UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Extraction and integration of constraints in multiple data sources using ontologies and knowledge graphs

THESIS IN COTUTELLE

publicly presented and defended on the 25/03/2025

for obtaining the

PhD of the University of Lorraine
(Automatic control, Signal and Image Processing, and Computer Engineering)
PhD of the National University of Quilmes
(Science and Technology)

by

Mariano Julián Ferreirone

Composition of the jury

President:	Prof. William DERIGENT	University of Lorraine
Reviewers:	Prof. Regina MOTZ	University of the Uruguay Republic
	MdC HDR. Thierry COUDERT	University of Toulouse
Examiners :	MdC HDR. Gabriela AREVALO	National University of Quilmes
	Prof. Pablo MARTINEZ LOPEZ	National University of Quilmes
Invitees:	M. Jéremie NOYREY	ESF
	M. Cedric MARCONE	Valraiso
	M. Laurent TESTE	ESF
	M. Yoann COUBLE	Valraiso
Supervisors:	MdC HDR. Mario LEZOCHE	(supervisor) University of Lorraine
	Prof. Hervé PANETTO	(co-supervisor) University of Lorraine
	Prof. Diego TORRES	(co-supervisor) National University of Quilmes

IAEM Lorraine Doctoral School
Centre de Recherche en Automatique de Nancy (CRAN) — CNRS UMR 7039
Centro LIFIA, Facultad de Informática, Universidad Nacional de La Plata

Mis en page avec la classe thesul.

Acknowledgments

It's almost impossible to understand how difficult and challenging a Ph.D. is until you experience it. Going through this process, being far from home, learning a new language and the never stopping progressing of life, have created about this, the biggest challenge of my life. It's only with dedication, braveness and love that I got to the point in which I'm today. Love represented in many ways that came from people to whom I want to thank.

First, I want to thank the jury. I see their love for the research, translated in their availability and willing for reading my manuscript and giving a very valuable feedback that will guide the horizon of my next challenges.

I want to thank my supervisors, who were a fundamental part of this work. Their love not only for the research, but also for the guidance of new researchers, it's shown in their daily work. Specifically, I want to thank Hervé for the work that he did to make the existence of this Ph.D. possible. Thanks to Mario, for welcoming me in France, for helping me in so many ways, for trusting in me, for the understanding, for the long research (and life) discussions, and overall, for his friendship. Thanks to Diego, for always challenging my ideas, for pushing me to be the best Ph.D. that I could, for the comprehension and for the Argentinian fellowship.

I would like to thank to all the people that I met of the scientific and administrative staff of the University of Lorraine, National University of Quilmes, and National University of La Plata, for welcoming me in the best way, for giving me the possibility to have very valuable exchanges and for helping me to feel supported. I want to thank to ESF, for supporting the research and accepting me as a Ph.D. candidate. I want to thank to Valraiso, and most of all to Yoann Couble, for welcoming me, for being interested in my work and for allowing to me to have my first professional introduction into the French enterprise environment. I want to thank to the Polytechnic University of Madrid, to the Ontology Engineering Group and to the professor Raúl García Castro, for welcoming me in Madrid during three months, paving the way for a very interesting research and cultural exchange.

I want to thank to all the Ph.D. students, postdocs and master students that I met in these three years. For all the moments that we shared, for the dinners, the games and for the memes (laughing everything is easier). I want to thank particularly to Chiara and Nico, for always believing in me, for the never ending talks, for the trips, for your true friendship, for the laughs and for the tears, for being part of my European family, for being an invaluable support. I want to thank also to all the amazing friends that I've made in Nancy, for the trips, the beers and the unforgettable moments.

I want to thank to all my friends in Argentina, who never hesitated to support me in the hardest moments, for reminding me that they are always there. Particularly, to my friends from the school, Sergio, Fernando and Gabriel. I want to thank also to my friends from my childhood neighbourhood and to my friends from "Gesell". I'll be always thankful to all of you, for being on my side when I had to go to Argentina in the most difficult moments, for always waiting my following visit, for making me laugh when everything was dark.

I want to thank my mother Claudia (my biggest inspiration), my brother Tomas and my sisters Natalia and Daniela, to my little nieces Roma and Julieta. For your support and your comprehension, for being always there. For understanding my decisions even in the hardest moments, for all the welcome and farewell hugs. For all the love and for taking care of dad. I'll always be grateful to you. I want also to thank to my uncle Marcelo, which has also taken care of my dad, to my grandmother Nelly for always showing me the importance of the education. To the memory of my grandfather Nestor, who started planting from very early the curiosity in me and for teaching me that having strong ideals is a way of living. To my grandmother Elena who

is always with us. To my family in-law, Ana, Dani, Tito, Yaya and Facu, for always supporting our projects, for always being there, for the love and respect with which you accompanied me in the tough moments. I want to make a special mention to Tito and to his memory. He became my grandfather in very little time and showed me love each time that I saw him.

I want to thank to my wife Micaela, for never doubting a second of my capabilities, for always being there, for sharing my crazy dreams, for being my pillar in the most difficult moments, for showing me love in all its possible forms, for helping me day by day to become a better person and for sharing her life with me. Only the two of us know all the things that we've gone through during this time. I've no doubt that this Ph.D. is also in part yours. Thank you for everything.

Finally, I want to thank to my father, Sergio. This manuscript is completely dedicated to him, to his memory. I'll never forget your braveness and the love with which you fought. I'll never forget the hugs and the tears each time that I flew far away from home, even when everything was good. I'll never forget that you never ever gave me a minimal sign to abandon everything for you, and that was my fuel to fight with all my strength to get to the end of this road. I will never forget the last hug. This is for you, dad, I don't have any doubt that you are here with me, in this new milestone.

To all of you, my gratitude is forever.

With love,
Mariano

"Knowledge once gained casts a light beyond its own immediate boundaries." - by John Tyndall, in Fragments of Science for Unscientific People: A Series of Detached Essays, Lectures, and Reviews.

Abstract

This thesis explores the introduction of Shapes Constraint Language graphs in semantic environments which present heterogeneous sources and different context requirements. This research presents a proposal for the enrichment of Semantic Web based systems, providing benefits for several domains such as the Industry 4.0. The thesis starts with a wide review of the current works related to the validation of semantic constraints on knowledge representation models. Based on a systematic literature review, a definition of a taxonomy which describes the related types of works is proposed. The open challenges related to the creation of shape graphs and their inclusion in existing semantic environments are highlighted. The needs for a shape graph representation which is able to attend different contexts and for the integration of shape graphs stand out. Based on the Shapes Constraint Language standards, a semantic restriction model which represents groups of shapes which share target and could hold inter-shape conflicts is presented. A pre-validation configuration process to activate the model's sub-graph that best fits the current context is described. Moreover, an approach for the integration of these graphs which belongs to a common environment is proposed. The integration procedure resolves the constraint conflicts with the specialization of shapes. A practical use case based on the French Ski School demonstrates the usability of the proposed contributions. Evaluations of the correctness and consistency of the generated shape graph are carried out. The implemented procedures' performance is also evaluated. The thesis concludes by summarizing the contributions and suggesting future research directions to further improve the integration and representation of Shapes Constraint Language graphs.

Keywords: Knowledge Representation, Semantic Constraints, RDF Validation, Shape graph, Ontology, Knowledge Graph

Résumé

Cette thèse explore l'introduction des graphes de formes SHACL dans des environnements sémantiques présentant des sources hétérogènes et des exigences contextuelles variées. Cette recherche propose une approche pour l'enrichissement des systèmes basés sur le Web sémantique, apportant des bénéfices à plusieurs domaines, notamment l'Industrie 4.0. La thèse commence par une large revue des travaux actuels liés à la validation des contraintes sémantiques sur les modèles de représentation des connaissances. Sur la base d'une revue systématique de la littérature, une définition d'une taxonomie décrivant les types de travaux associés est proposée. Les défis ouverts liés à la création de graphes de formes et à leur inclusion dans les environnements sémantiques existants sont mis en évidence. Le besoin d'une représentation des graphes de formes capable de s'adapter à différents contextes ainsi que l'intégration de ces graphes sont particulièrement soulignés. En s'appuyant sur les standards SHACL, un modèle de restriction sémantique représentant des groupes de formes partageant une même cible et pouvant contenir des conflits inter-formes est présenté. Un processus de configuration de pré-validation permettant d'activer le sous-graphe du modèle le mieux adapté au contexte actuel est décrit. De plus, une approche d'intégration des graphes de formes SHACL appartenant à un même environnement est proposée. L'intégration proposée résout les conflits de contraintes grâce à la spécialisation des

formes. Un cas d'utilisation pratique, basé sur l'École de Ski Française, démontre l'utilisabilité des contributions proposées. Des évaluations de l'exactitude et de la cohérence du graphe de formes généré sont réalisées. Les performances des procédures mises en œuvre sont également évaluées. La thèse se conclut par un résumé des contributions et une proposition de pistes de recherche futures visant à améliorer davantage l'intégration et la représentation des graphes de formes SHACL.

Mots-clés: Représentation des connaissances, Contraintes sémantiques, Validation RDF, Graphe de formes, Ontologie, Graphe de connaissances

Resumen

Esta tesis explora la introducción de los Grafos de Shapes SHACL en entornos semánticos que presentan fuentes heterogéneas y diversos requisitos contextuales. Esta investigación propone un enfoque para el enriquecimiento de los sistemas basados en la Web Semántica, aportando beneficios a varios dominios, incluyendo la Industria 4.0. La tesis comienza con una amplia revisión de los trabajos actuales relacionados con la validación de restricciones semánticas en los modelos de representación del conocimiento. A partir de una revisión sistemática de la literatura, se propone una definición de una taxonomía que describe los tipos de trabajos asociados. Se destacan los desafíos abiertos relacionados con la creación de grafos de Shapes y su inclusión en los entornos semánticos existentes. Se subraya especialmente la necesidad de una representación de los grafos de Shapes capaz de adaptarse a diferentes contextos, así como también la necesidad de poder realizar la integración de multiples grafos de este tipo. Basándose en los estándares SHACL, se presenta un modelo de restricción semántica que representa grupos de restricciones que comparten un mismo objetivo y que pueden contener conflictos entre si. Se describe un proceso de configuración de pre-validación para activar el subgrafo del modelo que mejor se adapte al contexto actual. Además, se propone un enfoque para la integración de los grafos de Shapes SHACL que pertenecen a un mismo entorno. La integración propuesta resuelve los conflictos de restricciones mediante la especialización de los conjuntos de restricciones. Un caso de uso práctico, basado en la Escuela de Esquí Francesa, demuestra la usabilidad de las contribuciones propuestas. Se realizan evaluaciones de la exactitud y la coherencia del grafo de Shapes generado. También se evalúan el rendimiento de los procedimientos implementados. La tesis concluye con un resumen de las contribuciones y una propuesta de direcciones de investigación futuras para mejorar aún más la integración y representación de los grafos de Shapes SHACL.

Palabras clave: Representación del conocimiento, Restricciones semánticas, Validación RDF, Grafo de Shapes, Ontología, Grafo de conocimientos

Summary

Chapter 1 Introduction **1**
1.1 Motivation . . . 1
1.1.1 A deep dive into the specific field . . . 3
1.1.2 SHACL shapes graphs integration . . . 4
1.1.3 SHACL shape graphs for multiple contexts . . . 5
1.2 Scope and objectives of the thesis . . . 6
1.3 Contributions summary . . . 7
1.4 Outline of the thesis . . . 7

Chapter 2 Background **9**
2.1 Introduction . . . 9
2.2 Knowledge representation . . . 9
2.3 Resource Description Framework (RDF) . . . 10
2.4 Ontology . . . 11
2.5 Knowledge graph . . . 12
2.6 SPARQL Protocol and RDF Query Language . . . 13
2.7 Semantic constraints . . . 13
2.8 Semantic constraint languages . . . 14
2.9 Relevant SHACL elements . . . 15
2.10 Industry 4.0 and the Semantic Web . . . 17

Chapter 3 State of the art **19**
3.1 Introduction . . . 20
3.2 Related reviews . . . 20
3.3 Method . . . 21
3.3.1 Research question . . . 22
3.3.2 Keywords and search string definition . . . 23
3.3.3 Database selection . . . 24
3.3.4 Study selection and quality assessment . . . 25

3.3.5 Data extraction . . . 27
3.3.6 Analysis and classification . . . 27
3.4 Results of the review . . . 28
3.4.1 Publications year and venue (RQ1 and RQ2) . . . 28
3.4.2 Application domains (RQ3) . . . 29
3.4.3 Shared repository (RQ4) . . . 31
3.4.4 Knowledge representation models involved (RQ5) . . . 33
3.4.5 Semantic constraints and their representation (RQ7 and RQ8) . . . 34
3.4.6 Evaluated characteristics (RQ9) . . . 35
3.5 Type of contribution taxonomy (RQ6) . . . 36
3.5.1 Constraint language analysis and review . . . 38
3.5.2 Constraint language semantic and syntactic definition . . . 38
3.5.3 Constraint language implementation . . . 38
3.5.4 Constraint language applied cases . . . 41
3.5.5 Constraint language recursion . . . 41
3.5.6 Constraint language satisfiability and containment . . . 41
3.5.7 Constraint language further characteristics . . . 42
3.5.8 Mapping techniques . . . 42
3.5.9 SPARQL constraint implementation . . . 43
3.5.10 Definition of specific types of constraints . . . 44
3.5.11 Schema extraction . . . 44
3.5.12 Visual constraints creation and validation . . . 45
3.5.13 Validation reports . . . 45
3.5.14 Error treatment . . . 46
3.5.15 Inference in validation . . . 46
3.5.16 Validation of semantic constraints for query optimization . . . 47
3.5.17 Validation algorithms . . . 48
3.5.18 Other approaches . . . 48
3.6 Discussion . . . 49
3.6.1 Integrated results . . . 49
3.6.2 Research gaps . . . 52
3.7 Shape graphs integration related work . . . 53
3.7.1 Integration of constraints from heterogeneous sources . . . 53
3.7.2 Integration of SHACL shape graphs . . . 54

Chapter 4 Integrated Restrictions Model **63**
4.1 Introduction . . . 63
4.1.1 Motivation . . . 63
4.1.2 Preliminaries . . . 65
4.2 Representation approach . . . 68
4.2.1 Design of the Integrated Restriction Model . . . 68
4.2.2 Pre-validation configuration . . . 73
4.2.3 Source provenance representation . . . 78
4.3 Conclusion . . . 81

Chapter 5 Integration of SHACL shape graphs with shapes specialization **83**
5.1 Introduction . . . 83
5.1.1 Motivation . . . 83
5.1.2 Preliminaries . . . 85
5.2 Integration procedure . . . 86
5.2.1 Conflict Resolution and Integration Process . . . 88
5.3 Conclusion . . . 92

Chapter 6 Experiments and evaluations **95**
6.1 Introduction . . . 95
6.2 Ski Teaching school use case . . . 95
6.2.1 Integration of the resorts' characteristics . . . 96
6.2.2 Validation of the general system data for multiple contexts . . . 98
6.3 Implementation . . . 98
6.3.1 ESF knowledge representation models . . . 99
6.3.2 IRM pre-validation configuration . . . 100
6.3.3 SHACL shape graphs integration procedure . . . 101
6.4 Experiments and evaluation . . . 102
6.4.1 Integration of two SHACL shape graphs corresponding to two different resorts . . . 102
6.4.2 Activation of the generated ESF IRM related to the weather context . . . 104
6.5 Conclusion . . . 107

Chapter 7 Conclusions and future work **109**
7.1 Summary . . . 109
7.2 Perspectives and future directions . . . 110

Résumé étendu en français **113**

Resumen extendido en español **119**

Appendix A SLR final queries **125**

Appendix B Complete constraint conflict resolution by shapes specialization scenarios **129**

Appendix C List of Abbreviations **135**

Bibliography **137**

List of Figures

1.1 A common scenario which relates heterogeneous sources with Semantic Web models. Some of these sources represent data, others knowledge, schemas or even scripts. Through the design of mappings, these sources could be used for the construction of a related RDF Knowledge graph. 2
1.2 This figure shows first the relationship between a shape graph and a data graph in SHACL. Then, it shows a simple shape definition which restricts the persons to have one passport as maximum. Finally, two data graphs examples are shown, where the first one doesn't accomplish to the shape, meanwhile the second one does. 4
1.3 A scenario which shows the need for the integration of multiple shape graphs. . . 5

2.1 A graphical representation of a simple RDF graph example. The nodes in orange represent IRIs, the green one represents a literal and the light blue a blank node. This graph represents knowledge related to ski lessons of type groupLesson. In addition, the graph presents knowledge related to a student which participated in an instance of the aforementioned lesson. 11

3.1 The Systematic Literature Review Steps 22
3.2 Number of articles included and excluded in each step of the review. 26
3.3 Papers by year 30
3.4 Papers by venue 30
3.5 Papers by domain 31
3.6 Tweaked Semantic Web Technology Layer Cake published by Idehen ([89]) 32
3.7 Papers by domain - Semantic web exploited 32
3.8 Papers by repository shared 32
3.9 Papers by validated model 33
3.10 Constraint languages used 34
3.11 Papers with specified constraints 35
3.12 Specified constraints 36
3.13 Evaluated characteristics 37
3.14 Taxonomy of research trends related to validation of formal semantic constraints 39
3.15 Distribution of the selected papers by their type of contribution. CL stands for "Constraint Language". 40

4.1 Description of a set of Source Shape Graphs (SSG) which are a set of Shape Graphs which are desired to be integrated in the IRM. Furthermore, the structure of a shape graph is detailed, highlighting the importance of the target Ti definitions. . 68

4.3 This figure represents two shapes which are part of the same group (GroupLesson_has_student). The hierarchical relationship if represented with the arrow which states an extension. . . . 73
4.2 Brief graphical description of an IRM . . . 73
4.4 IRM pre-validation configuration workflow proposed. . . . 76
4.5 General workflow of the source provenance knowledge acquisition. The provenance ontology represents instances of integrations which links a shape from the IRM with the two shapes that were the respective sources of the specific integration. . 80
4.6 Integration of two input SHACL shapes which resolve a conflict of unidirectional contained type for sh:maxCount SHACL constraint. . . . 80
4.7 Example of the knowledge related to an integration in the source provenance ontology proposed. The resources in green represent classes, the resources in black represent properties and the resources in violet represent individuals. . . . 81

5.1 Motivation for the integration of individual shape graphs which aim to a general RDF knowledge graph. *"S.G."* states for *"Shape graph"*. This image illustrates the target of each shape graph. . . . 84
5.2 The IRM knowledge is extended sequentially with each new integration of each SHACL shape graph. The bottom part of the figure shows the order in which the shape graphs are integrated with the IRM (e.g. the shape graph 1 goes first). The section in the middle of the image shows a set of source shape graphs which are desired to be integrated with the IRM. Finally, the top of the picture shows a Venn diagram which represents the universe of restrictions held by the IRM. The intersections in the diagram represent those restrictions found in equivalent shapes which were integrated. . . . 87
5.3 The integration procedure's main workflow. The black rectangles represent the sources. The green rounded rectangles represent procedures. The numbers in green circles indicate the order in which the workflow is executed. CRIP stands for "Conflict Resolution and Integration Process". . . . 87
5.4 The "Conflict Resolution and Integration Process" main workflow. . . . 88
5.5 Inconsistency among SHACL constraints defined in [50] . . . 93
5.6 A detail of the scenarios defined, which can appear in the comparison of two constraints which belong to equivalent shapes and are aimed to be integrated. . . 93

6.1 A representative image of a simplified ESF architecture which shows how data from all the resorts are centralized. . . . 96
6.2 Main classes of the ESF ontology. . . . 99
6.3 A graphical representation of the IRM generated as the integration of the Annecy SHACL shape graph with an empty IRM. . . . 103
6.4 A graphical representation of the updated IRM generated as the integration of the Chamonix SHACL shape graph with a previous version of the IRM which already contained the Annecy shape graph. . . . 103
6.5 The representation of the desired restrictions for the group lessons when there is a storm. . . . 105
6.6 Link of a class and its associated characteristics with a specific context. . . . 106

Chapter 1

Introduction

Contents

1.1 Motivation **1**
1.1.1 A deep dive into the specific field 3
1.1.2 SHACL shapes graphs integration 4
1.1.3 SHACL shape graphs for multiple contexts 5
1.2 Scope and objectives of the thesis **6**
1.3 Contributions summary **7**
1.4 Outline of the thesis **7**

1.1 Motivation

The current world wide web could be seen as an enormous network of connected documents and programs, where most of the representational content is plain text that can only be interpreted by the computers in order to show it in a nice format to the humans that will consume it. In this way, there is an important amount of valuable information that is not possible to process by the machines. In order to start a change in this sense, a sub-field of *Artificial Intelligence*, the *Knowledge Representation*, provided the necessary foundations. Based on this field, the *Semantic Web* was proposed by Tim Berners Lee in 2001 ([18]) as a new version of the internet, in which the machines could *"understand"* what they are storing and representing. This understanding is possible if the web is provided with structured content and inference rules that allow to reason automatically about the represented content. While this vision was revolutionary at the time, the Semantic Web has since evolved significantly, driven by advances in linked data, artificial intelligence, and data integration technologies. Recent advancements, such as the adoption of underlying languages in building Semantic Web's graphs and the integration of constraint languages (CLs) for semantic constraint validation, demonstrate the ongoing relevance of Semantic Web technologies ([54]). Furthermore, the main technology related to the Semantic Web is the *Resource Description Framework (RDF)*, which represents the real world concepts in the form of *triples (s,p,o)*, where *s* is the subject (e.g. "The Beatles"), *p* is the predicate or relationship (e.g. "is the author of") and *o* is the object (e.g. "Strawberry Fields"). Moreover, the graphs of the Semantic Web that represent RDF triples, are also known and interpreted nowadays as *Knowledge Graphs* (KGs). These representations are defined as formal, labeled and directed graphs, where the nodes represent concepts of the real world, meanwhile, the edges represent

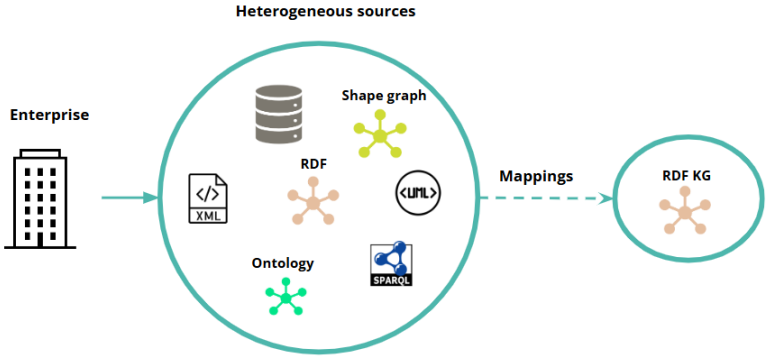


Figure 1.1: A common scenario which relates heterogeneous sources with Semantic Web models. Some of these sources represent data, others knowledge, schemas or even scripts. Through the design of mappings, these sources could be used for the construction of a related RDF Knowledge graph.

the existing relationships between the concepts. Normally, these models represent the knowledge at an instance level, with a lower expressiveness capacity (in comparison with ontologies [60]) and frequently fed from heterogeneous sources ([86]). Another family of technologies as the *Web Ontology Language (OWL)* are used for the creation of *ontologies*, which are conceptual, formal and shared representations of a part of the world that count with a rich expressiveness that allows to represent a wide amount of characteristics ([111]). Also, both the KGs and the ontologies, are normally designed for being interpreted with the *Open World Assumption (OWA)* and *Non-Unique Name Assumption (NUNA)*. The former states that a fact that is not in the explicit definitions could not be defined as false, but rather as unknown. In addition, the latter states that it's possible to have two entities with different names which are referring to the same real-world concept. These assumptions define a framework for the inference of new knowledge based on the explicit facts ([164]).

