [Arbeitskreises V, 2003], [AFKzV, 1999]) and laws of federal-state defining the organizational structure for disaster management are strategic plans. Besides such

[11] under Art. 35 para. 2 para. 2 para. 2 sentence 2 of the Basic Law

[12] Annex 2 to Resolution No. 32 of the 200th Meeting of the Standing Conference of the Ministers and Senators of the Interior of the Länder of 11/12 December 2014 in Cologne

[13] Lükex: `https://cutt.ly/wtl7Fcr`

disaster management system, some infrastructures as hospitals, industrial enterprise, office, or highway have a specific emergency plan. These plans can be an operational plan when they describe the tasks and associated responsibilities according to risk, or a tactical plan when they define a protocol of actions based on situation conditions and associated with resources.

The two next sub-subsections describe the working of the two different staffs composing the disaster management model to highlight the use of operational and tactical plans inside the system.

Plans usage inside the administrative-organizational staff The administrative-organizational staff is composed of permanent members and some event-specific members. Among the permanent members, a coordinating group provides an internal service that aims at describing and documenting the situation, decisions, and actions made during a disaster. A permanent member, who has a media work, is in charge of informing the population. The other permanent members are representant of Disaster protection, Health, Environment, Social, Security, and order. There are also persons of liaison that allow the communication with external permanent members as the police and the operative-tactical staff. According to the event, members such as offices (in particular budget office) join the crisis committee. Some other external members, such as authorities (e.g., forest authorities), impacted communes and competent third parties, keep in contact with the crisis committee according to the situation of disaster. Each this member has a domain of competency. A member has to collaborate with the other members by bringing his expertise at the service of the crisis committee to solve problems resulting from the disaster situation. Each of them has to prepare a plan according to the element at risk in his domain of competency. This staff aims at securitizing and informing the population, but also at supporting the operative-tactical team in resources. They collect information about problems resulting from the disaster situation. Their role is to decide together tasks and responsible for the tasks to solve the disaster situation. The decision-making about required tasks according to the situation can follow an operational plan prepared for such a situation or can result from collaborative training to identify tasks and stakeholders necessary to have the capacity to achieve it. Then, responsible for the tasks must achieve their dedicated tasks and provides feedback about their success of implementation or further occurring problems to the administrative staff. The application of a task depends on the tactical planning elaborated by the task responsible or following a protocol of actions, which is a tactical plan prepared for such a task.

Plans usage inside the operative-tactical staff This staff shall order the deployment of civil protection organizations and ensure the coordination of all intervention measures. It is composed of a director of operations, technical advisers, and persons of liaison. This staff has to manage tasks coming from the administrative-organizational team and tasks from their rescue job. That is why this staff has internal operational management to distribute tasks among its members. This operational management has an impact on the acceptance of task responsibility coming from the administrative-organizational staff. This decision depends on the number of rescue tasks and the capacity of the staff. The management of tasks coming from the administrative-organizational staff hinges on the person of liaison. This person allows the collaboration with the administrative-organizational staff through information exchange about the situation. The person of liaison also communicates about the led operations to coordinate the required resources and tasks for supporting the population. Some tasks of the operative-tactical staff are particular to a business domain (e.g., rescue) and are automatically assigned to the corresponding member. However, some other tasks to support the population (e.g., build sandbag-based protections) are not specific to a business domain and are assigned according to the disponibility capacity of the members. This work is operational management, operational plans can manage some of its aspects, but some others must be handled according to the situation. This staff uses its business knowledge to work according to the situation. That is why operational management and tactical management work closely together. The choice of tactical management of some tasks in terms of resource impacts the possibility or not to take new task responsibilities.
Since 2012, this staff uses tactical signs to describe the situation, process information, and elaborate the tactical planning. In the sense of a common language, tactical signs are presented as a prerequisite for an efficient situation assessment. They are attributed to cross-organizational and cross-national importance. They allow describing how many men and the associated equipment or vehicles are required to a particular place to do a specific action. Each type of civil protection organization has its color of tactical symbols, aiming at representing their specific equipment or intern organizational structure. The tactical map containing the tactical signs and resulting from the decision-making of the operative-tactical staff are tactical plans.