These models are nowadays adopted in the academy and in the industry as well, as part of research and innovation projects that aim to improve some aspects of their systems. Example of this adoption is the *Google Knowledge Graph* ([32]) which is being exploited by the related web search engine, or the *European Register of Infrastructure* (RINF[1]), which is a system based on Semantic Web technologies that aims to enable the search of the characteristics and compatibility of the European railways. Furthermore, it's common that systems such as the aforementioned ones, present a semantic web architecture, which involves the integration and processing of data from heterogeneous sources into an RDF KG. In these scenarios, data could come in different formats such as Extensible Markup Language (XML), JavaScript Object Notation (JSON), Comma-Separated Values (CSV), relational databases (RDBs) or RDF triples. Besides, these systems could also involve different models such as ontologies, shape graphs or even XML schemas. A representative image of this scenario is shown in figure 1.1. However, the presented models, and the mentioned systems, still have some drawbacks such as the complexity related to the integration of heterogeneous sources, the difficulties related to effectively represent and maintain knowledge, the scalability challenges when representing big data, the treatment of incomplete knowledge and the validation of the represented instance data ([85]).

In order to increase the adoption of these models, it's key to continue the related research and get to solution proposals. In particular, with respect to the validation of the models, there

[1]https://www.era.europa.eu/domains/registers/rinf_en (last accessed on 2/11/2025)

are many domains in which the consistency of the models has a vital importance because of the sensitivity of the outcomes, or even because of the desire to count with a model as trustable, effective and efficient as possible. These requirements left in evidence the need for having the possibility to add consistency to the models by the creation of some process which could validate the models in a closed way, such as it's done in RDBs. As a result, related research focused on developing knowledge representation models that adhere to the *Closed World Assumption* (CWA), where facts not explicitly defined are considered false, and the *Unique Name Assumption* (UNA), which ensures that each real-world concept is referred to by a unique name. These models were proposed to address the restriction and validation challenges of Semantic Web models, which are typically based on the OWA and NUNA ([154], [108]).

During the last years, some approaches have been proposed in order to be able to validate knowledge representation models in a closed fashion. More specifically, the focus has been *mainly concentrated on the validation of RDF graphs.* These approaches have evolved from methods centered on generating *SPARQL (which stands for "SPARQL Protocol and RDF Query Language")*[2] queries (the standard query language for RDF) or augmenting existing language models, such as OWL, with new related features, to the development of entirely new languages capable of representing the constraints of a model. From the latter type of approach, there were a few CLs that were created, such as *Shape Expressions (ShEX), SPARQL Inferencing Notation (SPIN), and the Shapes Constraint Language (SHACL)* ([98]), being the latter defined as the standard by the World Wide Web Consortium (W3C[3]). The SHACL models are also known as *"SHACL shape graphs"*, since this language respects a structure of "shapes" that represent groups of constraints expressed in the form of an RDF graph that targets and restricts one or more nodes of a data graph, which is itself another RDF graph that represents instance knowledge. To show how these graphs work, let's see the figure 1.2. In this image, the relationship between the shape graph and the data graph is shown. Moreover, there's a SHACL shape example which targets the instances which are of class Person (line sh:targetClass ex:Person). This shape expects that each Person's instance has as much one value for the Passport property (lines sh:path ex:Passport; sh:maxCount 1;). In addition, two RDF data graphs examples are shown, where the first one doesn't accomplish with the shape's restrictions (the person presents two different passports), meanwhile the second one respects the requirements of the shape.

1.1.1 A deep dive into the specific field

The validation of semantic constraints on knowledge representation models could be taken itself as a subfield of the knowledge representation and the Semantic Web technologies. Currently, there is an interesting amount of scientific works that could be directly or indirectly related to this goal (such as the ones defined in [137], [97], [34], [106]). Moreover, since the CLs arose a few years ago, many challenges have been stated by the community and many related proposals have been presented. These efforts are related to the improvement of these languages, the development of their interoperability with heterogeneous sources, the study of the way in which these models could be created and more. Although this field have presented an interesting activity during the last years, there's still a lack of a comprehensive analysis and review of all these works having as the common point the validation of models in knowledge representation. In consequence, one of the challenges that arise from this scenario is to understand the different related research lines that currently exist and to detect their limitations and research opportunities. A detailed description of the related types of contributions would help to create the bases for a better

[2]https://www.w3.org/TR/sparql11-query/ (last accessed on 2/11/2025)
[3]https://www.w3.org/ (last accessed on 2/11/2025)

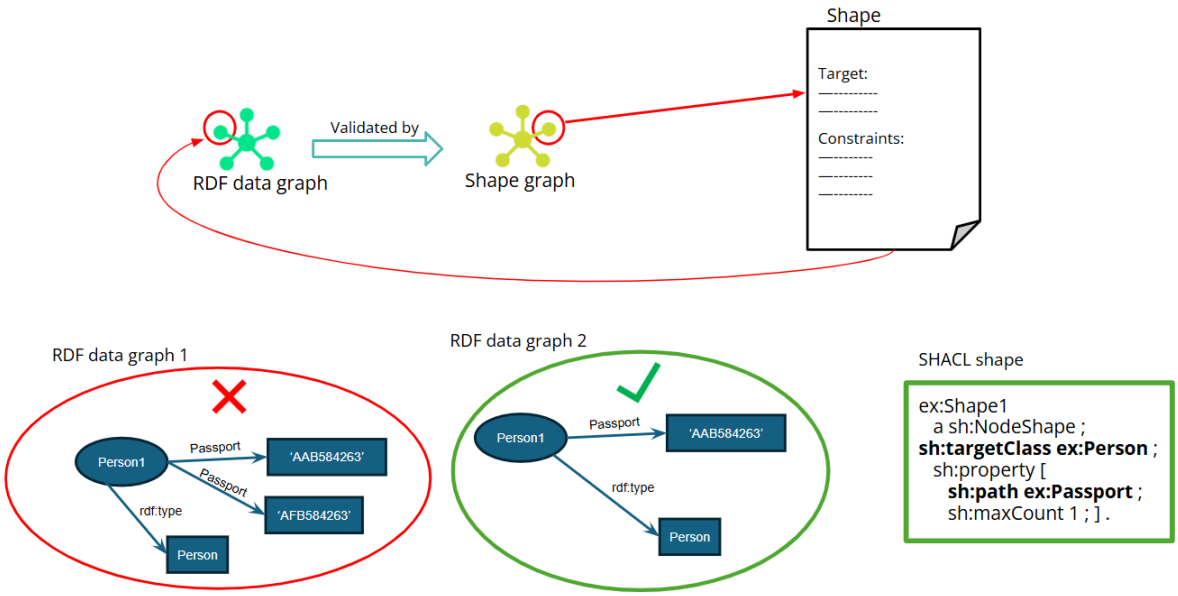


Figure 1.2: This figure shows first the relationship between a shape graph and a data graph in SHACL. Then, it shows a simple shape definition which restricts the persons to have one passport as maximum. Finally, two data graphs examples are shown, where the first one doesn't accomplish to the shape, meanwhile the second one does.

understanding of the associated challenges, paving the way for the further development of the field.

1.1.2 SHACL shapes graphs integration

The SHACL CL is the one with most adoption since it was defined as the standard CL for the Semantic Web by the W3C. However, there are still many related gaps which are being studied by the community and are reflected in an interesting variety of proposals. One of the most important ones is related to the way in which these constraint models are created. In most cases, they are created manually, which is a very time-consuming task and strongly error-prone. As an answer, some semi-automatic and automatic generation techniques have been proposed, in order to automatize, as much as possible, the creation of these graphs and increase their adoption ([97]). With these techniques, the shapes graphs are being extracted from one single model or source type, such as an ontology, or an RDF KG, between others. Moreover, the extraction of SHACL shape graphs from a source of another type, involves finding implicit or explicit patterns in the input, that could be mapped to the SHACL vocabulary. However, recalling the figure 1.1, there is a common Semantic Web scenario which includes heterogeneous sources as part of the architecture and it would be very valuable to take into account the characteristics that are represented in each of them. As a result, the extraction of a SHACL shape graph from more than one source would generate a scenario in which more than one shapes graph is obtained. This opens a new research line which will study the following natural step, meaning the integration of the obtained SHACL shape graph models (see the figure 1.3).

The integration of more than one input SHACL shape graphs is a complex task, which has as an objective, the generation of one individual SHACL shape graph which represents the restriction knowledge found in the inputs. This process includes a number of necessary tasks,

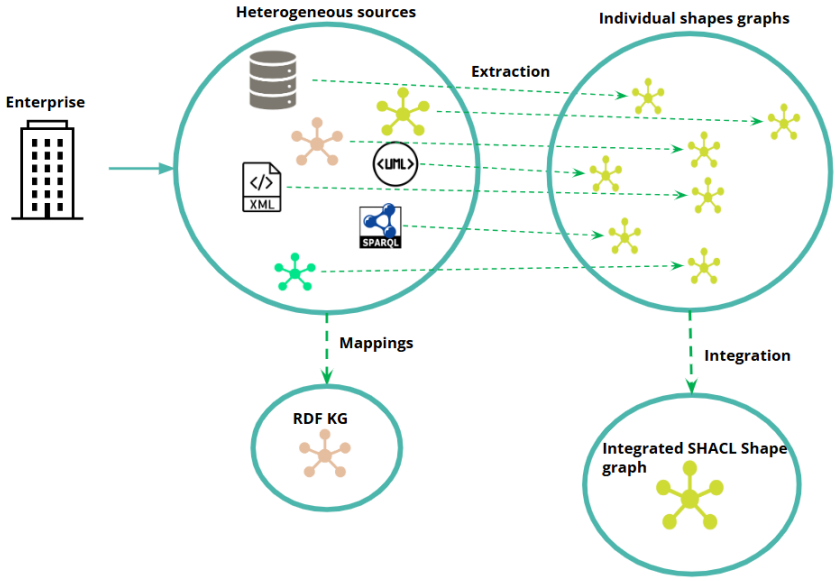


Figure 1.3: A scenario which shows the need for the integration of multiple shape graphs.

such as the decision of which characteristics of the shapes are taken into account in the definition of the shapes that are going to be integrated into one. This task also defines which shapes are not necessary to be integrated and which could be inserted directly in the outcome. In addition, having into account that each shape could have a set of constraints, an integration of two shapes includes the integration of their two sets of constraints into one. Moreover, during this process, many decisions have to be taken in order to resolve the integration of conflicting constraints which are not compatible. For this reason, a conflict resolution strategy definition is necessary for guiding the procedure. Furthermore, this strategy also defines the restriction knowledge that is kept in the output, as well as the way in which the shapes are constructed. Having all that said, the integration of SHACL shape graphs is a complex task which presents some interesting challenges.

The state of the art presents an interesting related proposal which includes a scenario in which more than one shapes graph is obtained and consequently, there is an interest in the integration of these models for the generation of a unified shape graph ([50]). Nevertheless, the richest integration of this proposal includes the resolution of conflicts by disjunction, which generates softer shapes in the outcome in comparison with the inputs.

1.1.3 SHACL shape graphs for multiple contexts

SHACL shapes graphs represent a set of constraints which are aimed to a specific part of the RDF data graph which is going to be validated. For example, the shape shown in the listing 1.1 validates those nodes of type *Person* which posses a property *nationalID* with a minimum cardinality of 1. In addition, as these graphs are RDF graphs, then it's possible to reuse restrictions from other graphs. However, it's important that those shapes which aim for the same targets are consistent enough between each other, otherwise a validation report with the related errors would be generated. For example, there's a shape that validates songs and defines that they must have a release date after the 2000's. In addition, there's a second shape that also validates songs but expects a release date after 1980's. Having these two shapes in a graph would represent an

inconsistency between them, creating a scenario in which a 1990's song would be validated by the second shape but rejected by the first one. An interesting challenge that comes out of this is finding the way to create a SHACL shape graph which is flexible enough to represent these inconsistent shapes, creating a scenario in which the shapes graph is able to answer to more than one different context, and depending on the current conditions, the correct shape is used. This would represent SHACL shapes graphs which are more useful, since they would be able to cover more scenarios than usual. Moreover, having different contexts with different characteristics, is a scenario that reflects better what happens in the real world use cases.

Listing 1.1: Example of a simple SHACL shape which validates the nodes of type person which have at least one national ID.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLessonShape
    a sh:NodeShape ;
    sh:targetClass ex:Person ;
    sh:property [
        sh:path ex:nationalID ;
        sh:minCount 1 ;
        ] .
```

1.2 Scope and objectives of the thesis

This thesis is a contribution to the *French Ski School (ESF stands for "École du Ski Français" in french)*[4], which is a French company dedicated to the management of ski resorts in France. The goal of the related project was to construct a Semantic Web environment which represents the specific domain business with knowledge representation models that involve the instance, conceptual and restriction levels of representation based on existing heterogeneous sources. Furthermore, **there is a particular interest in the validation of the related instances of knowledge, having a scenario in which there exist more than one shape graph and where the restrictions defined require a flexibility for being adapted to different possible contexts.**

In broad terms, **the objective of this thesis is to design rules and tools for constructing a unified shape graph within a semantic environment characterized by heterogeneous sources**. Moreover, these sources often lead to the generation of multiple shape graphs, and the scenario may involve diverse contextual requirements. In this context, the specific objectives of the thesis are the following:

1. Analyze and describe the validation related works which are directly or indirectly involved in the resolution of the consistency problem in knowledge representation models.

2. Based on the SHACL standards, provide a shapes graph design which is able to be adapted to different contexts.

3. Propose a method for the integration of SHACL shapes graphs which generates a unified SHACL shape graph that respects the design presented in the previous goal.

4. Show its potential usefulness in a practical scenario.

[4]https://www.esf-uk.co.uk/ (last accessed on 2/11/2025)

1.3 Contributions summary

The first defined specific objective (1) aims to a review of the main related trends. In order to attend it, a systematic literature review has been done, inspired in the methodology presented by Petersen et al. ([135]) and aiming to answer the main research question *"which are the existing trends related to the validation of semantic constraints on knowledge representation models?"*. As an outcome, the definition of a related taxonomy which includes 18 categories is presented. In addition, the articles selected through the review are classified in these categories and the associated current open challenges are enumerated.

The second specific objective of the thesis (2) seeks to the design of a shape graph model. To address it, an *Integrated Restrictions Model (IRM)*, based on SHACL standards, is designed and presented. This model will comprehend the outcome of the integration of SHACL shapes graphs and will keep equivalent (aiming at the same targets) conflicting shapes available. The IRM representation groups these conflicting shapes through the use of sh:group definitions and organize them in hierarchies that respect a pattern in which a sub-shape is contained in its super-shape. In addition, a pre-validation configuration of the IRM is designed in order to make it useful for the validation task.

The third defined specific objective (3) focuses on the design of an approach for the integration of SHACL shape graphs. To tackle it, a method has been presented for the integration of more than one SHACL shape graph which generate as an outcome, an IRM. This approach includes a conflict resolution strategy for the integration of SHACL shapes graphs which present equivalent shapes. This technique is mainly based on the containment relationship for the constraints comparison, meanwhile the resolution is guided by shapes generalization and specialization. Besides, a lightweight ontology for the source provenance information is presented. This structure aims to link each shape with its respective source shape graph.

Finally, the fourth defined specific objective (4) intends to test the two latter made contributions in a practical use case. In order to manage it, the integration procedure has been put in practice in the context of the Semantic Web environment created for ESF. More precisely, a few experiments have been carried out, which include the integration of two related SHACL shapes graphs, the generation of an IRM as an outcome and the flexible utilization of the generated IRM for the validation of related contexts that could vary. The latter includes a set of steps that conforms to the pre-validation configuration of the IRM to make it useful for the validation of the desired current context by assessing the user in the decision of the IRM sub graph that will be used for the validation.

1.4 Outline of the thesis

This thesis is structured as follows: First, the chapter 2 introduces the concepts which are needed for the understanding and development of the thesis, including Knowledge Representation, Resource Description Framework, Ontology, Knowledge graph, SPARQL, Semantic Constraints, CL and Shapes Graphs Containment.

Then, in chapter 3, the state of the art found in the literature related to the validation of knowledge representation models is described through the creation of a taxonomy, an article classification and research open challenges. In addition, the state of the art of the integration of shapes graphs is highlighted.

Following, in chapter 4, the design of the Integrated Restrictions Model is presented. Based on the SHACL standards, this model is able to represent shapes which aim for the same targets

and holds inter-shape conflicts. Furthermore, the details of how this model represents these groups of shapes and how they are expected to be managed are presented. This includes a hierarchical organization which is based on the shapes containment relationship. Finally, taking into account that the IRM is the result of the integration of more than one shape graph, an ontology for the representation of the source provenance information is described, which allows to identify which source graphs are related to each shape.

Further, chapter 5 presents an approach for the integration of multiple SHACL shapes graphs which generates as an outcome, an Integrated Restrictions Model. The main focus of the approach is in the resolution of conflicts for those equivalent shapes which are aimed to be integrated. In this regard, a constraint conflict resolution strategy based on the resolution by specialization/generalization of shapes is presented. This approach takes into account the types of constraints that are being compared, their respective values and the containment relationship that they hold. Then, the approach attempts to generate as an outcome a comprehensive combination of the individual constraints, generating new shapes with a variety of restrictive hardness. In addition, the outcome of the strategy is represented as a hierarchy of shapes where a sub-shape is contained in its super-shape.

Next, in chapter 6, a Python implementation of the integration procedure is briefly described. In addition, a set of SPARQL queries which are useful for the pre-validation configuration task and their management through a second Python implementation are presented. Furthermore, the ESF use case needs and characteristics are described and a few related experiments are done. This includes the execution of the integration procedure which generates a related IRM and the pre-validation configuration of the obtained model, taking into account the characteristics related to a current context. Overall, these experiments include a related use case Semantic Web environment which allows to evaluate the behavior of the contributions in a real-world scenario. Moreover, the correctness, the consistency, the usability and the performance of the contributions are analyzed.

Finally, chapter 7 concludes the thesis with the final thoughts and points to several interesting paths that represent how this research line could be further developed in the future.

2

Background

Contents

2.1 Introduction 9
2.2 Knowledge representation 9
2.3 Resource Description Framework (RDF) 10
2.4 Ontology 11
2.5 Knowledge graph 12
2.6 SPARQL Protocol and RDF Query Language 13
2.7 Semantic constraints 13
2.8 Semantic constraint languages 14
2.9 Relevant SHACL elements 15
2.10 Industry 4.0 and the Semantic Web 17

2.1 Introduction

In this section, the keystone concepts related to semantic constraints and the validation of knowledge representation models are briefly presented, approaching from the most general concept up to the most specific ones. In addition, the usefulness of the Semantic Web technologies for the Industry 4.0 is briefly described.

2.2 Knowledge representation

Knowledge representation has been defined by Davis et al. ([42]) as a surrogate, a set of ontological commitments, a fragmentary theory of intelligent reasoning, a medium for efficient computation and a medium of human expression. This kind of representation contains a symbolic description of a part of the real world, in the form of statements that reflect the facts related to a domain. Furthermore, the use of classes is a natural way to formally express certain knowledge types and roles at the knowledge level, as already advanced by Akkermans et al. ([11]). In addition, as a science which is a branch of the artificial intelligence, the knowledge representation aims to simulate the human intelligent behavior. In this regard, the background of these models is constructed on mathematical logics, which means that the knowledge represented could be then analyzed and processed by a machine through reasoning algorithms, getting

to implicit knowledge as the result. In other words, these models represent declarative knowledge (explicit knowledge) and procedural knowledge, which is a set of rules that provides logical inference ([130]). Moreover, by providing formal semantics, logic-based formalism lay the basis for the automated deduction. Without this, the representations could be ambiguous and hard to process. On the Semantic Web, two logical formalisms are the most used ones: Description Logic, which is delimited to unary and binary predicates and is a subset of first-order logic, and on the other hand, logic programming formalism ([161]).

In the current Semantic Web, knowledge representation appears in different forms, the most prevalent of which are based on semantic networks, rules and logic. Semantic networks can be found in RDF graph representations or Topic Maps, and they concentrate on expressing the taxonomic structure of categories of objects and the relations between them. Moreover, graphs are considered for knowledge modelling and for computation. A benefit of this approach is that labelled graphs, schemata and drawings provide an intuitive vehicle for knowledge representation ([133]). Following the latter, ontology and KG are two of the most popular knowledge representation models that could be presented as a graph. The work presented in this thesis is based on knowledge representation models which are graph-based. Furthermore, these networks are closely related to another form of knowledge representation called frame systems, but these use different representation metaphors. Regarding the rules and the logic representation, they often describe the business knowledge in the form of business rules or logic programming formalism ([161]).

The creation of knowledge representation models opens the possibility for the several interesting exploitations that could be made based on them. Some of these functionalities could be enumerated as follows: data integration, data governance, data enhancement, fraud detection, search improvement, chatbot creation and recommendation systems ([110], [15], [107], [5]).

2.3 Resource Description Framework (RDF)

RDF is the standard data model for representing knowledge in the Semantic Web and in the Linked Open Data[5]. This framework is defined as graph-based and its syntax follows the structure of a set of triples of the form (subject, predicate, object). A set of these triples is called an RDF graph, where the subject and the object are represented as nodes, meanwhile the predicate is represented as a directed relationship which goes from the subject to the object. Furthermore, the nodes could be represented as International Resource Identifiers (IRIs), literals (representing different data types such as strings, integers, dates, etc.) and blank nodes. The first two denotes a concept in the real world, having the IRIs as a unique way of representing a resource, and a literal representing a literal value of a specific data type which can only be represented by object nodes. On the other hand, the blank nodes do not specify a particular resource and instead, they represent the existence of something that is not identified. RDF supports simple semantics and it's based on XML, so it inherits the XML datatype definitions. Furthermore, the RDF vocabulary presents some properties such as the *rdf:type*, which defines that a resource is an instance of a class (this property could be found as *"a"*). Additionally, the RDF model is complemented by query languages in which SPARQL is the most prominent. The complete specification could be consulted in [39]. A graphical representation of a simple RDF graph is shown in picture 2.1.

In addition, the Resource Description Framework Schema (RDFS) is a semantic extension of RDF which provides a vocabulary for increasing the expressiveness of these models. This

[5]LOD, https://www.w3.org/egov/wiki/Linked_Open_Data (last accessed on 2/11/2025)

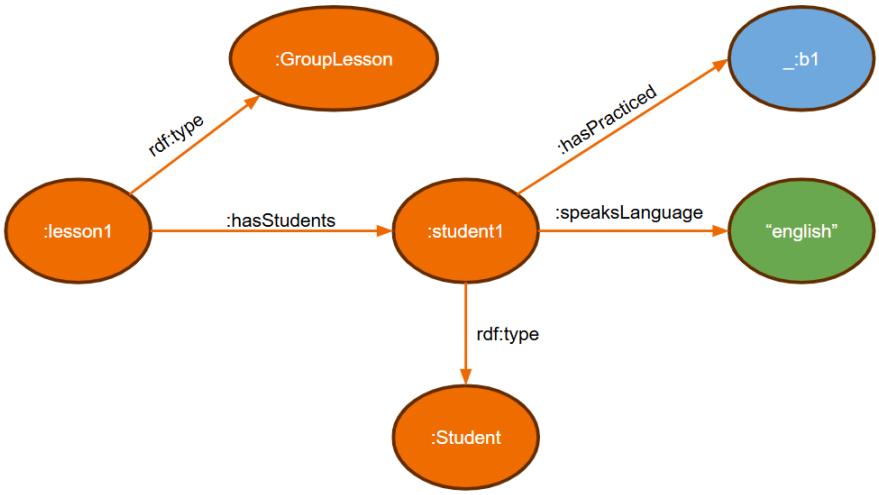


Figure 2.1: A graphical representation of a simple RDF graph example. The nodes in orange represent IRIs, the green one represents a literal and the light blue a blank node. This graph represents knowledge related to ski lessons of type groupLesson. In addition, the graph presents knowledge related to a student which participated in an instance of the aforementioned lesson.

framework allows to define classes for describing groups of related resources or entities. Moreover, this hierarchy could also be defined over the relationships that are represented as predicates (properties and subproperties). Finally, the expected types of subjects and objects related to a specific predicate can be determined through the definition of domain and range properties, respectively. This additional vocabulary allows to make richer representations and inferences. The specification of this framework can be found in [31].

2.4 Ontology

A well known knowledge representation model is the ontology, which provides the means for describing explicitly the concepts behind the knowledge represented in a knowledge base ([72]). There, the knowledge is formalized using five kinds of components: classes, relations, functions, axioms and instances. In addition, a well known ontology definition is the one given by Thomas Gruber ([74]) as a formal and explicit specification of a shared conceptualization. The ontology is defined as shared because once it's public, it's supposed to be accepted by different points of view of the community before it's reused. Furthermore, rigorously formal ontologies are defined in a language with formal semantics, theorems and proofs of such properties as soundness and completeness ([171]). There are three different levels of knowledge abstraction that need to be formalized: methodological knowledge (ontology language), conceptual level (classes and their relationships) and factual knowledge (individuals and their relationships). Moreover, ontologies are mostly used to represent the conceptual level of the knowledge, creating a kind of schema that represents the bases of a specific domain. Besides the general definition of public ontologies, there exist ontologies which are created and exploited in private environments such as enterprises and institutions. These organizations could prefer to keep their knowledge representation models as private ones as a way of protecting their structure, data and strategies. In addition, the ontologies could be classified depending on their complexity and expressiveness. On the one hand, the lightweight ontologies are simple, mainly focus on class hierarchies and don't describe

complex axioms or constraints. On the other hand, the heavyweight ontologies represent richer semantic relationships aiming to a deep understanding of the related domain ([36]).

Ontologies can be expressed on different description languages, and each of them means a trade-off between expressiveness and complexity of reasoning ([111]). In the Semantic Web community, the ontologies are commonly represented with the Web Ontology Language (OWL)[6] and its family, which is a knowledge representation language based on description logic ([102]). These languages present an interesting level of expressiveness, quite wider than the one offered by RDFS, allowing to represent characteristics of the resources and the properties such as the cardinalities, the type of properties depending on their behavior (transitive, symmetric, reflexive, etc.) or on their type of ranges (object property or data property), and much more. OWL has a rich vocabulary with several features, but in this section, only those properties which represent more relevance to this thesis are briefly described. To start, a class in owl (*owl:class*) is an abstract representation of a group of individuals with similar characteristics. In addition, the classes are defined with *class axioms* which could contain different constructs for the description of the characteristics and its hierarchical relationship with other classes. One of the constructs useful for the creation of a class axiom is *owl:equivalentClass*, which allows to define that the class being defined is equivalent to another class. Furthermore, the class axioms could include *class descriptions* of different types, such as *property restrictions*. A property restriction describes a class as the set of individuals that satisfies the proper restriction. An example of a property restriction is the *owl:maxCardinality* which limits the amount of allowed different individuals of the class. Finally, owl allows to define *properties* which could be of two different types: *Object Properties* (which link individuals to individuals) and *Datatype Properties* (which link individuals to data values).

2.5 Knowledge graph

A Knowledge Graph (KG) is a formal representation of instance knowledge in the form of a labelled directed graph, where the nodes represent concepts or an actual entity from the real world, meanwhile, the edges represent different directed relations between these nodes ([87]). A standard data model to represent KG is the Resource Description Framework (RDF) ([39]). In addition, the KGs could be fed with instance data from heterogeneous sources, such as RDBs, XML or JSON documents, between others. These feedings are possible thanks to the development of mappings with specific languages as the RDF Mapping Language (RML[7]), which allow to link the source structures with a target schema knowledge representation structure to finally generate an RDF as the outcome. The KGs are part of the linked open data, and consequently, a representation of the web of data. Since RDF was proposed as the standard model for publishing content on this new web ([19]), the amount of RDF KGs has been increasing and the most big and popular knowledge representations of this type are represented by this formalization language. As an example, Wikidata[8] is a very big open KG (it has more than 100 million concepts) that is exported on the RDF standard ([172]). Moreover, the RDF KGs could be ontology-based in order to add expressiveness to the model. In these cases, the models are more inter-operable since the ontological structure is unambiguous and it has an accepted and common meaning in the community ([87]).

Similar to what happens with the ontologies, KGs are not always created for being openly

[6] https://www.w3.org/TR/owl-guide/ OWL (last accessed on 2/11/2025)
[7] https://rml.io/specs/rml/ (last accessed on 2/11/2025)
[8] https://www.wikidata.org/ (last accessed on 2/11/2025)

published. There are alternative versions which are created in a controlled and private environment, where the architecture for the feeding of the model, the model itself and the exploitations based on it are kept protected in the structure of the organization. These are known as Enterprise Knowledge Graphs and aim to represent all the knowledge related to a specific company at a semantic level. In this way, the model allows the enterprise to benefit from all the interesting exploitations that could be done based on these models ([67]).

2.6 SPARQL Protocol and RDF Query Language

SPARQL[9] is the standard query language defined for querying RDF data. This language is useful for consulting and modifying the data which is stored in one or more RDF graph, where the data is stored in a triple RDF format. In other words, this language contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. In addition, there are different kinds of queries that could be created and executed, such as SELECT which directly returns variables and their bindings, CONSTRUCT which returns a single RDF graph, ASK that is useful only for returning a boolean answer to whether the data consulted exists or not and finally DESCRIBE returns a single result RDF graph containing RDF data about resources. The definition of ASK SPARQL queries has been used as an alternative approach for constraining a knowledge representation model in a closed manner (following CWA).

Moving forward, a brief overview of the most relevant SPARQL features for this thesis is presented. The *SELECT* query form is a type of query that returns variables and their bindings directly. Furthermore, this query is normally followed by a *WHERE* clause, which provides the basic graph pattern (including the variables defined in the *SELECT*) to match against the data graph. Moreover, the results shown could be constrained by using the *FILTER* keyword which, as its name suggests, filters those results which don't accomplish to the related expression definitions. Going forward, the *GROUP BY* term is used to calculate aggregate values for a solution which is first divided by the defined groups. Besides the use of *GROUP BY*, the term *HAVING* allows to apply a kind of filter over the grouped solutions. Finally, two more SPAARQL query forms are relevant to this thesis, which are described in the SPARQL Update specification[10]. On the one side, the *INSERT DATA* operation which directly adds a desired set of triples into the RDF data graph. Lastly, the *DELETE/INSERT* operation helps to remove or add triples from/to an RDF data graph, based on variable bindings obtained for a pattern specified in a *WHERE* clause. The deletion of the triples takes place before the insertion, and this is useful to avoid the insertion of a duplicate or a redundant triple. This thesis applies these types of queries for manipulating RDF graphs which are needed to modify.

2.7 Semantic constraints

The semantic constraints are a fundamental part of the knowledge representation model. They can be seen as rules applied to the models which define logical terms with the objective of defining logical consequences ([144]). Ontology modelling languages allow to define these constraints on classes, properties or on the entire ontology, and this increases the expressiveness that the model offers ([9]). These constraints could be of different types, with different objectives, such as the cardinality constraints, which restricts the amount of relationships that a set could have with another set. Another example is presented by the domain constraints, which restricts the

[9] https://www.w3.org/TR/rdf-sparql-query/ (last accessed on 2/11/2025)
[10] https://www.w3.org/TR/2013/REC-sparql11-update-20130321/ (last accessed on 2/11/2025)

type of entities that are part of the subjects of a specific relationship. Furthermore, the object property path constraints present an example of a more complex restriction which states that if an individual x is connected through a chain of object property relationships (r1, r2,..,rn) with another individual n, then the individual x is directly connected to the individual n with this type of property ([27]). In general, the RDF/s and the OWL models respect the OWA and NUNA. The former means that every fact that is not present in the data set is taken as unknown and cannot be taken as false. The latter states that an actual entity in the real world could be referred with more than one name or identifier. This configuration reflects that the semantic constraints on these models were not designed for validation purposes. In contrary, OWA and NUNA pave the way to the inference of new implicit knowledge as the result of logical consequences. However, when it comes to the Closed World Assumption (CWA) and the Unique Name Assumption (UNA), the semantic constraints could be dedicated to validate the consistency of the model. In those cases, they are known as integrity constraints. The community has been working on this issue and defined many approaches to create a solution. There are some works where the intention was to directly extend the formal definition of a language, adding the concept of integrity constraints (ICs). Examples of these works can be found in [154], [164] and in [108]. In addition, other works were based on the development of new languages, as it'll be detailed in the following section.

2.8 Semantic constraint languages

Many models published on the Semantic Web are types of graphs which do not have a rigid semantic schema behind, since they weren't designed for validation purposes. This means (besides other issues) that the model is not able to ensure consistency. Furthermore, their configuration with OWA and NUNA doesn't allow to validate the data in a closed manner as done in other models such as RDBs. In an attempt to solve this problem, the approach that gained most of the attention was to define Constraint Languages (CLs) that could be an add-on to the existing models ([169]). These new languages follow the design of a set of directed constraints which aim to validate the data following CWA and UNA configurations. The most popular CLs are the Shapes Constraint Language (SHACL)[11], which is the current standard defined by the W3C, and the Shapes Expressions (ShEx) language ([105]). Furthermore, the constraints described on these languages could be represented as graphs (known as "shapes graphs"), which could be used to validate the structure of RDF graphs (called in this case "data graphs"). In this process, there is a mapping of each constraint node which contains a targeted constraint or a group or targeted constraints, to a data node (or a group of data nodes). The creation and improvement of these languages have been attracting the industry and the academics during the last years ([97]). Moreover, there are other CLs that were defined for similar purposes. An example of this is the aforementioned ShEx, which is defined as a structural schema language for RDF graphs and was widely used before the standardization of SHACL. Finally, it's worth to mention that previously to this trend, some languages that were defined for other purposes, were adapted for their use as a CL. One example is the use of the SPARQL query language to define and validate semantic constraints as defined in [101]. Most of these approaches present the design of specific SPARQL ASK queries and through their execution, the specific constraint violations can be checked.

Furthermore, the characteristics of the relationship between two shapes graphs are interesting since their understanding is useful for the definition of procedures. In this thesis, the containment characteristic in the domain of shapes graphs is particularly useful. The study of the containment

[11] SHACL, https://www.w3.org/TR/shacl/ (last accessed on 2/11/2025)

defines whether a particular shapes graph is subsumed by a second one; that is, whether all graphs that are validated by the first are also validated by the second. This applies in a similar way at the shape and constraint levels. More specifically, the containment of SHACL has been studied and defined at a graph, shape and constraint level in [127] and [128]. These works studied the decidability of the containment by taking into account different fragments of the SHACL semantics. Since SHACL is the constraint language in which an important part of this thesis is based on, the following section is dedicated to describing the most relevant features of the language which are used in the contributions.

2.9 Relevant SHACL elements

SHACL is a constraint language which allows the validation of an RDF graph, in the sense of verifying the consistency of the model against a designed shape graph which represents the desired structure of the graph. It can also be seen as a way of describing an RDF Graph through the set of conditions (if it validates the conditions). Moreover, this is a broad language which vocabulary comprehends several classes, properties, constraints and more. However, when it comes to the contributions of this thesis, there are some elements which have a particular interest.

To start, there exist two types of shapes: the *Node Shapes* which target a specific node (called *focus node, fn*) and the *Property Shapes* which target a *value node vn* to which is possible to arrive from the *focus node fn* through the predicate *path px*, which could be represented as the triple pattern *(fn,px,vn)*.

Then, the element *sh:node* is useful for stating that the targeted set of nodes *t* of the shape in which the element is being declared, must conform to a specific *node shape* which is declared as the *value* of the element *(sh:node nodeShapeIRI)*. In addition, the *sh:node* element could be used either to reuse constraints but also to create hierarchies of shapes. This is mostly done when the element is declared in a *node shape*, stating that all the corresponding *focus nodes fn* of that shape must conform to the node shape specified in the element. Furthermore, the element *sh:property* provides the means to define a property shape that all the related value nodes *vn* must accomplish. This element could define also a property shape *PSx* as a blank node nested in a node shape *NSx*.

Furthermore, the *sh:targetClass element (Class-based targets)* is a predicate that allows the definition of the *set of instances of a specific class c* as the *set of targets t* of a specific *node shape*. There exist other elements for the definition of targets in different manners, however, this element is currently the one that is used in the contributions of this thesis. Being able to define all the instances of a class as the set of focus nodes brings clarity and simplicity.

Going forward, the core definition of SHACL presents different types of *Constraint Components* which are listed as follows: *"Value Type"*, useful for the restriction of the type of value nodes with specific elements such as *sh:nodeKind*; *"Cardinality"*, which allows to restrict the quantity of value nodes for a given focus node (e.g. *sh:minCount*); *"Value Range"*, that defines the range which must be met by value nodes, which can be compared with operators as '<=' (e.g. *sh:minExclusive*); *"String-based"*, which restricts the string representation of value nodes (e.g. *sh:minLength*); *"Property Pair"*, that specifies conditions on the sets of value nodes with respect to other properties (e.g. *sh:equals*); *"Logical"*, which represents the logical operators such as *sh:and*; and finally there are constraint components defined as *"Other"* which include interesting constraints such as *sh:hasValue*, that is useful for the definition of values which at least one value node vn must represent.

Next, there are a few SHACL constraints which accepts as a value, one of a set of *prede-*

fined values. An example of this is the constraint *sh:nodeKind*, which accept as a value one of the following: *sh:BlankNode, sh:IRI, sh:Literal sh:BlankNodeOrIRI, sh:BlankNodeOrLiteral and sh:IRIOrLiteral*. These constraints are highlighted because they play a particular role in the integration procedure.

Also, the SHACL definition provides a set of *syntax rules*[12] which must be respected by every shape of a shape graph. Those shapes which respect these rules are defined as *"well-formed"*, meanwhile those shapes which don't respect at least one of these rules are defined as *"ill-formed"*. Examples of these rules are *'maxCount-datatype'*, which defines that the values of the constraint *sh:maxCount* are literals of the datatype *xsd:integer* or *'maxCount-maxCount'*, which defines that the same constraint can appear at most once in a property shape.

Moreover, the element *sh:group* is a *"Non-Validating Property Shape Characteristic"*. This property allows to cluster a *set of property shapes PS = {ps1, ps2, ps3, ..}* through their link to an instance of a *sh:PropertyGroup*, which is an *IRI* that could have extra properties associated. Normally, these groups are used for application purposes such as user-interface design.

Additionally, the property *sh:deactivated* is useful for disabling a specific *shape Sx* of a shape graph. The property has *boolean values*, and when it's set to *true*, then all the RDF data that is validated against *Sx* will conform to it, independently on if it's accomplishing the *set of constraints C of Sx* or not.

An example of a SHACL shape it's shown in the listing 2.1. The first two lines defines the prefixes that are used in the shape, where *sh* is the prefix for the SHACL standard language and the prefix *ex* aims to an ontology from which the properties and classes are referenced. This shape is more precisely a node shape (line 5) identified with the IRI ex:InstructorShape. The line 6 defines the target of the node shape (focus nodes fn) as those instances of class ex:instructor. The line 7 defines a nested property shape, which is represented as a blank node. Next, the line 8 states that the property shape will constraint the value nodes vn which are connected to the focus nodes through the property ex:teaches_discipline. Besides, line 9 defines the sh:class value type constraint, which restricts the value nodes to those instances of class ex:Discipline. Finally, the line 10 defines a minimum cardinality of one. In summary, this shape can be interpreted as a profile of instructors which must teach at least one discipline.

Listing 2.1: Example SHACL shape

```
1  @prefix sh:   <http://www.w3.org/ns/shacl#> .
2  @prefix ex: <http://localhost/ontology/esf_ontology/> .
3
4  ex:InstructorShape
5      a sh:NodeShape ;
6      sh:targetClass ex:Instructor ;
7      sh:property [
8          sh:path ex:teaches_discipline ;
9          sh:class ex:Discipline ;
10         sh:minCount 1 ;
11     ] .
```

Finally, the validation report in SHACL is another RDF graph which details the validation results that are generated by a specific validator, once the validation of a SHACL shape graph against an RDF data graph has been done. Even though the structure of this report is left to each specific validator implementation, there are some arguments that are mandatory to be included. *sh:conforms* allows boolean values and details if the graph conformed to the SHACL document (true) or not (false). An example of a validation report which details that the RDF graph conforms to the respective shape graph is shown in the listing 2.2. In case that the RDF

[12] https://www.w3.org/TR/shacl/#syntax-rules (last accessed on 2/11/2025)

data graph doesn't fully conforms to the SHACL document, the property *sh:result* will contain a set of details related. Furthermore, the property *sh:ValidationResult* reports individual SHACL validation results, meaning details related to each node, which didn't conform to a shape. This validation result must present, as mandatory, values for *sh:focusNode* (the specific focus node that caused this result), *sh:resultSeverity* (which value is defined in each shape with *sh:severity* property, although there's a default value in case it's not) and *sh:sourceConstraintComponent* (details the specific constraint that caused the result). In addition, more details could be presented depending on each validator implementation. Taking into account the SHACL shape defined in the listing 2.1, consider an RDF graph which is validated against this shape and presents the triple shown in the listing 2.3. Since in this case the RDF data doesn't conform to the Shape graph, a validation result is generated, which is shown in the listing 2.4. This report states that there was an error of severity Violation (line 6) and it defines that the constraint which hasn't been accomplished is the sh:minCount (line 7). Moreover, the report identifies the specific shape which generated this result (line 8), the specific focus node (line 9), the related path (line 10), and finally, it provides a text message which is useful to clarify the result.

Listing 2.2: SHACL validation report of a data graph that conforms to a shapes graph

```
1  @prefix sh:   <http://www.w3.org/ns/shacl#> .
2
3  [   a sh:ValidationReport ;
4      sh:conforms true ;
5  ] .
```

Listing 2.3: Simple RDF data graph example

```
1  @prefix ex: <http://localhost/ontology/esf_ontology/> .
2  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3
4  ex:Fransson rdf:type ex:Instructor .
```

Listing 2.4: Example of a validation result generated when an RDF graph doesn't conform to a SHACL shape graph.