A.3.3 Comparison of disaster management strategy

Disaster management preparation in France and Germany has several common points. Firstly, their preparation is based and adapted according to the estimated risks. Secondly, the definition of their organizational structure is for both countries, an essential point in the strategy of disaster management to facilitate the coordination between the different stakeholders. Thirdly, each of these organizational structures defines the political responsibility for the administrative area impacted by the disaster as the main responsible for disaster management. Fourthly, they both illustrate two main aspects of management: administrative management and emergency management. The administrative management, carried out by administrative staff, consists mainly of the protection, information, and aid at the population (e.g., evacuation, hosting, water, and food supply). The emergency management carried out by the emergency team, the most often lead by the fire brigade, consists mainly of rescuing and medical emergency management. Finally, each of them uses strategic, operational, and tactical plans.

Comparing both preparations also allows the identification of differences between the French and German models. The main difference between France and Germany appears in the organizational structure. France has a single-line system [Schulte-Zurhausen, 2005] whereas Germany has a member system [Schulte-Zurhausen, 2005]. This difference has a substantial impact on the communication and information exchange. The member system facilitates the collaboration work since all members have an overview of the situation and tasks led by other members. They can thus, identify information relevant for their business domain and communicate it to their organization. On the contrary, when information goes only to one person, the distribution of information is more global, and some useful details are lost in the system. However, the structure of the plan ORSEC in France enforces the preparation of different potential stakeholders like managers of an element at risk thanks to the plans requirements definition according to risks. This elaboration must be done in collaboration with the municipality to facilitate a collaborative preparation. The strategy ORSEC enforces the gathering of plans and, thus, the collaborative work. In Germany, tasks are more distributed among the different members. Each member has a role and has to be able to achieve it. Nevertheless, plans prepared by each member are not wholly shared. Therefore, each member has an overview of tasks and their results but a limited view of the task processing of other members.

Table A.4 shows the comparison of disaster management strategy between France

Type of plan		France	Germany
Strategic plan	Name	Plan ORSEC, Plan ORSAN	Consortium, Laws of federal states
	Content	Single-line system	Member system
Operational plan	Examples of general plan	Communal safeguard plan of Montbard, Departmental ORSEC plan of Côte d'Or, Zonal ORSEC plan of Bourgogne Franche-Comté	Alarm and operational plan: - for flood - for evacuation - for highway - etc
	Plan for significant number of casualty	NOVI plan, White plan	Hospital alarm and emergency plan
Tactical plan		Action sheets in communal safeguard plan	Tactical map

Table A.4: Comparison between disaster management in France and Germany according to strategic, operational and tactical plans

and Germany. On the one hand, this comparison allows the identification of the fundamental base of disaster management preparedness through their common points. The common goal of this preparedness is to facilitate the collaboration between the different stakeholders — this preparation results in plans with different levels or types of details. The common points highlight the essential role of the geospatial dimension through risks and administrative areas. It also shows some typical roles and tasks related to administrative management and emergency management.
On the other hand, the identification of differences allows the specification of the different models. The two models have a different organizational structure implying a contrast of roles, various plans, and different organizations as stakeholders (e.g., SAMU vs. JUH[14]).

A.3.4 Discussion

This section provides an overview of disaster management and highlights the essential role of the preparedness on disaster management success. The preparedness and, more precisely, the elaboration of plans is fundamental to obtain efficient management of a crisis. Around the world, three aspects of planning often appear in a

[14] JUH: abbreviation of "Johanniter-Unfall-Hilfe", the German organization of the St. John Ambulance

quite similar way: strategic, operational, and tactical elements. The strategic aspect is the global model of disaster management of a jurisdiction. The operational aspect is the implementation of the jurisdiction model specifically for an administrative area. Finally, the tactical aspect corresponds to the resource management in the activation frame of some parts of the implemented model according to a specific situation. The description of these three aspects combined with the comparison of disaster management strategy in France and Germany provide a first view of the essential concept for describing disaster management planning. Table A.5 provides this first overview of the model concepts according to the strategic, operational, and tactical plans obtained through the descriptions inside this section.