```
1  @prefix sh:   <http://www.w3.org/ns/shacl#> .
2  @prefix ex: <http://localhost/ontology/esf_ontology/> .
3
4  [
5          a sh:ValidationResult ;
6          sh:resultSeverity sh:Violation ;
7          sh:sourceConstraintComponent sh:MinCountConstraintComponent ;
8          sh:sourceShape ex:InstructorShape ;
9          sh:focusNode ex:Fransson ;
10         sh:resultPath ex:teaches_discipline ;
11         sh:resultMessage "Less␣than␣1␣values" ;
12 ] .
```

2.10 Industry 4.0 and the Semantic Web

Semantic Web technologies are valuable across multiple domains due to their ability to represent knowledge in a formal and structured manner. One notable application of these technologies is in Industry 4.0, where they contribute to innovative developments that push the boundaries of current industrial systems. A common challenge in this domain arises from the need for multiple systems and devices to communicate seamlessly, often leading to issues related to interoperability and the standardization of representations. In this context, Semantic Web models provide

standardized data representations that facilitate interoperability between heterogeneous systems, ensuring seamless data integration and communication ([178]).

These standardized representations are actively improving key tasks in Industry 4.0, such as real-time condition monitoring and predictive maintenance in manufacturing lines ([71]). Furthermore, Semantic Web technologies enhance cooperation between different industrial components and support more informed decision-making. These improvements align with the ongoing trend toward smart manufacturing, where automation and data-driven processes are transforming industrial operations ([12]).

Beyond conventional applications, the integration of Semantic Web technologies with Cognitive Digital Twins (CDTs) is further revolutionizing Industry 4.0 by enhancing data interpretation, interoperability, and autonomous decision-making. By leveraging Semantic Web standards like RDF, OWL, and SPARQL, CDTs can define and retrieve knowledge in a way that establishes meaningful connections between diverse data sources. This capability enables a deeper contextual understanding of operational environments, allowing industries to optimize workflows, predict failures, and automate operations with greater precision.

For example, in smart manufacturing, semantic-enhanced CDTs integrate data from production lines, logistics, and supply chains, facilitating dynamic process optimization and reducing downtime. Similarly, in the energy sector, these technologies contribute to intelligent grid management by analyzing real-time consumption patterns, weather forecasts, and renewable energy data to improve resource distribution ([16], [109]). In smart cities, CDTs utilizing Semantic Web technologies enable interconnected infrastructure systems (such as traffic management, public utilities, and emergency services) to synchronize data exchange, improving response times and resource efficiency ([57]).

By structuring knowledge and enabling automated reasoning, Semantic Web technologies empower CDTs to support intelligent, autonomous decision-making. This advancement paves the way for more adaptive and innovative industrial systems, fostering efficiency, sustainability, and resilience in Industry 4.0. As research progresses, the adoption of Semantic Web principles will continue to drive the evolution of digital twins and industrial automation, addressing challenges such as ontology management, computational efficiency, and data security ([17]).

3

State of the art

Contents

3.1 Introduction . . . 20
3.2 Related reviews . . . 20
3.3 Method . . . 21
3.3.1 Research question . . . 22
3.3.2 Keywords and search string definition . . . 23
3.3.3 Database selection . . . 24
3.3.4 Study selection and quality assessment . . . 25
3.3.5 Data extraction . . . 27
3.3.6 Analysis and classification . . . 27
3.4 Results of the review . . . 28
3.4.1 Publications year and venue (RQ1 and RQ2) . . . 28
3.4.2 Application domains (RQ3) . . . 29
3.4.3 Shared repository (RQ4) . . . 31
3.4.4 Knowledge representation models involved (RQ5) . . . 33
3.4.5 Semantic constraints and their representation (RQ7 and RQ8) . . . 34
3.4.6 Evaluated characteristics (RQ9) . . . 35
3.5 Type of contribution taxonomy (RQ6) . . . 36
3.5.1 Constraint language analysis and review . . . 38
3.5.2 Constraint language semantic and syntactic definition . . . 38
3.5.3 Constraint language implementation . . . 38
3.5.4 Constraint language applied cases . . . 41
3.5.5 Constraint language recursion . . . 41
3.5.6 Constraint language satisfiability and containment . . . 41
3.5.7 Constraint language further characteristics . . . 42
3.5.8 Mapping techniques . . . 42
3.5.9 SPARQL constraint implementation . . . 43
3.5.10 Definition of specific types of constraints . . . 44
3.5.11 Schema extraction . . . 44
3.5.12 Visual constraints creation and validation . . . 45
3.5.13 Validation reports . . . 45

3.5.14 Error treatment 46
3.5.15 Inference in validation 46
3.5.16 Validation of semantic constraints for query optimization 47
3.5.17 Validation algorithms 48
3.5.18 Other approaches 48
3.6 Discussion 49
3.6.1 Integrated results 49
3.6.2 Research gaps 52
3.7 Shape graphs integration related work 53
3.7.1 Integration of constraints from heterogeneous sources 53
3.7.2 Integration of SHACL shape graphs 54

3.1 Introduction

As part of this thesis, a Systematic Literature Review (SLR) was done in order to understand and know in a deeper way the specific domain. This SLR is based on research articles related to the validation of semantic constraints on knowledge representation models, which were published until September 2023. This analysis is guided by a set of research questions (RQs) that plan to investigate different types of details. A part of these questions aim to study the demographic characteristics, including information such as the year of publication and the venues where the papers were published. Other questions aim to analyze other types of data, such as the domain of study in which the authors have applied their contribution, the amount of papers which shared a repository in order to ease the reproducibility of the presented experiments and the characteristics that are taken into account in the evaluation of the proposal. However, the questions with more relevance aim to analyze the information related to the main topic of this review, such as the knowledge representation models involved in the works (ontology, RDF graph, etc.), the type of semantic constraints that are being treated and the languages in which the semantic constraints are being represented. Finally, the most relevant RQ aims to define the type of contributions presented in the selected articles, being a direct answer to the main objective of this review in order to conform the state of the art related to the validation of semantic constraints on knowledge representation models. This includes every contribution that performs in some part of the proposal, the validation of these kind of models. In summary, the contribution of this review includes a taxonomy which describes a set of categories of related research trends, the classification of the selected articles in these categories and the enumeration of the current open challenges on the validation of semantic constraints in knowledge representation. Each of these contributions is obtained through the conduction of the aforementioned SLR. In addition, a discussion based on the results is presented, where the research trends which were more active during the last years and present more potential are highlighted. Some related conclusions are stated, and the most interesting challenges found are defined as part of the future steps of the authors. Finally, based on one of the categories of the defined taxonomy, an extra state of the art is described, which is related to the integration of shape graphs.

3.2 Related reviews

There are three works that could be taken as related works to this article, and that are briefly described in this section. The first article found was “Requirements on RDF Constraint For-

mulation and Validation" ([80]), which presents a review of the existing requirements related to the definition and validation of semantic constraints. It presents an analysis of many types of constraints and constraint languages (CLs) and defines if they can manage the aforementioned requirements. The second work identified was "RDF Validation: A Brief Survey" ([169]), which shows a description and comparison of the state of the art CLs at the moment. Finally, the third related paper is "Challenges in RDF validation" ([97]), which presents several challenges that are interesting to research related to RDF validation. However, most of the article is dedicated to the comparison of two existing CLs, ShEx and SHACL.

Even if the aforementioned articles are interesting and present valuable information related to the validation of semantic constraints on knowledge representation models, none of them are based on a SLR method. Furthermore, considering the ongoing advancements in this area, the work presented here serves as an updated review of the topic. Going forward with the differences, the approach of [80] is limited to the study of the constraint formulation and expressive capabilities of the existing CLs. The core of the work presented in [169] is quite similar to the latter, since the objective is to define the requirements that a CL should full fill and the overview is limited to the existing approaches defined in RDF validation. Moreover, in [97] the review is limited to a comparison between the core version of ShEx and SHACL languages through a rewriting of both languages to a new intermediate language. The three found related works differ to the review presented in this thesis, since this is the first SLR focused on the validation of semantic constraints on knowledge representation models from a broad point of view. Furthermore, this SLR is open to all the models, constraints, CLs and taking into account also the intersection between the solutions for this issue and its relation to other well-known issues.

3.3 Method

In this section, the methodology followed in the construction of the state of the art is explained in detail. There are different ordered steps, and each of them generates the input for the following one. The last steps will generate a group of selected papers.

The SLR methodology used in this thesis is inspired in the work presented by Petersen et al. ([135]), with the objective of reviewing the scientific work in a research area and identify the state-of-the-art. The methodology requires the definition of RQs, find remarkable papers related to them, review these papers, select the most important ones, classify them and define research challenges. Similar reviews have been applied in different published works as in [40] and [64].

The main aim of this review is to identify and analyze all the relevant works related to the **validation of semantic constraints related to knowledge representation models**. The selected papers and their distributions will be analyzed, guided by a set of RQs defined in the following subsection. Through this analysis, the activity related to the aforementioned domain will be described and characterized in different ways, having as the main contribution the definition of a taxonomy which represents the research trends related to the topic and a classification of the selected works with respect to this taxonomy. The defined research trends could be related not only to the validation task, but also to a combination of this activity with other techniques such as the mapping of constraints or query optimization. In addition, the research gaps related to each group will be then analyzed, and consequently, open research challenges related to the main topic will be described. The findings of this review, present a clear picture of the challenges related to the validation of semantic constraints on knowledge representation models and states several directions for the further development of the domain.

This work starts from a clear and well defined research area, and follows the aforementioned

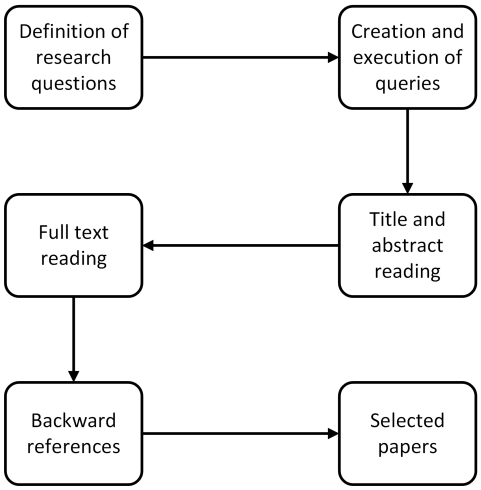


Figure 3.1: The Systematic Literature Review Steps

structured steps, that help to reduce the subjectivity of the study and allow to evaluate the relevant studies related to the topic. The applied methodology starts with the *"Definition of the main research question/s"* step, that represent the scientific topic that is aimed to be reviewed. The next step is *"Creation and execution of queries"* and consists first in the definition of the main keywords, which are extracted from the main concepts present in the defined RQ/s, with the addition of their synonyms and related words. Second, and following the advice given in ([93]), four scientific databases were selected. Third, the keywords are used for the creation of the main query that then will be adapted and executed in the selected scientific databases. Finally, this step ends with the results of the execution of the query, where a group of articles is obtained. Those papers which are in the group and are not duplicated, are the input of the next step *"Title and abstract reading"*, which is the first filter stage and consists in reading the title and the abstract of each of the input papers and their selection or rejection that is based on previously defined inclusion and exclusion criteria. Those papers that are selected in this stage, represent the input of the following filter stage, namely *"Full text reading"*, which consists in the inclusion or exclusion of the input articles based on the findings caught from the reading of the full text. Moreover, those papers that go through the full text reading stage represent the base of the final selected papers and are the input of the following step, which is the *"Backward references"*. This stage represents reading and analysing the references that are present in the selected articles. Those references which accomplish with the selection criteria will be added to the group of papers, having as the outcome the final group of selected works. The figure 3.1 describes the steps of the SLR method, which are also detailed in the next sections.

3.3.1 Research question

The goal of this review is to identify all the relevant literature related to the validation of semantic constraints on knowledge representation models. This leads to the following definitions:

The main research question (RQ):

- **RQ:** Which are the existing trends related to the validation of semantic constraints in Knowledge representation models?

Demographic research questions:

- **RQ1:** When the articles have been published?
- **RQ2:** In which venues the articles have been published?

Specific research questions:

- **RQ3:** In which domains of study are the validations applied? This includes, for example: Medicine, Biology, Industrial engineering.
- **RQ4:** Which of the articles shared a repository with the resources used in the experiments?
- **RQ5:** Which are the knowledge representation models involved?
- **RQ6:** Which are the types of contributions presented? This includes, for example: Definition of constraint languages, mapping techniques.
- **RQ7:** Which types of semantic constraints are represented?
- **RQ8:** Which languages are used for the representation of the semantic constraints?
- **RQ9:** Which are the characteristics involved in the evaluation of the contributions?

3.3.2 Keywords and search string definition

The following step in the methodology is searching in scientific databases using some defined keywords. The keywords definition is done with the intention of giving a structure to the search, in line with the defined main RQ, and therefore, with the goal of the review. To formulate them, the PICO (Population, Intervention, Comparison and Outcomes) methodology defined by Kitchenham and Charters ([94]) is applied having the following configuration:

- Population: Works related to the validation of semantic constraints on knowledge representation models.
- Intervention: Methods, tools, procedures or theorems.
- Comparison: In this study, the comparison is done between the different detected approaches that accomplished with the objective target detailed in the population.
- Outcome: Set of classified articles which detailed the aforementioned approaches.

The identified main keywords are *"semantic constraints"* and *"knowledge representation"*. To organize them, and the consequent creation of the query for the databases, the keywords are divided into two groups (*"A"* and *"B"*), with their respective synonyms and related words. The first group has terms related to the semantic constraints, aiming to ensure that the articles will present the representation of restrictions which could be validated. Furthermore, the second group is related to the knowledge representation models, aiming to obtain articles where the semantic constraints are represented in these kind of models. In order to construct the final query, the words that are part of the same group are connected with an *"OR"* Boolean operator, meanwhile, both groups are connected with an *"AND"* Boolean operator. The groups are detailed in the table 3.1 and a general final query was defined as:

("Semantic constraint" OR "Semantic restriction" OR "Ontological constraint" OR Shapes OR SHACL OR SHEX) AND ("Knowledge Representation" OR "Knowledge formalization" OR

"Semantic Network" OR RDF OR OWL OR Ontology OR "Knowledge Graph")

The selection of the keywords is done after testing several combinations of potential keywords, analyzing the influence that they had in the obtained results. *"Semantic constraints"* and *"Semantic restrictions"* are taken as synonyms, since most of the literature uses them interchangeably. *"Ontological constraints"* represents constraints which are represented in an ontology. The *"Shapes"* keyword is related to a popular semantic constraint format, which is represented by CLs. The keywords *"SHACL"* and *"ShEx"* represent the two most popular languages used for represent shapes constraints. Terms related to *"Shapes"* are added to the query as a consequence of the popularity of these techniques for the creation of semantic constraints at the time when this review was started. Regarding the group B, the term *"knowledge representation"* refers to the artificial intelligence domain in which the models of interest are studied, meanwhile, *"Knowledge representation"* aims to those models which respect a semantically structured framework. The keyword *"Semantic network"* refers to the most popular type of the aimed models, meanwhile the terms *"Ontology"* and *"Knowledge graph"* target the two specific representations. Finally, the keywords *"RDF"* and *"OWL"* are added, since they represent very well-known languages for the representation of the aforementioned models. It is good to mention that the creation of a third group of keywords related to *"validation"* was analyzed during the design of the search. However, it could be stated that the validation concept is implicitly represented in the group A, since one of the most frequent tasks done with semantic constraints and CLs is the validation. In order to confirm this, some queries with this third group included were tested, finding that the new group generated an important impact in the initial set of papers obtained, decreasing it almost a ninety percent. This low amount of papers would mean a very likely possibility of losing important articles related to the RQ. Furthermore, many times the keywords are not mentioned explicitly in the title, neither in the abstract, and there could be many ways of referring to a validation work. Consequently, the authors considered enough the design of the search having two groups of keywords, and analyzing the content to detect the kind of works of interest.

Table 3.1: Table to describe the groups of keywords

Group A	Group B
Semantic constraint	Knowledge representation
Semantic restriction	Knowledge formalization
Ontological constraint	Semantic network
Shapes	RDF
SHACL	Ontology
SHEX	Knowledge graph
	OWL

3.3.3 Database selection

Four databases were selected for this work: *Scopus*, *Web of Science*, *IEEE Xplore* and *ACM*. They were the selected ones since they cover different scientific fields, including peer-reviewed articles related to computer science, Semantic Web journals and conferences. Furthermore, they offer advanced search features, what helps to create customized queries in a structured way. This is valuable for creating a reproducible set of papers. The selected databases include some differences in the syntax of their search tool that makes necessary to adapt the aforementioned

general query and to create a specific version for each of them. The adapted queries for each database are detailed in table 3.2.

Zotero[13], a reference management tool, was used to store and organize all the obtained papers and their references. In addition, the collaborative online tool *Scolr*[14] was used to load the metadata related to the set of papers obtained from each database, delete the duplicates and discard those articles that have incomplete information (title, abstract or year). This tool was also used for the classification of the papers. This study was conducted by the end of August 2023, so the related articles published up to that moment are taken into account.

3.3.4 Study selection and quality assessment

In the beginning, the papers that had at least one combination of two keywords (one from group A and one from group B), in their title, abstract or keywords are selected. This initial search retrieved 2715 papers. There are four filtering stages that are applied, based on the inclusion and exclusion criteria defined in table 3.3. It is worth to mention that the exclusion criteria EC7 is defined taking into account that the first OWL specification was done in 2004[15], so it is not expected to find articles related to the validation of knowledge representation models before that year. In order to have some extra margin, the search is done starting from 2001. The four stages are: filters on databases (creating a new and adapted final query per database), discarding duplicates through Scolr, reading the title-abstract and reading the full-text. In addition, more articles are added through the application of backward snowball sampling.

After the search in each database with their corresponding query detailed in the table 3.2, filters are added to each query in order to accomplish the IC3, EC1, EC2, EC3 and EC7 inclusion/exclusion criteria as part of the first filter stage. The general query is first adapted to the Scopus database syntax, adding the following database filters: only the articles published between 2001 and 2023, being of type "conference paper", "article", "conference review", "book chapter" and "review", written in English and related to the "Computer Science" subject area, are taken into account for the next stage. After applying all these filters, 1054 documents were obtained. Going forward with the process, the same steps are applied using the databases Web of Science, IEEE Xplore and ACM, obtaining 606, 524 and 531 results respectively. The final version of the queries is detailed in the appendix A.