Aspect	Concepts and some relationships
Strategic	Administrative area, Plan, Organization, Role and Task
Operational	Risk, Event, Person and Organizations associated with a Role, Service and task associated with a Role, required Resources for a Task
Tactical	Situation conditions, Actions associated with a task or a service, Human resources, Equipment, Vehicles

Table A.5: Overview of disaster preparedness concepts according to the strategic, operational and tactical aspects

First of all, at the strategic level, the preparedness requires the description of an organizational structure. This structure is dependent on an administrative level that provides a spatial dimension of disaster preparedness. This structure is described in a strategic plan that defines the stakeholders through organizations, attributes them roles and tasks. The strategic plans also defined requirements of operational plans according to certain conditions. Then, operational plans define actions, a person in charge, and required resources according to risks and resources specific to the administrative area. Finally, the tactical plan corresponds to the management of activated operational plans according to the events occurring and situation conditions. It aims at managing resources (e.g., humans, equipment, vehicles). Tactical plans result from the decision made during a disaster to solve the situation. Tactical plans are situation-specific. Therefore, it is not possible to plan all situation configurations and, thus, not possible to define all tactical plans. However, it is possible to train for tactical planning according to a scenario to prepare and improve the decision-making.

Such an aspect of scenario diversity for assessing plans appears not only at the tactical level to manage resources but also at the operational level to determine the stakeholders intervening in the disaster response. The preparedness cycle contains a sub-step of assessment and improvement (see Figure 3.2). Therefore, the assessment of plans implies two main aspects. Firstly, it requires to verify the coherency of tactical plans according to operational plans, and operational plans according to strategic plans.

Let us take the example of a French municipality exposed to a technological risk; the system must be able to verify if this municipality has a safeguard communal plan according to the ORSEC strategy. Secondly, it requires to assess the effectiveness of the overall plans. This type of assessment needs to simulate the application of plans according to diverse scenarios to quantify the effectiveness resulting from the preparation. According to [Federal Emergency Management Agency, 2010], the assessment and improvement of plans are based on real exercise (cf. Figure 3.2) to simulate the application of plans. However, in reality, there is too much possible combination between the different stakeholders and the plans, to be assessed through a real-exercise. A proper assessment of plans through real exercise would be too much cost and too much time to be available. The evaluation of plans is the main limit of the preparedness that conducts to the problems of collaboration highlighted in subsection 3.1.1.5. Even if plans are prepared, the lack of assessment does not allow the guarantee of their effectiveness. The main problem of real exercises is to limit the number of scenarios. It thus reduces the assessment to a few exercises and does not allow testing the overall plans. That is why virtual simulations are commonly used to assess plans of disaster management, in addition to the exercises that allow stakeholder training. The use of simulation techniques provide virtual exercise with a lower cost than a real one and not limited in the testing scenario. The next subsection presents the simulation domain.

The disaster management is about efficiency (achieving tasks optimally) but mainly effectiveness (delivering the right task at a good time). Effectiveness plays a crucial role in disaster management. The individual preparation allows the obtention of efficiency in task doing, but not efficacy. The effectiveness depends on the collaborative preparation by taking into account the task dependencies. If a task is optimally achieved, but do not respect the interdependency or harms other tasks performed by different stakeholders, then the achievement of this task is not sufficient. Thus, the increase of effectiveness is as, perhaps more, significant as, than the rise of the efficiency for disaster management.

B Method's implementation appendix

B.1 Processes intervening in the approach of extraction and integration of knowledge from data (section 6.1.2)

B.1.1 Geometry processing

The geometry processing has three aspects to explain: its modeling, geometry matching, and the geometry comparison.

Geometry modeling: Processed geospatial data are represented through the GeoSPARQL vocabulary [Battle and Kolas, 2011]. It allows the extension of the vocabulary using the interpreted data sets. GeoSPARQL defines a spatial object by geometry and a linked feature. Representing a geometry in GeoSPARQL requires to retrieve and identify the geometry type of each entity of the geospatial data set and automatically detect the correct representation of the feature linked to the geometry.

Geometry matching: The geospatial Semantic Web consists of geospatial ontologies such as GeoNames[1] and LinkedGeoData Ontology [Auer et al., 2009]. These ontologies gather a great number of classified geometries to describe the object corresponding to said geometry. The first step of the integration approach is to use a small enough buffer around the geometry to identify a concept. Buffer means either an encompassing rectangle around a point geometry or an encompassing rectangle around the centroid of a non-point geometry. If the last iteration does not provide results, the buffer is increased dynamically. The process of one iteration is illustrated in Figure B.1.