Once applied the filters aforementioned, the set of papers obtained from the selected databases is loaded into the tool Scolr. Taking advantage of the utilities of the tool, at this point, the duplicates are removed and the papers that have the title, abstract or keywords not available are automatically discarded. In other words, during the second filter stage, the EC4 and EC6 exclusion criteria are applied. The papers obtained from Scopus are the first ones being uploaded to Scolr, and all of them went through this filter. Going forward, the set of Web of Science is uploaded and compared with the previously uploaded articles. As a result, 415 articles were discarded and 191 new articles were added to the articles list. The following set to upload is the IEEE Xplore one, from which 149 are discarded meanwhile 375 were added to the selected list. Finally, in the case of the ACM set, 177 articles are able to reach the next stage, since 354 were discarded. As a result of the whole filter, a set of 1797 papers was obtained. The detail of the obtained papers per database and their uniqueness is detailed in table 3.4. In this table, the "Duplicated" column reflects the amount of duplicated papers with respect to the set of papers conformed by the scientific databases detailed on the previous rows. As an example, the first row

[13]https://www.zotero.org/ (last accessed on 2/11/2025)
[14]Scolr, https://scolr.lifia.ar/ (last accessed on 2/11/2025)
[15]OWL, https://www.w3.org/OWL/ (last accessed on 2/11/2025)

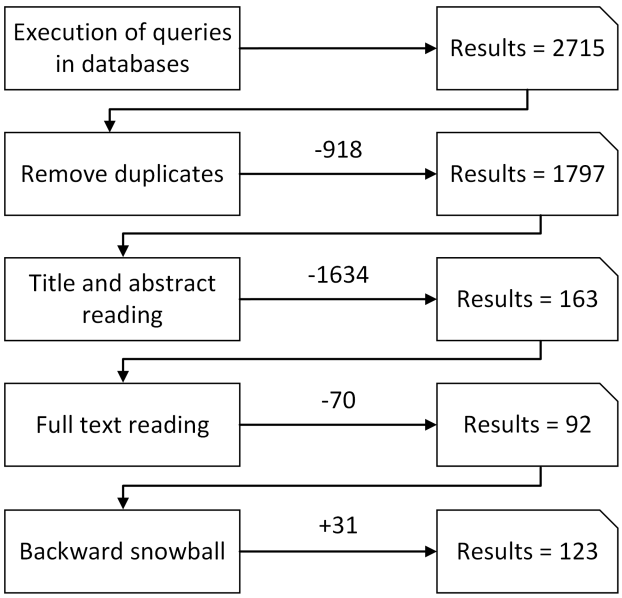


Figure 3.2: Number of articles included and excluded in each step of the review.

describes that no duplicates were found between the papers retrieved from Scopus, meanwhile the second row presents that 149 papers from IEEE Xplore were already retrieved on the Scopus set.

The papers that are finally uploaded into Scolr represent the input of the third filter stage. This consists in reading the title and abstract applying all the inclusion and exclusion criteria, with a special focus on the IC1 and IC2, looking for studies related to the task of validating semantic constraints in existing knowledge representation models. The criteria that were already applied to the previous filter stages are applied again with the objective to reassure the quality of the selected papers and discard any error that could have been made by the automatic procedures. In this stage, the articles that do not accomplish any of the criteria are classified as "excluded" and discarded. Furthermore, the papers that align with all the criteria are classified as "included" and saved for the next filtering stage. At this point 1634 articles were excluded, meanwhile 163 were selected for the following stage. Most of the articles discarded at this stage were the result of the noise that the keyword "Shape" generated, retrieving many papers that were related to geometric shapes represented in formal knowledge models.

Based in the group of papers obtained as the outcome of the previous filter stage, the fourth stage consists of the reading of the full text looking for valuable information related to the RQs of interest, with the goal of defining them as included or excluded but this time with a deeper comprehension of the content (IC1 and IC2). This leads to discard 70 papers, leveraging a result of 92 articles.

The selected papers after the last filter stage were used to conduct a backward snowball sampling ([90]). This represented a full text reading of the references that were present in the papers aforementioned. If there's an interesting article in this group of references which accomplished all the inclusion and exclusion criteria, it will be added to the final selection. Through this stage, 31 articles were added, creating a final selection of 123 papers. The detail of the amount of papers included/excluded on each stage is detailed in the figure 3.2. The selected papers are detailed by id in the tables 3.5, 3.6 and 3.7.

3.3.5 Data extraction

The table 3.8 details the items that were extracted from the selected papers. These extractions were intended to classify the content and order it aiming for a better understanding of the domain and helping for answering the RQs previously detailed.

3.3.6 Analysis and classification

The fields extracted for each selected article are tabulated and illustrated in section 3.4. These graphics represent the number of findings per type, inside each item. In some of these classifications, the groups are not exclusive, which means that the total number of classified articles could be bigger than the amount of obtained papers, since there are articles which will be present in more than one group.

The domains group is focused on describing the domain in which the proposal of the article is being designed, applied or evaluated. For those works which create a general proposal that is not related to any specific field, the domain related to the theory or the technology studied is selected, which in the case of this review, is the Semantic Web.

The purpose of the "models" item is to describe which kind of knowledge representation is being used in the article. RDF graph term will be used in those cases in which the model is not specifically stated, and instead there is a mention to RDF data or dataset and knowledge base.

The item contribution reflects the type of contributions that are present in the article. In other words, this item shows which are the problems in which the validation of semantic constraints on knowledge representation models is involved. Some examples would be optimization of queries, schema extraction, analysis and definition of CLs. These issues are present in the papers in quite an explicit way.

"Constraints groups" presents the set of constraints that are being represented (e.g. cardinality, integrity, etc.) and the language in which they are codified (e.g. SHACL, ShEx, etc.). If the set of constraints is not detailed, the work will be taken as including all the constraints belonging to the language used. Furthermore, if no language is used, the constraints will be classified as generic.

Finally, the characteristics evaluated represent the way in which the proposals are evaluated and measured. In this case, the types of evaluations are effectiveness, consistency, completeness, feasibility, scalability, performance, usefulness, accuracy, robustness, validity, ease of use, correctness and expressiveness. Most of these metrics are described in the taxonomy defined by Prat and Akoka ([121]), except expressiveness which refers to the variety of semantics that a language can represent.

Validity evaluation

As discussed by Petersen and Gencel in [134], the validation that should be applied to software engineering articles are the descriptive, theoretical, generalizability, interpretive and repeatability.

Descriptive validity

This validation is focused on detecting the accuracy and objectivity of the observations. In order to help this validation, the collection of the data related to each item is done following a data extraction form detailed in 3.8. This identification helps to control this threat. In addition, some clarifications related to the data extracted are detailed in section 3.3.6.

Theoretical validity

The theoretical validity evaluates the fact that the study is representing what the authors wanted to represent. This validation is influenced by different factors, as for example, the biases that could have placed on selected articles. One of the main threats is that, during the selection of the articles, it is likely that there are some interesting studies that are going to be missed. In an attempt to lower this possibility, the selection of the papers was complemented with a backward snowball sampling [90] of all the studies that appeared as references in the selected full-text read papers. However, the study selection was conducted by only one author, which is a threat to the validity.

Generalizability

This review is delimited by the inclusion and exclusion criteria to the computer science topics, and by the RQ to a specific type of constraints (semantic constraints) related to a specific type of models (knowledge representation). However, having this into account, the study could be generalized inside this delimited domain, since it covers different types of methodologies and models found on peer-reviewed articles.

Interpretive validity

This validation evaluates if the conclusions are reasonable having into account the data and the articles selected. A threat to this evaluation is the researcher bias. In order to respond to this, the study was done using mostly explicit information and the conclusions were formulated analyzing quantitative data derived from the classification.

Repeatability

To validate this, the review has to be explained and reported in detail. In this case, the SLR is explained step by step, following a structured process in a way that the search could be reproduced by the reader. However, a threat to this validation is the fact that the articles attached to the scientific databases could change, since there could be added or deleted articles (even setting an exact time span on the filters).

3.4 Results of the review

The selected papers are detailed in the tables 3.5, 3.6 and 3.7. In this section, the findings done as an outcome of the SLR are presented, and the defined RQs are answered.

3.4.1 Publications year and venue (RQ1 and RQ2)

The papers were grouped and counted by their year of publication, having articles from 2008 to 2023. The year with more publications is 2022, with 20 articles. The two following years with more activity in the domain are 2021 and 2019 with 15 articles each, meanwhile during 2020 and 2018, 13 works per year were made. Going forward, 10 articles were published in 2017, 9 in 2014 and 8 in 2015. Moreover, 4 articles were released in 2016, 3 articles in 2010 and 2009, meanwhile only 1 article was published in the years 2013, 2012, 2011 and 2008. Although the EC7 excludes articles published before 2001, the oldest article in the selected set is from 2008. The papers by year are represented in the figure 3.3, where the y axis represent the year of publication and

the x axis shows the amount of articles. Looking at the results, the first thing to mention is the continued activity that this topic has presented since 2008 until nowadays. Furthermore, it is interesting to highlight the growth in the published amount of articles starting at 2014. This is related to the publication of the popular CLs ShEx and SHACL, which were defined to validate the Semantic Web models ([97]). Finally, it is worth to mention that the database search was done in August 2023, and this means that is expected that the amount of published papers during this year will increase during the final months. The data extracted relating to the year of publication confirms the importance that the issue currently has, and ensures that this will continue being an active topic during the following years.

The figure 3.4 shows the venue distribution of the papers, where the x axis describes the types of venues found and the y axis represents the amount of articles. As shown in the picture, most of the papers were published in conferences (80), followed by 25 journal articles, 17 workshop papers and surprisingly, only 1 book chapter. With respect to the conference papers, they were published in 43 different conferences. The conference that appeared more times was the International Semantic Web Conference (ISWC) with 13 papers and the following one was the European Semantic Web Conference (ESWC) with 8 publications. Moreover, 5 of the selected papers were published in The Web Conference. Apart from the already mentioned, 4 other conferences were present with 3 articles per each, and 6 other conferences with 2 articles. The rest of the conferences (30) had only one apparition. With regard to the level of specification, some conferences were specifically related to a particular domain. As part of these, 9 conferences were related to databases as the International Conference on Database Theory (ICDT), 6 conferences were related to knowledge engineering as the Knowledge Capture Conference (K-CAP), 3 conferences were related to a specific model as the Knowledge Graphs and Semantic Web (KGSWC), 2 conferences were related to logic programming as the International Conference on Logic Programming (ICLP) and only the International Conference on Consuming Linked Data was related to the linked data. Moreover, 1 conference was related to Dublin Core and Metadata specific research. In contrary, some conferences were more general, as the 10 conferences found related to the Semantic Web (the already mentioned ISWC could be seen as an example), or the 3 Artificial Intelligence conferences found as the AAAI Conference and the 3 information systems related as the International Conference on Advanced Information Systems Engineering (ICAISE). Finally, 3 conferences were related to computing, 1 to data engineering and 1 to quality information.

3.4.2 Application domains (RQ3)

Most of the papers are related to the Semantic Web domain (88) and its technologies such as knowledge representation models, CLs and semantic constraints. Furthermore, 9 papers are related to the medical domain, positioning as the second domain with more papers, and being followed by works done in the architecture and construction field (7 articles). With less papers, industrial engineering (4), policy and compliance (3), internet of things (2) and social and economics science (2) works are found in the data set. Finally, the domains of geology, history, tourism, biology, social networks, git repositories and the World Wide Web are present in only one work per each of them. The detail of the percentage of papers per domains is detailed in the figure 3.5.

Having into account that the Semantic Web domain has more than the 70 percent of the papers, this domain is exploited following the "Tweaked Semantic Web Technology Layer Cake" categorization defined in [89] (see figure 3.6). The category of "Rules" was the one with more papers (62). Moreover, this category is divided into the following subcategories: "Schema Rules", involving those rules related to schema definition languages as SHACL (46), "Reasoning and

Figure 3.3: Papers by year

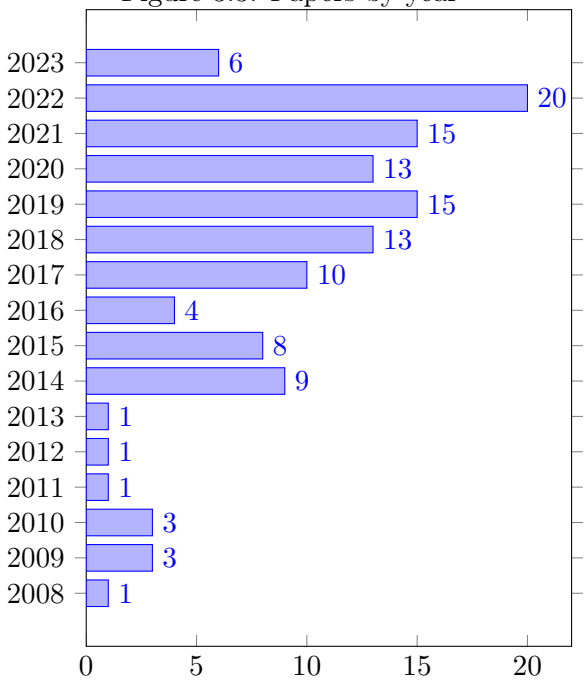


Figure 3.4: Papers by venue

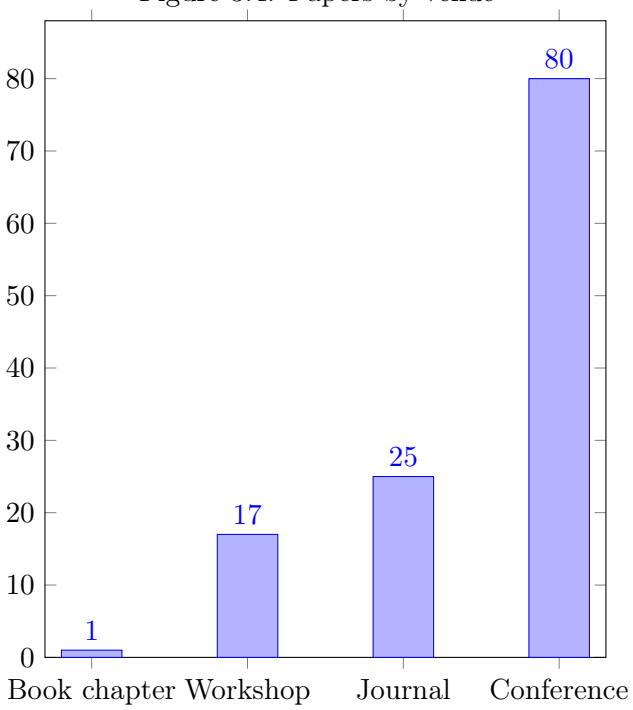
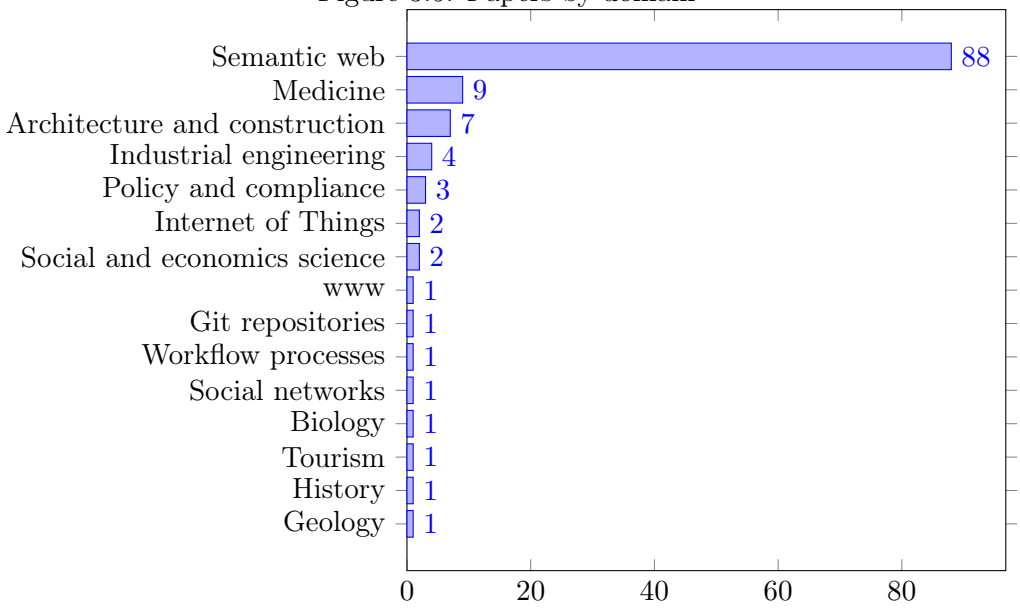


Figure 3.5: Papers by domain



Inference Rules" representing those works that are mainly focused on the discovery of implicit knowledge (9) and "Data Transformation Rules" related to papers where the main objective is the translation from one model to another (7). Going forward, the "Dictionaries" category had an important frequency with 20 papers, being a domain focused on a particular model as and RDF graph or an OWL ontology. Furthermore, those papers related to the study of queries were found 4 times and classified as "Query". Finally, only one article per category was found for "Trust" (privacy protection) and for "Smart Application and Services" (a smart MVC application). The rest of the categories described in the picture 3.6 were not used since no examples were found for them. The exploited "Semantic Web" domain is shown in picture 3.7.

3.4.3 Shared repository (RQ4)

In this RQ, the intention is to detect which of the selected articles share a link to a part of the work that is not included in the article. More specifically, the interest is posed in code repositories, but it is also accepted all the evidence related to the implementation and evaluation of the proposal. The inclusion of a shared repository could improve the quality of a scientific paper characteristics such as the reproducibility, the transparency, the adoption and the accessibility. Furthermore, the use of these repositories is aligned with the principles of the open science ([175], [123]).

The results show that the amount of papers that present additional resources is almost the same for the amount of those papers that do not present them. The authors found this as a consequence of those papers that present theoretical proposals, which are directly demonstrated inside the article, without the need of adding resources in a repository. Moreover, some of the articles present only an evaluation based on a proof of concept of their proposal, in which cases a repository has been neither often used. The distribution of the amount of papers that presented this extra evidence, and those which didn't, is detailed in the figure 3.8.

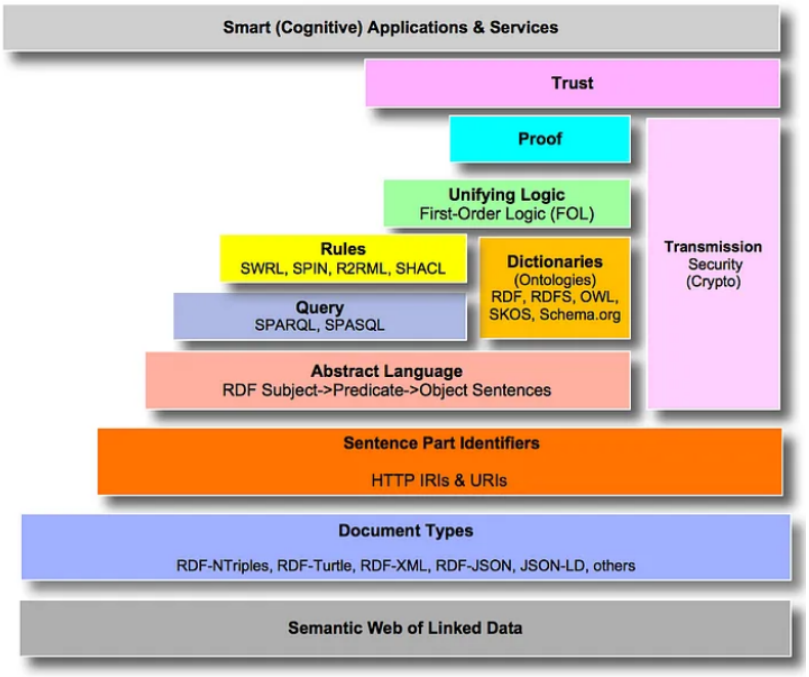


Figure 3.6: Tweaked Semantic Web Technology Layer Cake published by Idehen ([89])

Figure 3.7: Papers by domain - Semantic web exploited

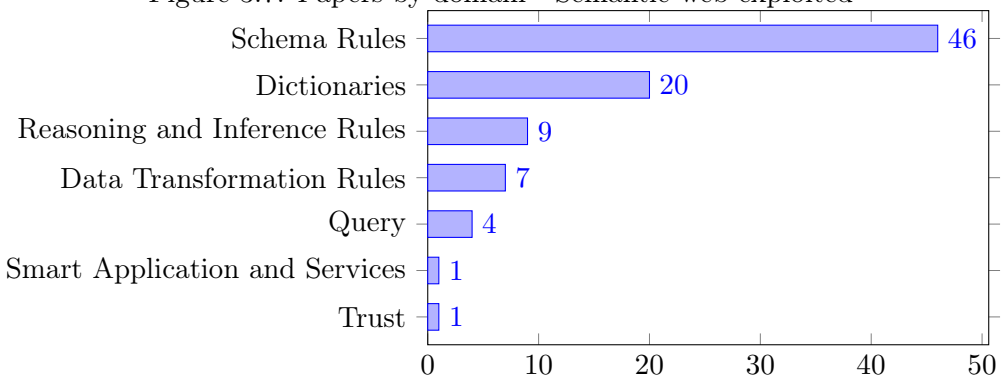


Figure 3.8: Papers by repository shared

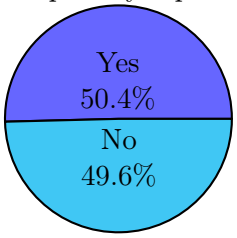
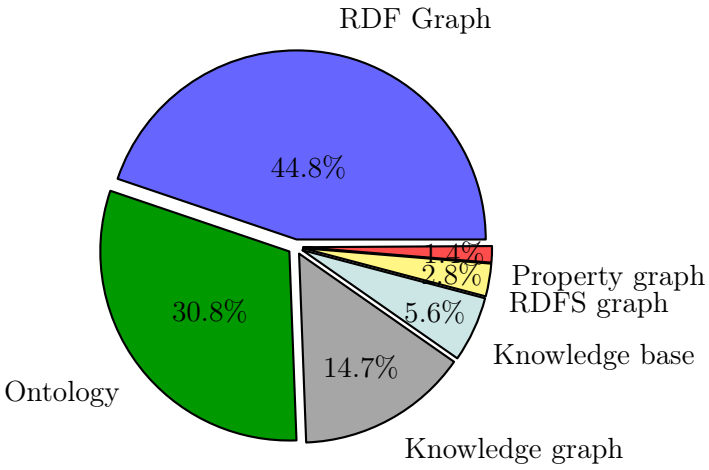


Figure 3.9: Papers by validated model



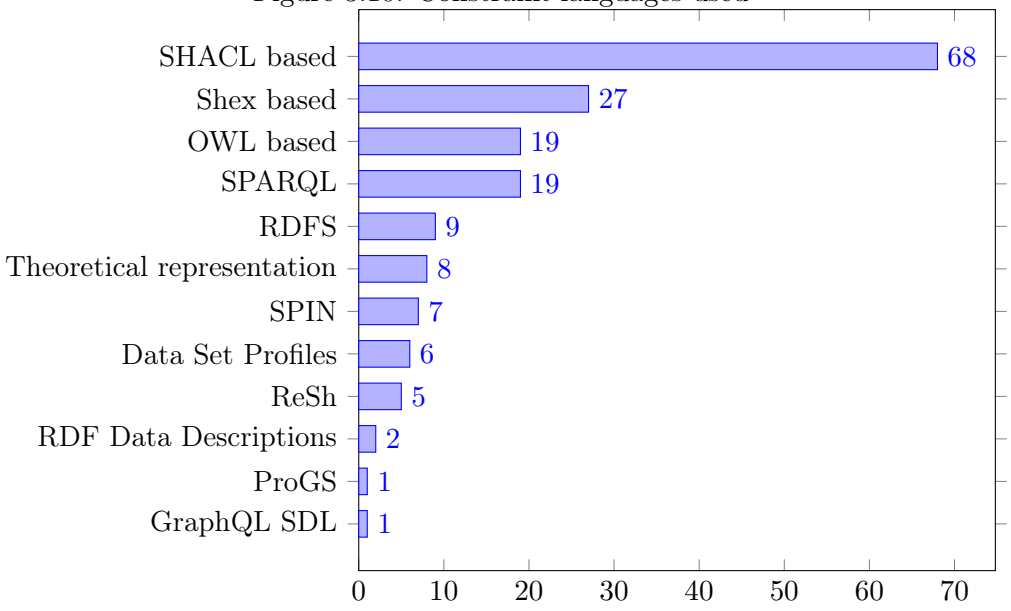
3.4.4 Knowledge representation models involved (RQ5)

This section presents the knowledge representation models that are being validated in the selected papers. The found models are the RDF graphs, ontologies, Knowledge Graphs (KGs), RDF/s graphs, knowledge bases and property graphs. The former four are already described in the section 2, meanwhile the latter two are briefly described in this section as follows. A knowledge base is a repository organized in a structured format (which could be an RDF graph based and also ontology based), that aims to store and manage domain knowledge. Moreover, this repository could be used for the inference of implicit knowledge, between other interesting features ([29]). With respect to the property graphs, this data model is used to represent and store information through a network formed by nodes, edges and key-value pairs which are related to both nodes and edges ([143]).

Almost the half of the models (64) was based on an RDF graph which represents RDF data or an RDF dataset, and were directly classified as an RDF graph. Not very far in the quantity (44), the second most used model is the ontology, which is followed by KGs (21) in the third place. Finally, knowledge bases (8), RDF/s graphs (4) and property graphs (2) had a minor presence in comparison with the already mentioned models. The detail of the knowledge representation models used is presented in the figure 3.9.

The high use of the RDF data model in the validation related works was expected, knowing that this is the standard model defined for the Semantic Web and that it represents mostly instance data, which creates an interesting scenario for validation research. Furthermore, the amount of papers that presented their work based on ontologies reflects on the one hand that the inference of these models doesn't ensure consistency, and on the other hand, that their expressive power is enough valuable for being on the priorities for further developments. Moreover, KGs are relatively new models, and their validation took on more importance from 2017. Since they represent mostly instance data, in order to validate them, it would be necessary to have an ontology behind, or to create a validation schema by the extraction of constraints from the graph. Going forward, the papers in which the input was a knowledge base presented these repositories mostly based on the RDF model. Finally, the low amount of published works dedicated to the validation of property graphs present evidence of a model that is still not mature with respect to the validation related tasks.

Figure 3.10: Constraint languages used



3.4.5 Semantic constraints and their representation (RQ7 and RQ8)

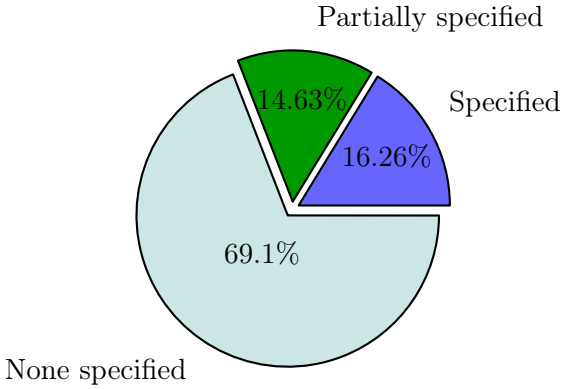
In the first part of this section, the analysis of the CLs that were used for representing the semantic constraints is presented. As expected, the most used language by far is the popular and standard constraint language SHACL, including subsets and extensions of the core language. It is followed by the ShEx constraint language, including extensions also in this case. Going forward and not very far from the latter one, the representations of Closed World Assumption (CWA) interpretations of the OWL family and SPARQL queries complete the top positions. The figure 3.10 shows how many times each CL has been used in the selected papers.

The second part of this section aims to describe which semantic constraints are being represented. The authors found that not all the works describe specific constraints, and sometimes, they work over all the constraints that are related to a determined CL, or simply, the authors do not specify if they are restricting their work to some specific constraints of the specific language used. In addition, there are proposals in which the constraints are partially specified. As an example of the latter, a work could be based on SHACL constraint language with negation (in this case, the negation constraint is being specified). The percentage of works which describe specific constraints is shown in the figure 3.11.

As shown in the chart, most of the selected papers do not specify the type of constraints with which they work, and instead, the proposal is based in a CL. However, since almost a third part of the set of papers is specifying totally or partially the constraints, it's worth to analyze what are the semantic constraints that are being highlighted.

Most of the semantic constraints are enumerated using as a reference the several constraint types defined in [82]. The *"Data Model Consistency"* constraints (also known as integrity constraints) are the semantic constraints that were used the most with 12 apparitions, what could be related to those works that were defined before the popularization of the CLs. This relation is made since, as shown in the set of selected papers, the first attempts to validate the consistency of knowledge representation models were based on the definition of integrity constraints that allowed to check the consistency by following the CWA interpretation. The *"Recursive"* constraints

Figure 3.11: Papers with specified constraints

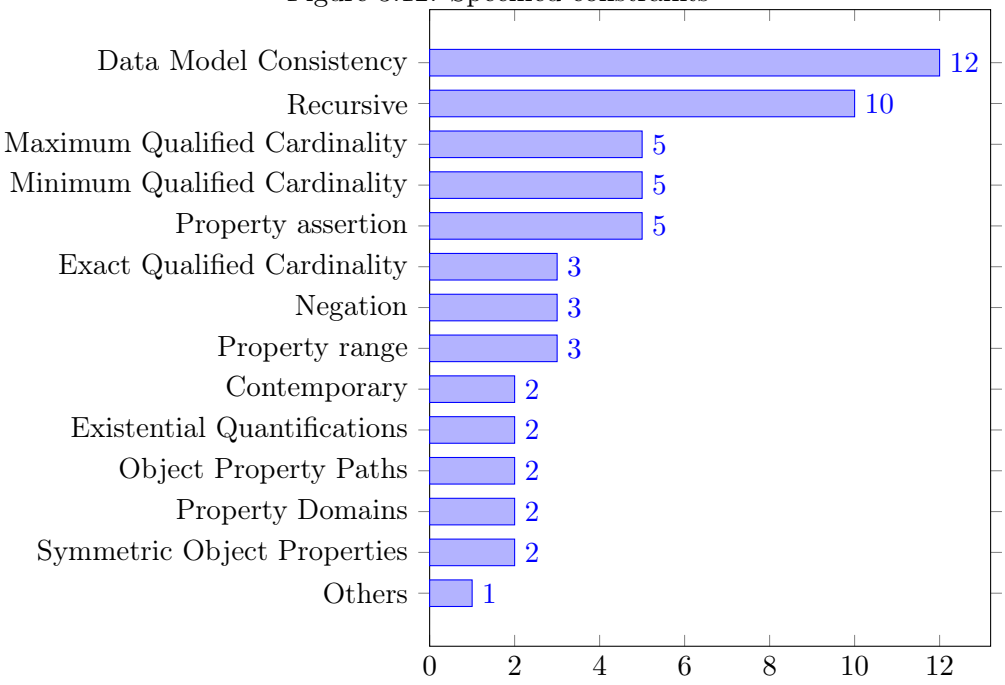


were the second type with more presence, which is partly a consequence of the lack of definition for them in the standard specification of SHACL. The podium is completed by constraints that have been treated in 5 different papers, which are *"Maximum Qualified Cardinality"*, *"Minimum Qualified Cardinality"* and *"Property Assertion"* restrictions. These constraints are frequently used for the definition of important schema characteristics. Going forward, 3 different articles presented proposals related to *"Exact Qualified Cardinality"*, *"Negation"* and *"Property range"* restrictions, meanwhile 2 examples were found for *"Existential Quantifications"*, *"Object Property Paths"*, *"Property Domains"*, *"Symmetric Object Properties"* and *"Contemporary"*. The latter constraint is not defined in the referenced classification, and instead, this constraint was defined in one of the selected articles ([4]). Furthermore, a list of constraints which were found only once in the selected papers is detailed as follows: constraints defined in the reference classification such as *"Allowed values"*, *"Asymmetric Object Properties"*, *"Class-Specific Property range"*, *"Context-Specific Property Groups"*, *"Disjoint classes"*, *"Disjoint properties"*, *"IRI Pattern Matching"*, *"Language Tag Matching"*, *"Literal Pattern Matching"*, *"Primary Key Properties"*, *"Subsumption"* and *"Universal Quantifications"*, *"Dependency"* constraint ([120]), *"Descriptive"* and "Prescriptive' constraints defined in [113], "User-Defined' restrictions ([76]) and finally constraints adapted to the property graph data model such as *"Comparison"*, *"Exact Qualified Cardinality"*, *"Maximum Qualified Cardinality"*, *"Minimum Qualified Cardinality Restrictions"*, *"Property specific key - edge path"*, *"Property specific key - node path"* and *"Recursive"* ([151]. The specified constraints and their distribution detail are represented in the figure 3.12. For a matter of simplicity and clarity, the semantic constraints with only 1 related papers will be comprehended in the category 'Others'.

3.4.6 Evaluated characteristics (RQ9)

The evaluated characteristics in the selected articles are counted in this section in order to answer this RQ. Moreover, as already stated in the section 3.3.6, almost all of these metrics are described in the taxonomy defined by Prat and Akoka ([121]), except the *"expressiveness"* metric, which is strongly related to this specific domain and refers to the variety of semantics that a language can represent. The *"performance"* was the characteristic most evaluated being present in 57 papers. Moreover, the *"accuracy"* was measured 47 times and the podium is completed by two characteristics that were evaluated 25 times, which are the *"validity"* and the *"feasibility"*. Furthermore, the presence of the rest of the evaluated characteristics was distributed as follows: *"completeness"* 14 times, *"consistency"* and *"scalability"* 10 times, *"effectiveness"* 9

Figure 3.12: Specified constraints



times, *"usefulness"* 8 times, *"ease of use"* 7 times, *"expressiveness"* 6 times and *"robustness"* 1 time. The distribution of all the evaluated characteristics is shown in the figure 3.13.

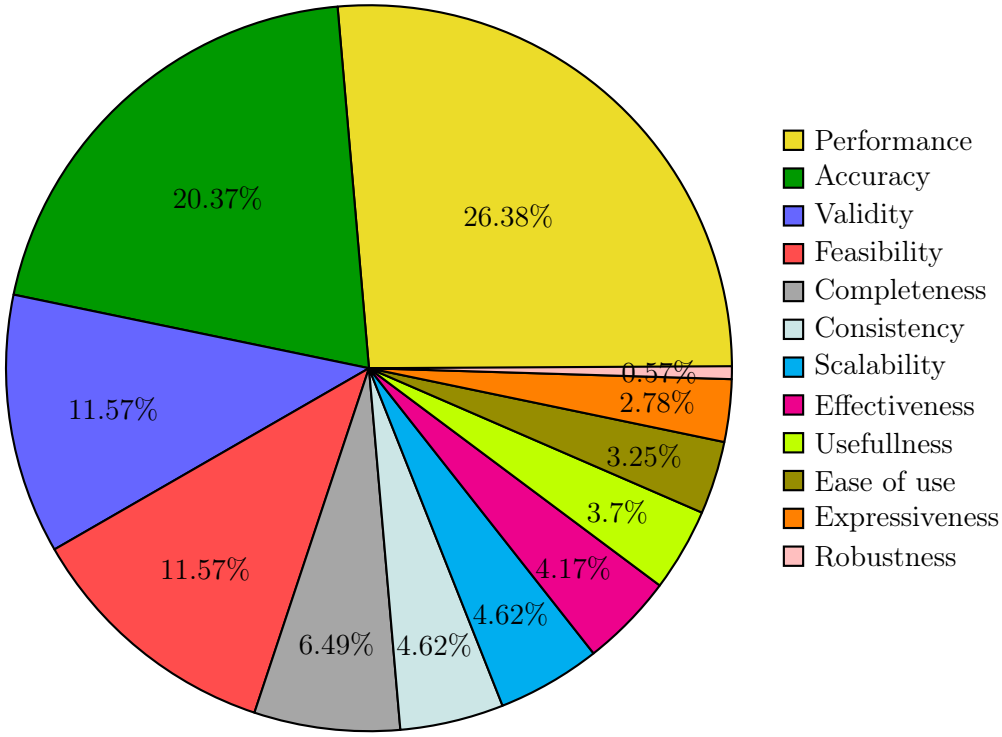
The performance is the most evaluated characteristic, since it is very important when it comes to work with big graphs and the validation of them. The high presence of the accuracy is related to the importance of checking that the designed proposal is doing what is supposed to be done. The validity evaluations are mostly related to the theoretical developments, meanwhile the feasibility is related to checking that the proposal is doable. The authors considered that the distributions of the characteristics evaluated reflect the exploratory nature of a domain that is in the path to go beyond the traced limitations.

3.5 Type of contribution taxonomy (RQ6)

The answer to this RQ is in a separated section because of the amount of the content that it presents and also because of the importance that the outcome has. In order to be able to answer this RQ, first, the definition of a taxonomy was done. This taxonomy represents the research trends found related to the validation of formal semantic constraints and can be seen in the figure 3.14.

The first category that can be seen in the taxonomy is **"Constraint languages" (CLs)**. This category is focused on this type of languages, already mentioned in the section 2.8 and presents the following subcategories: **"Analysis and Review"** represents the revision of a CL which was already defined in previous work and includes the comparison between existing languages. **"Semantic and Syntactic Definition"** refers to the definition of a new CL or to the extension of an existing one. **"Implementation"** includes the efforts for the development of an executable version based on a theoretically defined CL. **"Applied cases"** refers to the practical application of these languages as part of a bigger system, taking the role of validating data. **"Recursion"** includes the study of cyclic references in CLs since this was left undefined in the SHACL standard

Figure 3.13: Evaluated characteristics



definition. **"Satisfiability and Containment"** refers to the study of one or both characteristics related to this type of language, where the former evaluates the possibility of a data graph for being validated by a shape graph, meanwhile the latter refers to the definition of subsumption between shapes. **"Further characteristics"** includes all the CL features and problems that weren't included in the already mentioned categories.

Going forward, **"Mapping techniques"** refers to those works which develop one of these techniques in order to transform constraints which are represented in a non-semantic format into a semantic format. **"SPARQL constraint implementation"** refers to those works which use the query language for expressing constraints. **"Specific types of semantic constraints"** refers to studies that have their focus in the type of constraint they are representing. The following category is **"Schema extraction"** and includes those works which presents a technique for extracting constraints from an existing model. **"Visual constraints creation and validation"** comprehends those papers which aim to develop a visual interface for easing the manipulation of semantic constraints. **"Validation reports (explanation)"** includes those articles which focus on the outcome that the validation process returns to the users. **"Error treatment"** involves the analysis, prediction and correction of errors in formal models. **"Inference in validation"** represents those works which analyze the impact that these processes have in the validation (and vice-versa). **"Query optimization"** represents the efforts of the community for the improvement of the query engine performance, taking advantage of the validation of semantic constraints. **"Validation algorithms"** is represented in the definition of procedures that aim to check the consistency of a formal model against a set of semantic constraints. Finally, the category **"Other approaches"** includes those trends that weren't included in the already mentioned categories.

Furthermore, the classification of the selected papers by the taxonomy can be found in the table 3.9. In this table, the first column represents the titles of the orange categories presented

in the taxonomy, the second column presents the article ids (previously defined in table 3.5) that were classified as examples of the respective category, the third column presents the respective amount of articles and finally the fourth column contains the respective section in which the type of contribution is described. In addition, the amount of papers by research trend is shown in figure 3.15. In this bar graph, the x axis represents the amount of papers, meanwhile the y axis represents the title of the taxonomy's categories. It worth to mention that the classification is not exclusive, which means that one article could present works of more than one category. Each mentioned category is explained in more detail in the following subsections.

3.5.1 Constraint language analysis and review

This category represents those works which propose an analysis or review of one or more existing CLs. The group of selected papers presents six works that were classified into this category. There are some works which analyze a single CL, focusing on the most important characteristics and features that it offers as in [167] and [126]. Furthermore, other type of analysis that can be found is through a comparison between two CLs, or between a CL and another type of language. This can be seen in [21], where SHACL is analyzed through a comparison with Description Logics, meanwhile in [97] SHACL and ShEx languages are analyzed by doing a comparison between both of them. Finally, there are works in which the analysis is based on the comparison between more than two languages, and their ability for the support of certain requirements related to RDF validation ([169], [80]).

3.5.2 Constraint language semantic and syntactic definition

This type of work represents the definition of semantic and syntactic characteristics related to a CL. Fourteen of the selected papers were classified as part of this category. Some of these papers are based on the definition of a CL such as RDF Data Description language (RDD, [150], [62]), Shapes Expression language (ShEx, [137]) and Wikidata Shapes Expression language (WShex, [70]). On the other hand, an important amount of these papers is focused on the definition of a semantic extension over an existing CL, which normally consists in the addition of new features to the respective language. The extensions over ShEx are aiming to the definition of the 2.0. version and the support for single-type and multi-type features ([158], [23]). Meanwhile, the extensions to SHACL mostly aim to define new features such as recursion, and in some cases were inspired by other models such as logic programs, Answer Set Programming (ASP), and Approximation Fixpoint Theory (AFT) ([38], [13], [20], [33] and [22]). Another feature that started to capture the attention for the extensions in SHACL is the data provenance ([46]). Most of the aforementioned articles are explained in more detail in the categories which refer specifically to the CLs features related works. In addition, the totality of the already mentioned papers was related to RDF graph based CLs which were defined to constrain ontologies and KGs. Moreover, there is also a trend which aims to define the semantics and syntaxes of CLs which are defined to constrain property graphs, which are more expressive knowledge representation models ([79] and [151]).

3.5.3 Constraint language implementation

This type of work aims to the actual implementation of the semantics and syntaxes of a CL. Eight papers from the SLR outcome were classified into this category. An implementation of ShEx was found, having the inspiration in regular expression derivatives, meanwhile a review over four previously defined ShEx implementations was also presented ([69] and [137]). The

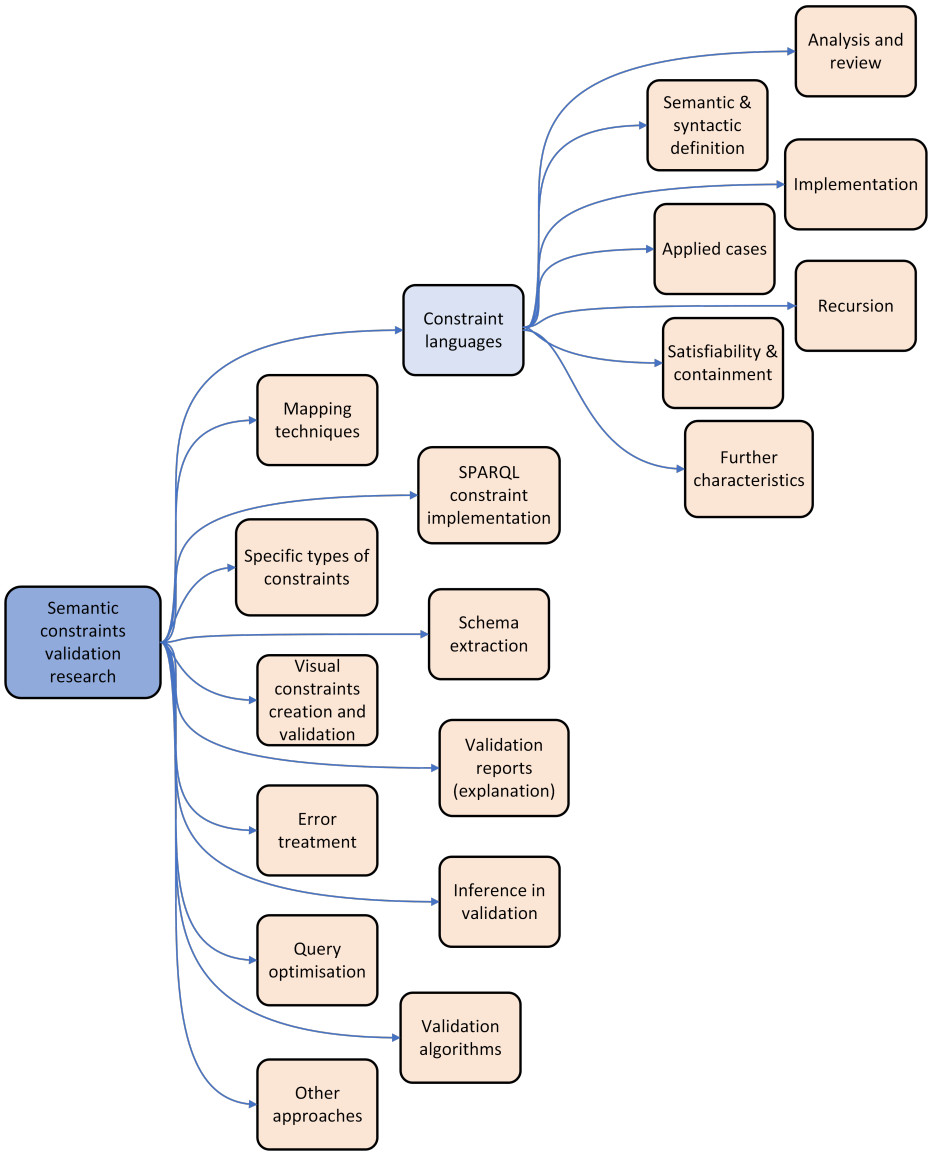
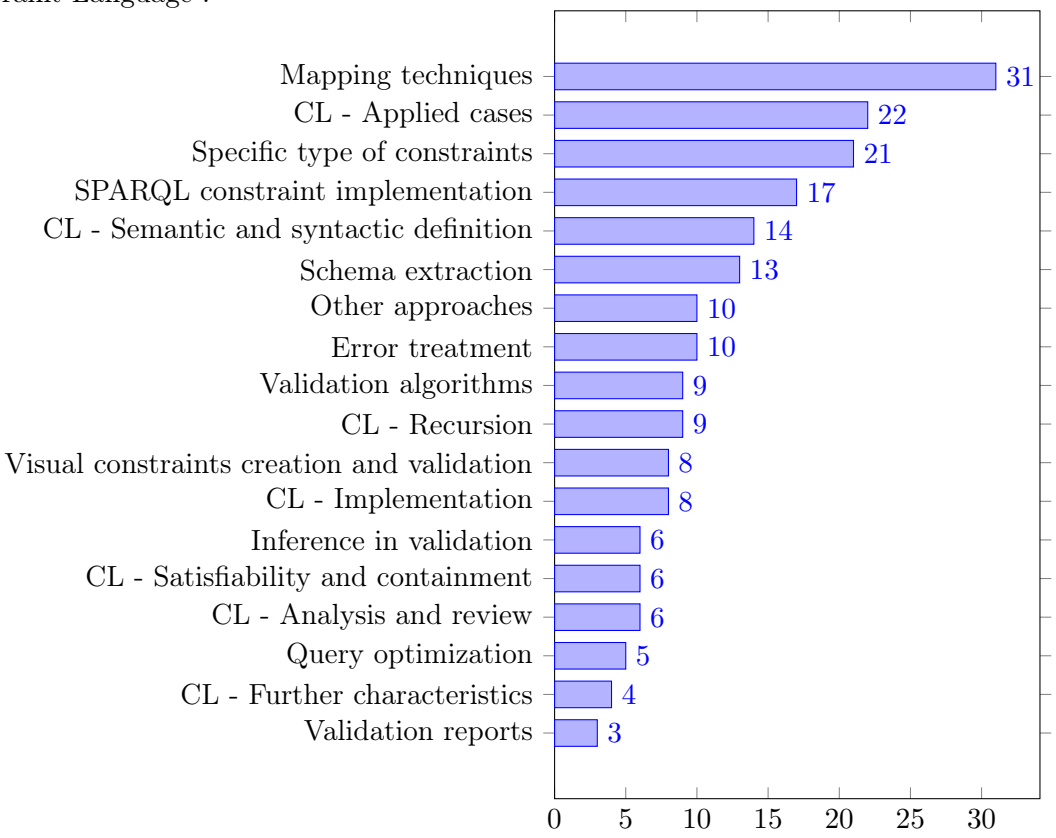


Figure 3.14: Taxonomy of research trends related to validation of formal semantic constraints

Figure 3.15: Distribution of the selected papers by their type of contribution. CL stands for "Constraint Language".



implementation of SHACL is another objective of the community, which could aim to avoid the validation of invalid entities ([61]) and the computation of shape fragments ([46]). In addition, the definition of a benchmark that allows to test existing SHACL implementations in specific graph databases, is found quite necessary, and only one proposal was found up to date ([148]). Furthermore, the SPARQL query language could be used for the implementation of CLs, having already an implementation version of RDD and Description Set Profiles ([131] and [26]). Finally, ASP could be used for the implementation of CLs which aims to validate property graphs ([151]).

3.5.4 Constraint language applied cases

Since the definition of CLs that follows the concept of shapes graphs, such as ShEx and SHACL, several models that are represented in Semantic Web technologies are validated by them. The authors found several works that presented the validation of a knowledge representation model through the application of a shape CL, which allows to validate it following CWA and UNA configurations. ShEx could be used to describe and validate existing models related to a determined domain (such as medicine) or even to validate linked data portals ([167], [99]). In the same way, SHACL is often employed for similar objectives, such as the definition of constraints and the posterior validation of domain related ontologies ([77], [88], [114], [125], [91], [160], [96], [55], [30]). SHACL is also used for the validation of other models such as domain related KGs ([157], [180], [162], [66]), already published shared vocabularies ([168]) and even systems that are not initially semantically structured on a model, such as Github repositories ([112]). In addition, SHACL validation is applied to accomplish specific functionalities such as resource access management ([174], [163]), user access authorization ([30]), product configurations ([75], [149]) and compliance checking ([125], [142]). The important amount of papers (22) inside this classification shows the high acceptance that shapes graphs (mostly represented in SHACL) are experiencing for practical applications over heterogeneous case studies.

3.5.5 Constraint language recursion

As mentioned before, SHACL is the CL defined as the standard by the W3C. However, in the definition of the language semantics, there are some characteristics that were left open for being defined and developed in the implementations. One of these missing definitions is the recursive shapes, which describe a reference cycle between shapes. The design of SHACL creates a favorable environment for this type of constraints which reference another constraint ([38]), and for this reason, this gap has awoken the interest of the community, and some trends have appeared. The definition of recursive semantics for SHACL shapes was presented in some papers, inspired by different concepts, such as the Approximation Fixpoint Theory (AFT), Second order logic, logic programs and ASP ([38], [20], [127], [33], [13]). Furthermore, a general analysis of the recursion in SHACL could be found ([126]), as well as more specific analysis related to specific features such as the satisfiability or disjointment of SHACL in the presence of recursion ([127], [22]). Finally, the study of the validation task over recursive shapes have place and comes along with the proposal of new algorithms that are able to perform the task ([37], [8]). A total of 9 papers from the set of selected articles was classified as part of this category.

3.5.6 Constraint language satisfiability and containment

Some works (5 papers from the selected group) have dedicated special efforts to the analysis of these two particular decision problems of CLs. Given a particular SHACL document, satisfiability is the problem of deciding whether there is an RDF graph which is validated by the

document. Meanwhile, containment studies whether a particular SHACL document is subsumed by a second one; that is, whether all graphs that are validated by the first are also validated by the second. The study of both characteristics opens interesting paths for the improvement of the state-of-the-art validation techniques. Some proposals were presented for the study of one of these problems. A sound and complete procedure for containment in ShEx, considering several SPARQL fragments, is presented in [2], meanwhile the same problem is studied in ShEx from a theoretical point of view in [159]. Furthermore, the analysis of satisfiability and containment in SHACL could be done through a previous translation to first order logic ([128]) or to a fragment of monadic second order logic which includes the recursive fragments ([127]). Moreover, the decidability of shapes containment in SHACL has been studied through a rewriting to description logics with the use of subsumption [103]. Finally, a checkpoint of the work done with respect to SHACL is presented as a review in [126].

3.5.7 Constraint language further characteristics

The study of the CL characteristics includes more than recursion, containment and satisfiability. In this category, those studies that analyzed other features apart from the mentioned ones are presented, which represent 4 papers from the total selected ones. One characteristic that is taken into account is the SHACL arbitrary and unrestricted negation including the study of the corresponding semantics and the validation of shapes graphs which includes this type of semantic ([38], [8]). Furthermore, the analysis of the SHACL semantics related to other four features is presented in one specific work, aiming to analyze their impact in the expressiveness of the language. These features are the following: equality, disjointness, closure constraints and zero-or-one paths ([22]). Finally, the decidability over filter features, and a proposal for their effective treatment is presented in [127].

3.5.8 Mapping techniques

This category includes those papers in which a mapping technique is applied, which means a syntactic and semantic translation from a source language to the most possible equivalent representation in another target language. Only those works in which the source language and/or the target language is able to represent formal semantic constraints are taken into account.

An important amount of these works is dedicated to mapping a source language into a generic target language (which could be even a new language) and in most of these cases, the real objective is to analyze theoretical characteristics of the source language or generate the rewriting the most possible generic way. SHACL is involved in many related works, and it was mapped to generic languages such as new languages of first order logic and second order logic ([127]), Description logic ([103], [21]) and logic programming ([33]). Furthermore, both SHACL and ShEx were mapped to a minimal and new intermediate language to facilitate the comparison between them ([97]). Moreover, an idea of generalization based on the transformation of a generic high level language to a generic CL was presented in [81], with the objective of paving the way for the transformations independently of the specific source language. Finally, the transformation of Structured Query Language (SQL) data to generic shapes is presented in [24].

Another trend well established has to be with the integration of the data represented in RDBs to the Semantic Web technologies, and the possibility to validate it through shapes graphs. This means to find an equivalence between the SQL constraints and their representation as shape constraints. There are different approaches to this kind of mapping, such as the transformation of SQL constraints to a generic representation of shapes ([24]), or the mapping from the SQL

schema to the target standard shape CL SHACL ([165], [166]). Furthermore, taking into account that RML is a rule language that is used to map SQL data to RDF, an approach for mapping the constraints directly from these rules to SHACL shapes was also presented ([45]).

Going forward, many works have proposed mapping techniques for determining equivalences between shape constraints and the SPARQL query language. Representing the constraints with the latter allows to have an executable version, which will be explained in more detailed in the category "SPARQL constraint implementation". Going to the approaches, the mapping could be done from a generic CL to SPIN-SPARQL as an attempt to generalize this kind of translation ([81]). In addition, some techniques were presented for mapping specific constraints represented in a language to their respective representation in SPARQL queries, having as the source languages such as SHACL ([126]), RDD ([62]), Description Set Profiles ([26]), Wikidata property constraints ([59]) and a combination of OWL language with Semantic Web Rule Language (SWRL) rules ([153]). Moreover, the mapping could be done in the opposite way, having SPARQL as the source and a CL such as SHACL as the target ([104]).

Another type of mapping that is often used is between a CL such as SHACL and the ASP semantics, aiming to have the possibility of analyzing the constraints on the well-founded semantics of the target ([149], [13], [7]). The mapping could be also done having a Semantic Web model as a source, such as RDF KGs ([155], [119], [52]), OWL family ontologies ([136], [147], [34], [160]) or even the specific Wikidata model ([59]).

There are still more source types which are mapped to SHACL such as Object Constraint Language (OCL, [100]), DECLARE templates (a well-established declarative process specification language, [47]), and even trained decision trees ([52]) or semantic profiles generated by the ABSTAT tool ([156]). Finally, there are also two more source types which are mapped to ShEx constraints, which are XML Schema ([68]) and CSV ([146]).

Most of these techniques ease the interoperability and integration between heterogeneous models and the Semantic Web. The important amount of works classified in this category (31 papers), could be interpreted as the popularity of the Semantic Web technologies, and how many communities present a tendency to move to the latter side.

3.5.9 SPARQL constraint implementation

An important amount of papers (17) presented the validation of Semantic Web models through the design and execution of SPARQL queries and its use as a CL. It's worth to mention that this category does not include those implementations which merely describes a specific CL such as SHACL. The motivation for the use of SPARQL queries in order to check semantic constraints with a closed interpretation is related to the importance and prevalence that this language has had since the beginning of the Semantic Web and also to the variety of constraints that can be expressed with this language. SPARQL constraints could be defined and implemented to validate a specific domain ontology ([120], [55]), but there are also some proposals which define the queries in a more generic way with the intention of being reusable for validating several ontologies and to reflect theoretical extensions ([154], [164], [95], [80], [113], [153]). SPARQL queries could also be related to KGs, and the objectives could be the validation of a specific KG such as Wikidata ([170], [59]), the definition of queries in a reusable generic way ([101], [131]) and even for determining semantically correct property value assertions of instance-property pairs ([48], [49]). Finally, the definition and implementation of the SPARQL queries are also done through the use of the SPARQL Inferencing Notation (SPIN) framework, and its applied with different already mentioned objectives such as the validation of ontologies ([65], [81]) and KGs ([83]) in a generic way.

3.5.10 Definition of specific types of constraints

An interesting amount of papers (21) presented in their work, an analysis and/or definition of the expressiveness of specific semantic constraints and their validation characteristics. This category puts the focus mainly in the type of the semantic constraints, their semantics and their expressiveness. These works could be presented in different ways, and to start describing one of them, there exist a type of work in which the attention is focused in a few semantic constraints (less than 10) and where the definition of these constraints is done independently of the data ([101], [154], [164], [10], [120], [153], [113], [4], [122], [140], [35]). Another type of work that could be found is also related to the definition of a few constraints, but in contrary to the latter mentioned, in this case, the constraints are defined through an extraction of patterns from the data ([117], [118], [119], [59]). Finally, there are works in which the analysis or definition of several specific constraints is presented, and their support by the existing constraint representation approaches and frameworks are reviewed ([80], [81], [84], [83], [169]). These kind of works shows how linked are the validation of knowledge formalization models and the definition of semantic constraints. Furthermore, these works correspond to the attempt of being able to represent, step by step, more sophisticated restrictions, allowing the validation in a more efficient way.

3.5.11 Schema extraction

Schema extraction includes those works which proposed a technique for the extraction of the schema (or a part of it) that is implicitly present in a knowledge representation model. This extraction is done to obtain a screenshot of the schema at one desired point in time, which will then be used for the validation of the model, between other interesting functionalities, such as the supervising the evolution of the model. Having the possibility of extracting these validating schemas in an automatic way has an interesting potential, since the manual creation of these is a time-consuming and error-prone task.

These techniques could vary in several ways, such as the model or structure from which the constraints are extracted, the format or model in which those constraints are then represented, the level of automation of the extraction, the way in which the algorithm is designed, the amount of constraints that is extracted, between others. 13 papers out of the set of selected papers presented works related to these techniques.