[1]GeoNames Ontology: `http://www.geonames.org/ontology/documentation.html`,

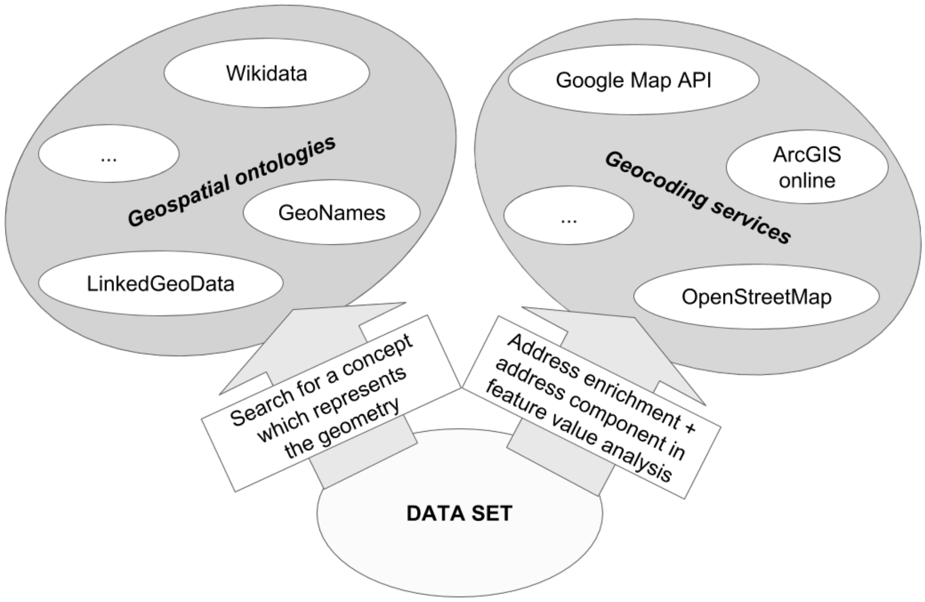


Figure B.1: Geometry and data set specification [Prudhomme et al., 2017a]

Geometry comparison: In many cases, not only the matching of a class is important, the merging of the attributes of two particular geometries and the relation of a particular geometry to encompass it is also of interest. Let us take the example of a hospital complex; such a complex is expected as composed of various buildings, some of which may share the concept of a hospital. However, some of which (e.g., a hospital chapel building) may not include a hospital annotation in the respective ontology. To be able to determine the exact geometry concerned, the following geometry matching and spatial fusion techniques are used from the geospatial world:

- Similarity Metrics: Haussdorff Distance [Huttenlocher et al., 1992], Fréchet Distance [ALT and GODAU, 1995], Shape Similarity [Veltkamp, 2001], Overlapping Degree [Berretti et al., 2000]

- Geometrical Features: Diameter, Length, NumberOfPoints

Similarly to a comparison of label values in the Semantic Web for concept matching, geometry matching algorithms are applied for data sets providing enough geometrical information for verification (Polygons or LineStrings). This approach is extendable to encompass possible further metrics, many of which are introduced

visited on 2020-09-22

in [Veltkamp, 2001]. The more metrics are applied, the more information our algorithm can evaluate as a basis for geometrical similarity assessment.

B.1.2 Feature value analysis

After the geometry processing, the following step is a feature value analysis. Its first step is to identify the information that frequently appears in the data sets. In this process, empty and NULL values are ignored. The types of information and their detection are described below and illustrated in Figure B.2.

- **Address components:** The specificity of geodata sets is that they contain a geometry for each spatial object. The usage of the spatial object geometry with a geocoding service (in our case, Google Maps API[2]), allows address enrichment, explained in [Prudhomme et al., 2017a]. The information retrieved is compared with the different value of the cell to determine which column contains information concerning the geographic address of the object.

- **ID:** The process of an eventual ID discovery corresponds to an analysis of values. It aims at identifying a column, which fulfills the following constraints: the value has to be an integer and has to be unique. If we discover UUIDs, for example, they will be categorized using appropriate regular expressions. IDs could be used as individual descriptors in a later process.