To start with the enumeration of the known approaches, those techniques which extract constraints from RDF graphs are described. A technique for extracting ShEx shapes (and also OWL axioms) is based in a semi-automatic process which exploits structural statistics that then are used in a simplified process ([41]). In addition, a similar technique for extracting SHACL shapes was explained in [136]. Furthermore, the need for the development of an automatic technique for the extraction of specific structural semantic constraints was already analyzed and proposed in [117]. There are also techniques which extract automatically a few specific constraints from these graphs and represent them as ShEx shapes, through a simulation-based approach ([73]). Finally, having as the origin a domain-specific RDF graph, a semi-automatic and database-driven technique for the extraction of SHACL shapes is presented in [145].

Moreover, the automatic extraction of certain constraints could also be done from a knowledge base. An automatic extraction of cardinality constraints represented in ShEx was presented, based on mining the relation cardinality bounds by prior normalizing and filtering the data ([118], [119]).

Going forward, the constraints could be also extracted from ontologies. SHACL shapes could be obtained from a domain-specific ontology as in [160]. Moreover, the extraction could be

generalized to any ontology as presented in [34], where the technique consists in the execution of mappings between ontology and SHACL patterns previously loaded in a KG.

Finally, the constraints could be also extracted from KGs, and some techniques reflect this possibility. A technique for the automatic induction of generic RDF shapes graphs based on machine learning and data profiling was presented in [116]. Furthermore, SHACL shapes could be extracted from KGs through the design of an algorithm based on approximation and summarization techniques ([139]). In addition, an automatic extraction of SHACL and ShEx shapes based on mining the graph structure was presented in ([58]), offering as well the possibility for inter-linkage between the generated shapes. Finally, a proposal for the automatic extraction of SHACL shapes based on Inverse Open Paths was presented by Omran et al. ([124]).

The presence and development of these sophisticated and complex techniques represent how valuable the validation of knowledge representation models currently is. In addition, it shows that it is becoming a good practice on the Semantic Web, to be able to validate models that were already published.

3.5.12 Visual constraints creation and validation

The representation and manipulation of semantic constraints require background knowledge related to the semantic and syntax of specific CLs designed for these purposes. In order to increase the adoption of the Semantic Web technologies and the existing languages designed for validation purposes, some papers (8) present the design and development of an intermediate graphic layer between the user and the language in order to ease its use. There exist web interfaces which were designed for the creation and edition of SHACL shapes aiming to ease the validation of previously loaded ontologies ([138], [152]). Moreover, a similar tool was presented in [176] as a form based web application which allows the creation and edition of SHACL shapes, but in this case aiming to validate KGs instead. Going forward and having into account the popularity of the Protegé tool for the creation of knowledge representation models, a plug-in for the creation and the execution of these kind of shapes was presented in [53]. Furthermore, there exist tools which aim to ease the creation of ShEx shapes, such as ShExStatements, that proposes using a tabular format or CSV file format for easing the task ([146]), and Validata, a web tool which is able to validate an RDF graph against these constraints, allowing in addition to define different validation requirement levels ([78]). Finally, there are works which are not only proposing visual tools for the creation and edition of SHACL shapes, but also defines baselines as an attempt to standardize the development of these kind of tools, such as the desired features that they should cover ([44]) and visual notations that could represent efficiently each kind of constraint ([106]).

3.5.13 Validation reports

The validation of shape constraints generates as an outcome a report with some details related to what happened during the process. The deepness and level of detail of these reports depend on the specific implementation, and presents an important potential for post-validation actions. The study of these reports was found only in 3 works between the selected articles. A rules-based reasoning technique which gives the possibility for extending validation reports with more detailed root-cause explanations, and having an accurate number of violations was presented in [43]. Moreover, the problem of the explanation of non-validation of constraints was formalized in [6], which also proposed presenting a set of RDF graph updates in the report, as a guide to repair the violations found. Another type of extension found was based on adding probabilistic information in the reports, expressing a set of metrics including the generality and likelihood of

shapes ([56]).

3.5.14 Error treatment

The validation of semantic constraints is also involved in error detection, error correction and error prevention activities over knowledge representation models. There is evidence of this relation in 10 of the selected papers. These works are aimed to improve the quality, maintenance and evolution of knowledge representation models through the exploitation of the semantic constraints represented.

Some techniques are exclusively dedicated to error detection, as in [63], where a technique for detecting erroneous numerical values in a linked data repository is presented, paying special attention to the choice of sub populations of values and cross-checking by means of a second set of data. Another proposal for error detection in linked data is based in an automatic test instantiation based on schema constraints or semi-automatically enriched schemata, allowing users to generate their specific tests ([95]). In addition, the error detection could be also focused in relation assertions of a KG, with a technique based on a combination of path and type usage ([115]).

Going forward, there exist techniques that not only detect the errors, but also propose alternatives for the correction of the found inconsistencies. The last mentioned technique for error detection has an extension in which it includes the automatic correction of the errors, relying on type predictors and on a classifier for every relation in the graph, exploiting local feature selection ([52]). Another work presented a proposal which begins from a detail of violations presented in a SHACL validation report, codifies the problem in ASP and generates the corresponding repairs ([7]). Furthermore, these repairs could be also presented directly in the validation report, constructing the explanation using the notion of a repair as a collection of additions and deletions whose application results in a repaired graph that does satisfy the given SHACL constraints ([6]).

Finally, there exist techniques which aim to the prevention of the errors. In [179], the presented technique first finds the key instances of the data graph, then detects possible inconsistencies that could be generated after the inference of the model and finally provides the corresponding repairs to avoid these errors. A similar technique is applied when updating RDF graphs, avoiding the identified possible errors through additional updates represented as blank nodes ([76]). Moreover, the error prevention could also be done in KGs through an initial evaluation of the semantically correct property value assertions, which then is used to provide a list of acceptable statements to the users. In this way, the errors are directly avoided before they are actually materialized ([48], [49]).

3.5.15 Inference in validation

The inference of semantic constraints could be applied in knowledge representation models with goals such as analyzing the validation procedure, improving it or even creating a new one. The selected articles include 6 proposals which involve and relate the inference process with the validation one.

Having an RDF graph and a set of semantic constraints, an interesting research consisted in analyzing the validation process by first validating the graph as it is, and second, applying a prior inference process in order to include the implicit knowledge in a new validation of the model. The comparison between the two scenarios could include the quality of the generated data, the performance of the validation task, and the dependence of the results with the underlying semantics ([25]).

A similar analysis was presented, having an RDF graph which is consistent with a defined set of constraints as the initial condition. In these cases, the idea is to apply inference rules to the model, and then validate it a second time against the same set of constraints, in order to determine if the consistency holds. Moreover, if the second validation returns violations, a repair is possible, either deleting the conflicting inference rules or modifying the set of constraints ([129], [179]).

The utility of the inference process has been also researched specifically with respect to the explanation of non-validation results in SHACL. In this technique, the reasoning procedures are defined to decide if there are possible explanations, to find the best candidate and to decide if an explanation is relevant as an addition or deletion to the graph ([6]).

Furthermore, rule-based inferences were presented as the core of a new validation approach proposal in [43]. Taking advantage of the customizable inferencing steps, this technique explains the root cause violations in detail and generates an accurate number of the inconsistencies found.

Finally, an approach to inference shapes graphs based on grammatical inference was presented in [73]. This technique proposes a simulation relation on the nodes of the input graph for identifying subtypes, and presents as an outcome, a schema that could be used to validate the input graph.

3.5.16 Validation of semantic constraints for query optimization

This category describes works which develop techniques for query optimization. These techniques, in some part of their procedure, are related to a validation or satisfiability check of a query (or query results) against a schema (shape graph) or a group of constraints (defined with SPARQL queries). The selected group of papers includes 5 related techniques.

The optimization of the queries could be faced from the construction of a CL. Following this, a semantic extension to RDF was presented in [101], which includes specific types of constraints that are able to represent general constraints and key information from RDBs. This approach aims to minimize the lost information when mapping from RDBs to the extended RDF model, and allows to semantically optimize SPARQL queries based on constraints-implied knowledge. Furthermore, there is another related approach ([62]) in which the RDD constraint language is involved and the validation process through SPARQL queries is studied. Starting from this point and aiming to improve the performance of the constraint checker, the query optimization is treated in different ways, such as intra-query (individual query), intra-constraint (individual constraint) and inter-constraint (set of constraints) processes.

Going forward, the optimization process could aim to reduce the execution cost of the SPARQL queries by finding the best possible order execution. Based in a first validation of the basic graph patterns against a previously defined ShEx schema, the subjects of the triple patterns are linked to the shapes to which they could belong. This relation aims to rank the patterns depending on how many shapes are attached and then use this ranking for the definition of the order execution ([1], [3]).

Finally, an approach proposed to improve the performance of SPARQL queries through an analysis of their containment relation, having into account a previously defined ShEx schema. The containment relation between the queries is computed before their execution in the data. The procedure is based first on ShEx validators to compute the containment between two queries taking into account the schema and second, in query containment solvers to check the outcome of the first step without taking into account the ShEx constraints ([2]).

3.5.17 Validation algorithms

This category describes those works (9 from the selected papers) which present or analyses an algorithm for the validation of a data graph against a set of constraints. The validation process is still nowadays a very challenging task, and more related proposals are expected in the following years.

The authors in [38] presented an algorithm for the validation of an RDF graph against a set of SHACL shapes which consists in two steps. First, a minimal fixed point is calculated, focusing on all the constraints enforced by the graph. Second, if the validation hasn't been defined yet, an extension is done by assigning the shapes to a node and its successors. Furthermore, this work has been extended with two algorithms prepared for being executed in a SPARQL endpoint ([37]). The first one uses the answers of a set of SPARQL queries to build a set of propositional formulas that are processed by an Satisfiability (SAT) solver. Meanwhile, the second one performs the necessary inference on the fly and works for recursive, but tractable fragments of SHACL. Both algorithms handle validation for total or partial assignments. Going forward with SHACL shapes, an algorithm for an incremental validation of a KG which switches between data retrieving and constraint validation was presented in [61]. In this work, the proposal selects the traversal shape plans and rewrites the target and constraint queries to the fast detection of invalid entities, allowing to reduce the number of constraints that need to be checked. In addition, an attempt to validate full SHACL was based on Magic Sets (from logic programming), aiming to validate only the relevant neighborhood of the targeted nodes. In case the hard validation with the original schema fails, a lazy and target-oriented validation is done, taking into account a transformed "magic" version of the same schema ([8]).

This kind of works was also based on the validation of ShEx schemas. Four different ShEx implementations are mentioned and analyzed with their respective validation algorithms in [137]. Furthermore, a proposal for partial validation (only a part of the graph) was done for multi-type and single-type semantics. Based in the construction of a pre-typing and its minimal valid extension, these algorithms are a modified graph flooding example that verifies that the nodes satisfy their respective types. In addition, for single-type semantics, the process prevents assigning two different types to the same node ([158]). Going forward, a derivative algorithm designed for checking the same sources was presented in [69]. In this case, once a regular shape expression matches a set of triples, the derivatives are computed for each of them, without the necessity for decompose the graph or do backtracking. Finally, two algorithms were presented in [23], having the refinement algorithm that calculates the maximal typing defined as the union of all correct typings by iteratively removing unsatisfied node-type associations until a fixed point is reached. The second presented algorithm is recursive and aims to visit only a sufficiently large portion of the defined typings.

A different approach based on a previously built ontology of semantic constraints was presented in [108]. A set of flags defined in the ontology, in addition to a set of SWRL rules, is exploited by the algorithm in order to complete the respective rules and to control the validation over the flagged instances.

3.5.18 Other approaches

This category includes approaches that are different from the ones presented in the so far mentioned categories. For simplicity, these were grouped into this final category.

In order to start with their description, a particular approach was presented in [177], which defines a framework for using ontological constraints from OWL-S (OWL for services) and OWL

for creating semantic test cases in order to check the robustness of a web service.

Moreover, an approach for the validation of RDF datasets based on the exploitation of a set of OWL ontologies and SWRL verification rules was presented in [108]. In addition to the domain ontology related to the specific dataset, this proposal is based on three more ontologies which describe generic constraints related to classes and individuals. If the validation algorithm finds a failure, semantic constraints are derived from the mentioned rules allowing to identify the specific conflicts on the dataset. This approach maintains the OWA in the modelling and validates the dataset following a CWA. A similar objective was aimed through a different proposal where the extension of certain concepts and roles are marked as complete by defining them as DBox predicates ([132]). In this case, the marked concepts are able to be validated as integrity constraints, meanwhile the non-marked ones are treated as incomplete.

Going forward, the relationship between an RDF data graph and a structure is evaluated in [14] by the definition of a framework where a user is able to define structuredness functions and validate how much the graph accomplishes with a schema structure. In addition, a related approach was based on a set of metrics and the concepts of type and class profiles ([92]).

Next, a method for updating RDF databases (OWA) with the characteristics of a relational database (CWA) through the use blank nodes for adding necessary triples in order to keep the consistency was presented in [76].

Another approach presents an analysis of the completeness for evolving knowledge bases using data profiling features. The outputs of the data profiling are then used as features in learning models that aim to predict correct integrity constraints which are then used to detect consistency issues in the knowledge base ([140]). The formal knowledge has been also involved with machine learning for the assurance of consistent predictions. This was proposed in [141], where the logical constraints deduced from ontologies are utilized in order to enhance and control the learning task by enforcing description logic satisfiability in a latent multi-relational graphical model.

Finally, proposals for checking the consistency in Ontology-Based Data Access systems (OBDA) and in Ontology-Based Data Management systems (OBDM) were defined by the respective formal definition of a framework that includes integrity constraints ([122], [35]).

3.6 Discussion

In this section, the relation between the different results presented in the previous section is analyzed and interpreted. In addition, the research gaps found in the results are described.

3.6.1 Integrated results

In this section, the obtained results will be analyzed in an integrated way by crossing the results in order to find and interpret the main intersections. The type of applications described in section 3.5 are quite interesting for the contribution of this paper. In consequence, this section starts by analyzing the intersection of the type of contribution with the publication year. The current trends will be defined in relation to the percentage of papers of the category that were published during the last six years (from 2023 up to 2018 included). Those categories which have more than the half of the papers published during this period of time are selected as trends. In addition, the percentage has to represent at least 5 papers.

The *"Constraint languages - Applied cases"* category has a 90 percent of their members in the period defined. This is interpreted as the result of the standardization of SHACL shapes in 2017 and the definitive adoption of shapes graph for the validation of systems which are based on knowledge representation models. Furthermore, at least one related work per year was published

for the last 7 years, which means that there is a constant utilization of these technologies and techniques.

A 100 percent of the works related to the recursion in shapes graphs (*"Constraint languages - Recursion"* category) were presented during this period of time. As mentioned in the section 3.5.5, these works look for filling the gap left explicitly by the standard SHACL specification. This trend reflects the intention and the effort of the community in extending the features and expressiveness of shapes CLs. Furthermore, at least one related work per year was released since the SHACL specification in 2017.

As another example of the efforts in extending the capabilities of shapes graphs, the 83 percent of the works that treat *satisfiability and containment* of shapes was published during the last 6 years. This trend is also active since the creation of SHACL.

A bit above the 80 percent of the works which present a *mapping technique* inside their contributions was published in the period of time defined. Taking into account the important amount of papers in this category, and the constant contribution in the last 6 years, the rewriting of constraints with validation purposes represents a trending topic. This reflects the growing interest in being able to represent and validate heterogeneous sources with Semantic Web technologies.

75 percent of the articles which presented a proposal for being able to *represent and edit shapes constraints in a visual manner* were published during the last years. This trend shows the efforts for paving the way to the massive adoption of these technologies.

The study of *validation involving inference* was published in a 83 percent during the last years. This trend, as some of the already mentioned ones, represent the attempt for improving the validation techniques in the Semantic Web, taking advantage of the inference power that the models posses.

The techniques developed for the *automatic or semi-automatic extraction of shape constraints* of Semantic Web models were presented in a 69 percent since 2018. With this effort, the community has been trying to ease and automatize the creation of the schema that then is useful for the validation of the models in a closed way. The intention behind this trend is to increase the consistency and robustness of these models, in order to make them more reliable and promote the adoption from domains that give special importance to the correctness of the model.

Almost a 65 percent of the works which presented a *constraint language definition* were published during the last six years. Taking into account the important amount of papers that are included in this category and that the oldest paper was presented in 2013, this kind of contributions could be taken as the main guide to understand the evolution of the validation tasks in the Semantic Web. Year by year the community has been putting the efforts in the creation of CLs that extend little by little their capabilities and features.

A 60 percent of the works related to *error detection, correction and prevention* were published in the period of time of interest. This could be interpreted as the intention to go beyond the validation task, and start to think in a more complex process, which includes the mentioned validation but also the correction of the errors found through the former task.