- **Unit:** A double generally represents a quantification, which is why an analysis of all columns determines that a column could represent a quantity with a unit if all values are Double or Integer. Something that is usually measured in any unit (e.g., 2.5°C) or is a description of an amount (2.5 apples). If we can identify the column type from its descriptor, then we may be able to use it to conclude the unit associated with this type. Otherwise, it will be associated to Unitless. Work on integrating, e.g., DBpedia [Auer et al., 2007] with unit ontologies has been done by [Rijgersberg et al., 2013] and is extended manually by the work of SemGIS project for most common units.

- **Regular expression:** A set of regular expressions has been defined for: A date, a phone number, email address, a website URL, and a UUID. This set of regular expressions is then applied to all strings to check whether the string matches one of those regular expressions. The elements identified as date are stored thanks to a data property with the name of the column and the type

[2]Google Maps API: `https://developers.google.com/maps/documentation`

xsd:date. Information corresponding to a phone number, email address, and a website is stored using FOAF ontology[Kalemi and Martiri, 2011] properties foaf:phone, foaf:mbox, foaf:homepage. The UUID is stored as a data property.

- **Remaining String:** Natural language processing in the form of named entity recognition, and POSTagging is applied to all strings which have not yet been identified (using the Stanford NLP Toolkit [Manning et al., 2015]). For the moment, this natural language processing is specific to German and English and may be extended to further languages in the future. It is aiming to determine whether the string is an adjective or a noun. The values of the column, containing a majority of adjectives, will become an instance of the concept linking to the general concept with an object property. When a column contains a set of nouns that occur frequently, we assume the column describes a type of the general object, as stated in [Prudhomme et al., 2017a]. The value of this column is processed to identify a set of nouns without redundancy. Then, the nouns, which composed this set, are added as a subclass of the general concept, which represents the file. When all values have been analyzed, the process of Feature Descriptor Analysis (cf. section B.1.3) begins. It is applied to all column names that have not yet been considered as an adjective column or a subclass by the value analysis process.

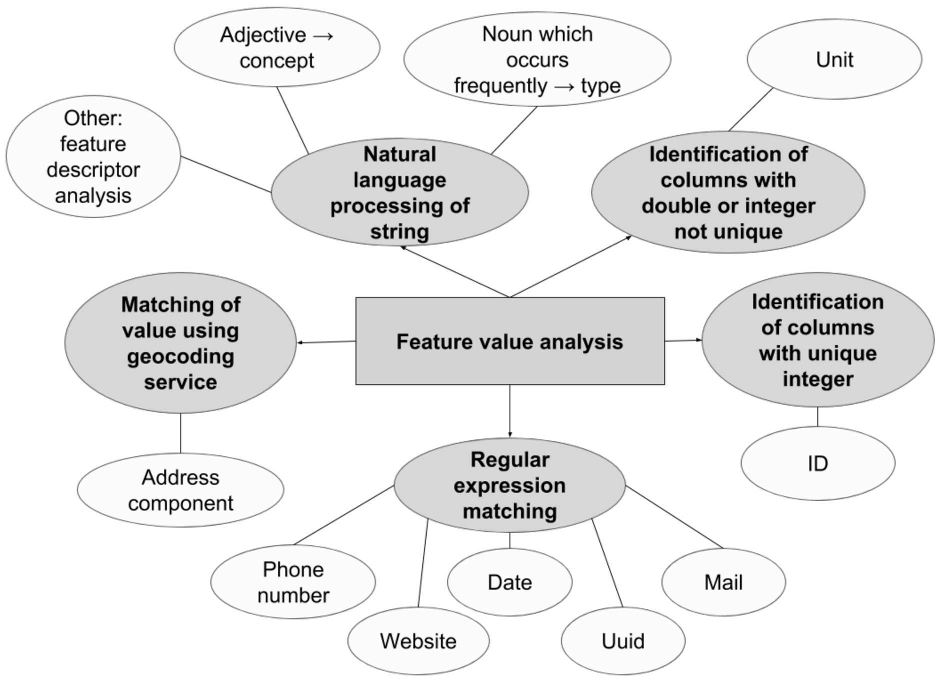


Figure B.2: Identification by value analysis [Prudhomme et al., 2017a]

The second step consists of the recognition of a named entity. When doing string value analysis, it is essential to separate named entities from nouns, because their representations in ontologies usually vary from each other. In a geospatial context, geographic names are likely found in a data set to link them to other relevant geospatial resources. One example could be an administrative district or a specific administrative building like the town hall or the parliament using its name [Buscaldi and Rosso, 2008].

B.1.3 Feature description analysis

The feature description analysis (Figure B.3) gives valuable information about properties and classes in ontologies that represent the column's content. However, column names are represented in natural language and with a limited context to parse from, which can limit disambiguation methodologies if needed. Besides, before an analysis of the feature descriptor can be conducted, the following preprocessing steps must be conducted:

- Detection of the language being used in the column's name using, for example, the Google Translate API
- Recognition of standard abbreviations and replacement of those with their long-form using abbreviation lists for the particular language

The process of analysis of column names begins by a concept matching with Wikidata concepts, first using its URI and, if this fails, using a label matching approach. If there is no concept after these two steps, the given column name is translated into English and try the steps as mentioned earlier again. Using an English translation is not always possible, as the interpretation of the full term does not necessarily represent a word that can be found in a dictionary or ontology. Compound words often needed to be split and investigated separately. In that regard, the parts of compound nouns are analyzed from their ending to their beginning to resolve possible concepts from those noun parts (c.f. German and English examples in Code B.1).

```
Bauarbeiter -> Arbeiter
primary school -> school
```

Code B.1: Splitting of compound nouns

If no concept is found for the column's name using all of the methods mentioned above, the column is declared unresolvable. If many results are obtained for the

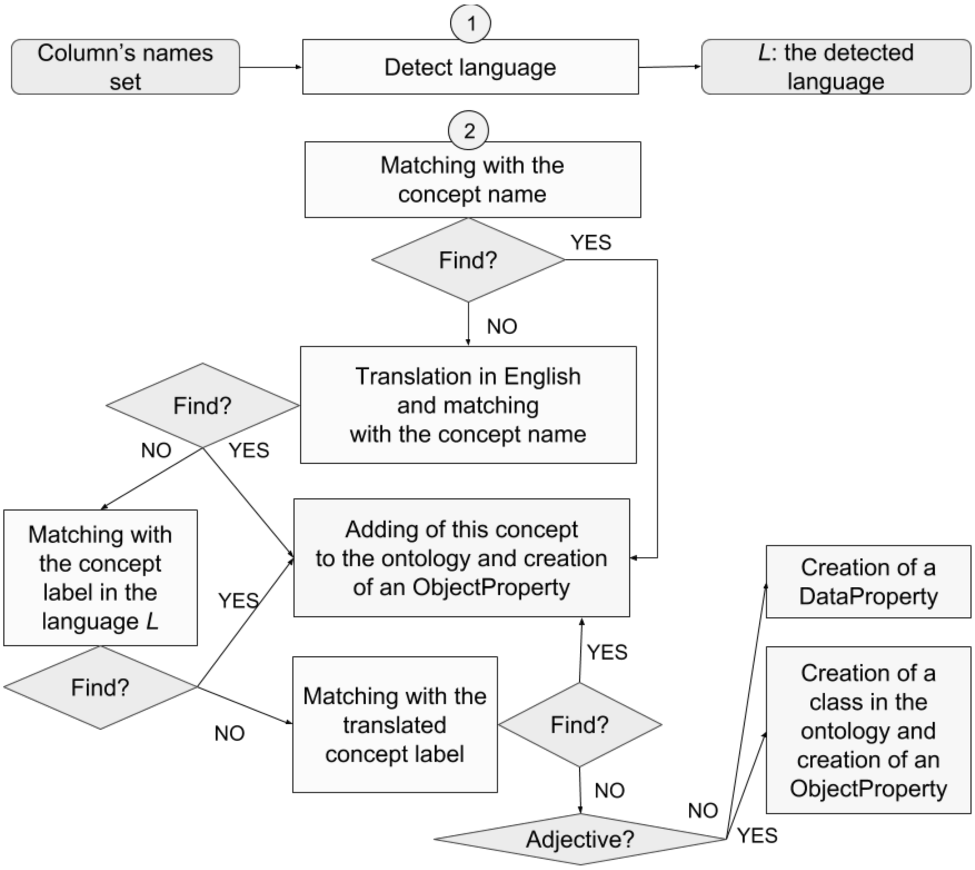


Figure B.3: Process of linking with Semantic Web resource [Prudhomme et al., 2017a]

respective column, the results are ranked using the Levenshtein Distance to find out the concept name, which comes closest to the column's title. This concept will be used to describe the column in the local ontology.

B.2 Example of a SHACL (Shapes Constraint Language) rule to explain its structure

As explained in section 4.3.1 of chapter 4, SHACL is a language for validating RDF graphs against a set of conditions. These conditions are provided as shapes and other constructs expressed in the form of an RDF graph. These shapes and constructs allow graph validation. However, SHACL is not limited to graph validation. It also allows rule-based reasoning thanks to advanced features [3]. In this advanced features, SHACL allows the definition of two types of rules: Triple rules (*sh:TripleRule*) and SPARQL rules (*sh:SPARQLRule*). The proposed approach uses

[3]SHACL advanced features: `https://www.w3.org/TR/shacl-af/`

the SHACL-SPARQL rules (*sh:SPARQLRule*) to benefit of SPARQL advantages. Figure B.4 presents the definition of the *sh:SPARQLRule* corresponding to the part presented in Code 6.16 of chapter 6.

```
@prefix rdf: < http :// www . w3 . org /1999/02/22 - rdf - syntax - ns# > .
@prefix rdfs : < http :// www . w3 . org /2000/01/ rdf - schema # > .              1
@prefix sh : < http :// www . w3 . org / ns / shacl # > .
@prefix semMAS : < http://www.i3mainz.de/SemGIS/semMAS# > .

semMAS:Species

      a rdfs:Class , sh:NodeShape ;                                            2
      sh:targetClass semMAS:Species ;

      sh:rule [

            a sh : SPARQLRule ;
            sh : order 1;
            rdfs : label " Defining skills of an agent species " ;            3
            sh : prefixes semMAS : ;
            sh : construct " " "

            PREFIX semMAS : <http://www.i3mainz.de/SemGIS/semMAS#>
            CONSTRUCT {
             $this  semMAS:hasSkill ?s .
            }                                                                  4
            WHERE {
            $this  semMAS:hasReflex  ?r .
            ?r semMAS:usesSkill ?s.
            }

            " " "
            ].
```

Figure B.4: Example of SHACL rule corresponding to the part presented in Code 6.16

All SHACL specifications begin with the definition of prefixes, which corresponds to the block number 1 in Figure B.4. Then, they describe each concept on which define constraints and rules. In Figure B.4, the *sh:SPARQLRule* is defined on the concept *semMAS:Species*, which is defined in block number 2. Block number 3 of this figure defines the rule presented in Code 6.16 of chapter 6. The rule is defined by:

1. a type, which is *sh:SPARQLRule* for all rules defined in this thesis,
2. a priority order, which is a number defined through the property *sh:order* (1 is the highest priority),
3. a label, defined through the property *rdfs:label*,

4. the prefixes, used in the *sh:construct* and introduced through the property *sh:prefixes*,

5. the SPARQL Construct introduced through the property *sh:construct*.

The SPARQL Construct is based on *$type* that represents all individuals of the concept on which the rule is defined. The SPARQL Construct defined in this exemple is thus equivalent to the following rule:

$$rdf:type(?this, semMAS:Species) \wedge semMAS:hasReflex(?this, ?r)$$

$$\wedge semMAS:usesSkill(?r, ?s) \Rightarrow semMAS:hasSkill(?this, ?s)$$

In this manuscript, only the SPARQL Construct's content of the used rules is presented to describe the SHACL rule. The type of the individuals on which the rule is applied (i.e., the concept addressed by the rule) has been added to the SPARQL Construct description to understand the short description better.

B.3 Prefixes

The prefixes used throughout this thesis in the different semantic definitions, rules, and queries are defined below in Code B.2.

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX ogc: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX wikidata: <http://www.wikidata.org/entity/>
PREFIX wkd: <http://www.wikidata.org/entity/>
PREFIX semDM: <http://www.i3mainz.de/SemGIS/semDM#>
PREFIX semMAS: <http://www.i3mainz.de/SemGIS/semMAS#>
PREFIX semTransform: <http://www.i3mainz.de/SemGIS/function/semTransform
    />
PREFIX semStatistics: <http://www.i3mainz.de/SemGIS/function/
    semStatistics/>
```

Code B.2: Listing of prefixes used in the thesis