It worth to mention that both categories *"Specific type of constraints"* and *"SPARQL constraint implementation"* presented and interesting activity since 2018 but they are not taken as trending topics because almost a 60 percent of their articles were published before. Furthermore, both kinds of works presented activity since 2008. These categories are interpreted as important tasks that guided the evolution of the general topic, but that nowadays are mostly replaced by the new type of CLs. Lately, SPARQL is more used to present a validation example that then compares with others applied in the work. However, it is worth to mention that this also reflects the importance that this language has in the validation of Semantic Web models, since it has been present for almost two decades of work.

Finally, the works related to *"Validation reports (explanation)"* and *"Constraint languages - Further characteristics"* were a 100 percent published during the last 5 years. Since only 3 works of the former and 4 papers from the latter were included in the selected articles, these are not yet taken as a trend. However, the authors considered them as very interesting topics with potential to be converted into related trends during the following years. In the case of the validation reports, this insight is related to the several utilities that the explanations could have in the validation process, such as complex information or repair advices. The insight for the study of further characteristics of the CLs consists in the importance that these languages are having in the representation and validation of semantic constraints and the constant study of them which is moving the focus to new characteristics as an attempt for improving the language.

With respect to the *CL* used in the mentioned trending categories, taking into account the period of time described, almost all of them found in SHACL their preferred CL. This was quite expected since it is the standard CL. In some categories, the tendency of change from other languages to SHACL can be seen in the results crossed. However, the exception is the category *"SPARQL constraint implementation"*, in which as the title states, all the works employed the query language for the validation of the model. It is interesting to mention that since 2018, SHACL is also present in almost all of these works. In most of those cases, the validation is done with both languages. Regarding the rest of the languages found in the results and described in section 3.4.5, most of them were mainly used before the year 2018.

Regarding the *evaluation* of the proposals, in the big majority of the cases, the performance and the accuracy are the two characteristics that were taken into account. However, in those categories that are based mostly on theoretical developments and demonstrations, the validity is highly evaluated. This is the case in the CL categories that study the satisfiability and containment, the recursion and the semantic and syntactic definition.

Finally, an analysis of the most frequent intersections between the types of contributions that are part of the aforementioned trends are described. Only are taken into account those intersections in which the amount of articles intersected of each category represents more than the 20 percent of the total amount of papers belonging to the respective group.

The category *"SPARQL constraint implementation"* has an interesting intersection with *"Specific type of constraints"*. This relation exists because many of these proposals use the query language to define and validate semantic constraints on a knowledge representation model. In these cases, both tasks are done with CWA in order to be able to validate in a closed way. Another interesting combination is the study of *recursive shapes* and the *definition of constraint languages*. This is explained quite straightforward, since the recursion is not defined in the standard of SHACL, and these works present proposals of extended semantics, including this feature. In addition, the definition of CL semantics includes frequently the definition of a validation algorithm, with which some experiments are carried. This is also shown when the definition is related to recursive shapes and when an implementation of the language is presented. Going forward, the *inference power* of the models is well involved in the *detection, prevention and correction of data errors*. Most of these techniques infer the implicit knowledge without materializing it. Having this information, some actions could be taken and the errors can be prevented or directly corrected. Moreover, the inference of conflicts between the data and the semantic constraints is frequently used for the creation of detailed and more complex explanations which are presented to the user through the validation reports. Finally, although the percentage related to *mapping techniques* is lower than the 20 percent, taking into account that this category has an important amount of members, two more intersections are worth to mention. These techniques are frequently part of the works that present an *analysis of constraint languages*. In most cases, the mapping is done from a CL to a logic representation (for example SHACL to Description Logic)

or to another CL (for example SHACL to ShEx). The objective in these translations is the analysis of the semantic equivalence between both sides. Furthermore, the *mapping techniques* are also related to those works that present a *schema extraction* technique. This is explained because some of the works related to the latter present a first step that consists in the extraction of a set of constraints, and then these constraints are needed to be mapped to a CL in order to represent them as a schema.

3.6.2 Research gaps

The validation of semantic constraints on knowledge representation models is immersed in a strong tendency related to the development of shape CLs. This tendency has grown since the definition of SHACL as a standard for this purpose. Taking into account that this has happened very recently, there are several scientific challenges to work with, in order go beyond the limits of this new path. In this section, the research gaps found in the results are briefly described.

In order to path the way for the growth in the acceptance of the Semantic Web technologies, it is necessary to develop **visual interfaces** able to abstract the complexity of dealing with CLs for those people who are not experts in the domain. Up to this date, only a few proposals were done, and even the more developed one presented just initial results.

The SHACL standard left open the definition of how the **validation reports** should be generated and shared with the users. Although very little work was found with respect to this feature, this topic will attract the community attention in the following years. The current reports could be improved with the quality and quantity of information that is shared. Furthermore, the report could be more than a simple informative log. Instead, **a list of recommended actions could be shared, including data and/or constraints updates**.

The **validation of property graphs** has only started to be researched. Only two related works were found which presented different semantic definitions proposals. The authors expect this topic to catch more attention in the near future because of the potential that these models have. **The definition of well-founded semantics and a validation approach are the natural following steps.**

The mapping techniques are useful for the interoperability and integration of heterogeneous models and sources with the Semantic Web. However, more work has to be done over these techniques in order to **maintain as much semantics are possible**. Furthermore, most of the found techniques are specific to the case in which they are being applied. The development of a **generic mapping technique**, independent from the specific context and capable to be the intermediate layer between SHACL (and ShEx) and the rest of the most popular data models would be a big step forward in the development of the Semantic Web.

Going forward, the creation of **extensions and analysis of the existing CLs** still has a lot to offer. An interesting path seen in the results is based on a soft validation of the constraint models, which could offer a customizable way of accepting the natural incompleteness of the knowledge representation models. Furthermore, the **combination of validation with error detection, correction and prevention** is an interesting path for focusing not only in the detection of the inconsistencies, but also in the repair of the model.

The development of schema extraction techniques represents the possibility to generate automatically a shape graph from a KG, an ontology and other knowledge representation models. Although there are some proposals in the results, more work has to be done in order to **generate the shapes in an efficient way**. First of all, since the knowledge models could have enormous sizes, **the performance of the techniques is a key characteristic that needs to be improved**. Secondly, **the constraints generated are currently restricted to subsets of the**

well-known CLs. In order to generate shapes graphs as expressive as possible, research must be done. Having into account that a common scenario in the Semantic Web would involve multiple and heterogeneous sources, another interesting step would be the generation of an architecture that could be able to **extract shapes from multiple heterogeneous sources**. Furthermore, the latter would generate multiple shape graphs which would bring the necessity to **generate a technique for the integration of multiple shape graphs**. Finally, the **generation of shapes of spurious data and the algorithms to avoid it** represent another research challenge. Besides, techniques for the **extraction of shapes from property graphs** could be studied.

Going forward, the currently studied features of **shapes containment and recursion** could be taken into account as a key part in the generation of **new techniques for schema extraction, rewriting and integration**. As a first step, they need to be included in the expected semantics. Furthermore, the detection of shape containment during the generation of the outcome could be useful to improve the performance and the quality.

The contributions of this thesis which are described in the following chapters are mainly related to the challenges that are derived from the creation of shape graphs of a heterogeneous context. More precisely, the integration of multiple shape graphs is studied through the proposal of a new technique.

3.7 Shape graphs integration related work

As already mentioned, the following contributions of this thesis are related to the integration of multiple shape graphs as a consequence of a scenario in which the schema extraction needs to be done from multiple heterogeneous sources. Since these challenges represent an extension of the taxonomy presented in the previous sections (directly linked with the *"schema extraction"* category), the related work of semantic constraint integration techniques was done after the creation of the aforementioned taxonomy and is presented as follows.

3.7.1 Integration of constraints from heterogeneous sources

Since there is very little work related to the integration of shape graphs in the Semantic Web environment, the integration of constraints from heterogeneous sources is first analyzed from a broader point of view, taking into account other types of models such as RDBs.

The systems which include heterogeneous sources could create a scenario in which is desired or necessary the integration of concepts and constraints in order to represent the characteristics and restrictions of a unified model. This scenario presents at least two challenges: first to define the **concepts equivalence**, meaning that two or more concepts from different sources are referring to the same concept in the systems environment and should be integrated into one individual concept in an integrated model. Second, the **constraint integration and conflict resolution**, which includes detecting the constraints related to the concepts which are equivalent and resolve the integration of them. The latter task includes some cases in which a conflict could be found, meaning that two constraints present an incoherence between them or could have some kind of relationship that creates space for taking design decisions. In addition, these scenarios could involve different technologies. More specifically, and involving RDBs, there are works which define the aforementioned scenarios and present some constraint conflict resolution strategies for easing the creation of an integrated model ([173], [28]). When it comes to the equivalence detection, the characteristics of the concepts are taken into account, such as the names and the related attributes. On the other hand, the integration strategies for the resolution of constraint

conflicts vary from leaving the decision to an expert, keeping the less restrictive constraints (creating a softer model) or the creation of an integrated model which keeps the conflicting constraints through the creation of concept hierarchies.

3.7.2 Integration of SHACL shape graphs

As already mentioned in the taxonomy's *"schema extraction"* category, there exist some techniques which are dedicated to extract SHACL shapes from a specific type of source. Furthermore, these techniques still present some drawbacks or limitations such as the incomplete expressiveness on the generated outcome or the scarce flexibility related to the individual homogeneous input that the techniques are able to process. However, as an update to the latter drawback and after the SLR was finished, a technique which presents a first proposal to improve this limitation was found in the literature. When it comes to the extraction of shapes from multiple sources that could have different formats, the only work presented in the literature is the one presented in [50]. In this work, SHACL shapes are extracted from ontologies through the use of Astrea ([34]), from RML mappings by the use of the RML2SHACL technique ([45]) and finally from raw data schemas in XML Schema Definition (XSD) using the tool XSD2SHACL ([51]). In addition, the aforementioned techniques are all part of a bigger system which is able to use them as needed. Moreover, the extraction of SHACL shape graphs from multiple heterogeneous sources creates a scenario in which more than one shape graph is obtained and the integration of them could be desired. In consequence, this new proposal also represents the only work found in the literature which is related to the integration of SHACL shapes. A part of this tool has the objective of integrating shapes graphs through a procedure that consists of three stages:

- The first stage is the equivalence detection and looks for determining which shapes will be integrated. In order to do this, the explicit and implicit targets and path-values combinations are compared.

- The second stage presents three alternative strategies for the integration and the resolution of conflicts. The SCOOP-ALL technique aims to integrate all the equivalent shapes from all the sources involved and resolves the conflicts by adding the constraints with the sh:or element. Moreover, the SCOOP-PRIOR strategy integrates the equivalent shape depending on a predefined source priority and discards the shapes when a conflict appears. Finally, the SCOOP-PRIOR-R makes sense in those cases in which one of the shapes comes from a source ontology, since those shapes which were generated from this source and have no equivalent shape are directly filtered.

- The third stage consists in the resolution of intra constraint inconsistencies that could be found in the integrated shapes, for example that it's not possible to have two sh:minLength definitions on a shape. In addition, inter-constraint inconsistency check are done by checking incompatibilities between constraints (for example sh:nodekind sh:IRI and sh:datatype xsd:string).

The aforementioned work is quite interesting, and some parts of the presented techniques are reused by the proposal presented in this thesis (mainly with respect to the equivalence detection). However, the main differences are reflected in the conflict resolution strategy defined and, in consequence, the unified shape graph generated is different. In this thesis, the constraint conflicts are resolved by a comparison of the involved constraints types, their respective values and the containment relationship that they hold. In addition, the generated shapes graphs will keep a

comprehensive representation of the characteristics found in both sources, representing a hierarchy for those shapes which aim for the same targets and presented conflicts. The unified shape graph generated by this thesis proposal represents, inside these hierarchies, different shapes that could be activated depending on which is the current context. In other words, the generated shape graph will have the flexibility for validating the data with different constraints, depending on which of them are desired at a precise moment. This shape graph representation and the integration technique proposed in this thesis are explained in detail in the following chapters.

Table 3.2: Queries per database. The use of * is needed in some databases for retrieving all the possibles word terminations after it.

Database	Query
Scopus	TITLE-ABS-KEY (("semantic* constrain*" OR "semantic restriction" OR "ontological constraints" OR shapes OR shacl OR shex) AND ("knowledge formali*" OR "knowledge representation" OR "semantic* network" OR rdf OR owl OR ontology OR "knowledge graph"))
WoS	TS=(("semantic* constrain*" OR "semantic restriction" OR "ontological constraints" OR shapes OR shacl OR shex) AND ("knowledge formali*" OR "knowledge representation" OR "semantic* network" OR rdf OR owl OR ontology OR "knowledge graph*"))
IEEE Xplore	("All Metadata"::"semantic constrain*" OR "All Metadata":"semantic restriction*" OR "All Metadata":"ontological constraint*" OR "All Metadata":shapes OR "All Metadata":shacl OR "All Metadata":shex) AND ("All Metadata":"knowledge formali*" OR "All Metadata":"knowledge representation" OR "All Metadata":"semantic network*" OR "All Metadata":rdf* OR "All Metadata":owl* OR "All Metadata":ontology OR "All Metadata":"knowledge graph*")
ACM	Title:(("semantic constraint" OR "semantic constraints" OR "semantic restriction" OR "semantic restrictions" OR "ontological constraints" OR "ontological constraint" OR shapes OR shacl OR shex) AND ("knowledge representation" OR "knowledge formalization" OR "semantic network" OR rdf OR owl OR ontology OR "knowledge graph" OR "knowledge graphs")) OR Abstract:(("semantic constraint" OR "semantic constraints" OR "semantic restriction" OR "semantic restrictions" OR "ontological constraints" OR "ontological constraint" OR shapes OR shacl OR shex) AND ("knowledge representation" OR "knowledge formalization" OR "semantic network" OR rdf OR owl OR ontology OR "knowledge graph" OR "knowledge graphs")) OR Keyword:(("semantic constraint" OR "semantic constraints" OR "semantic restriction" OR "semantic restrictions" OR "ontological constraints" OR "ontological constraint" OR shapes OR shacl OR shex) AND ("knowledge representation" OR "knowledge formalization" OR "semantic network" OR rdf OR owl OR ontology OR "knowledge graph" OR "knowledge graphs"))

Table 3.3: List of inclusion and exclusion criteria, having ICn = "Inclusion Criteria n" and ECm = "Exclusion Criteria m".

Criteria	Description
IC1	The article is related to the validation of semantic constraints on knowledge representation models
IC2	The article is related to a formal and semantic constraint language
IC3	The articles are in the field of computer science
EC1	Conference proceedings, entire books, short papers, demo papers and posters
EC2	Studies presenting non-peer reviewed material
EC3	Studies not presented in the English language
EC4	Studies with incomplete information (title, abstract, keywords)
EC5	Studies not accessible in full-text
EC6	Studies that are duplicates of other studies
EC7	Papers that were published before the year 2001

Table 3.4: Papers by scientific database

Database	Total	Unique	Duplicated
Scopus	1054	1054	0
IEEE Xplore	524	375	149
Web of Science	606	191	415
ACM	531	177	354
Total	2715	1797	918

Table 3.5: SLR selected papers

Article ID	Article	Article ID	Article
1	[25]	30	[49]
2	[108]	31	[151]
3	[174]	32	[176]
4	[75]	33	[177]
5	[73]	34	[120]
6	[77]	35	[126]
7	[155]	36	[165]
8	[2]	37	[152]
9	[168]	38	[148]
10	[38]	39	[140]
11	[33]	40	[158]
12	[62]	41	[96]
13	[20]	42	[159]
14	[37]	43	[103]
15	[34]	44	[157]
16	[97]	45	[22]
17	[24]	46	[88]
18	[100]	47	[149]
19	[47]	48	[8]
20	[81]	49	[43]
21	[45]	50	[116]
22	[119]	51	[76]
23	[52]	52	[6]
24	[137]	53	[55]
25	[127]	54	[129]
26	[68]	55	[53]
27	[58]	56	[21]
28	[128]	57	[138]
29	[48]	58	[180]

Table 3.6: SLR selected papers

Article ID	Article	Article ID	Article
59	[3]	88	[30]
60	[23]	89	[112]
61	[141]	90	[170]
62	[125]	91	[70]
63	[44]	92	[80]
64	[69]	93	[26]
65	[142]	94	[65]
66	[156]	95	[154]
67	[61]	96	[164]
68	[104]	97	[14]
69	[167]	98	[131]
70	[78]	99	[10]
71	[114]	100	[132]
72	[84]	101	[13]
73	[99]	102	[95]
74	[160]	103	[101]
75	[147]	104	[150]
76	[106]	105	[153]
77	[56]	106	[41]
78	[163]	107	[63]
79	[162]	108	[92]
80	[46]	109	[117]
81	[139]	110	[169]
82	[66]	111	[115]
83	[59]	112	[118]
84	[166]	113	[136]
85	[145]	114	[79]
86	[179]	115	[91]
87	[7]	116	[1]

Table 3.7: SLR selected papers

Article ID	Article	Article ID	Article
117	[83]	121	[113]
118	[4]	122	[146]
119	[35]	123	[124]
120	[122]		

Table 3.8: Data Extraction.

Data Item	Value	RQ
Study ID	Integer	-
Title	Name of the article	-
Year	Calendar year of the publication	RQ1
Venue	Conference, Journal, Workshop, Book chapter	RQ2
Domains	In which domains the works are applied	RQ3
Repository	Yes, no	RQ4
Model	Ontology, Knowledge Graph,..	RQ5
Contribution	Type of contribution	RQ6
Constraint languages	SHACL, ShEx, SPIN,...	RQ7
Constraints involved	Cardinality, Range,...	RQ8
Evaluated characteristics	Consistency, Accuracy,...	RQ9

Table 3.9: Selected papers organized by type of contribution. CL stands for Contraint Language

Contribution	Article IDs	Amount	Section
Mapping techniques	[155], [38], [62], [34], [97], [24], [100], [47], [81], [45], [119], [52], [127], [68], [126], [165], [103], [149], [21], [156], [104], [160], [147], [59], [166], [7], [26], [13], [153], [136], [146]	31	3.5.8
CL applied cases	[174], [75], [77], [168], [96], [157], [88], [149], [55], [180], [125], [142], [167], [114], [99], [160], [163], [162], [66], [30], [112], [91]	22	3.5.4
Specific type of constraints	[25], [81], [119], [120], [140], [84], [59], [80], [154], [164], [10], [101], [153], [117], [169], [118], [83], [4], [35], [122], [113]	21	3.5.10
SPARQL constraint implementation	[81], [48], [49], [120], [55], [59], [170], [80], [65], [154], [164], [131], [95], [101], [153], [83], [113]	17	3.5.9
CL semantic and syntactic definition	[38], [38], [62], [20], [137], [151], [158], [22], [23], [46], [70], [13], [150], [79]	14	3.5.2
Schema extraction	[73], [34], [119], [58], [116], [160], [139], [145], [41], [117], [118], [136], [124]	13	3.5.11
Other approaches	[108], [177], [140], [76], [141], [14], [132], [92], [35], [122]	10	3.5.18
Error treatment	[52], [48], [49], [76], [6], [179], [7], [95], [63], [115]	10	3.5.14
Validation algorithms	[108], [38], [37], [137], [158], [8], [23], [69], [61]	9	3.5.17
CL recursion	[38], [38], [20], [37], [127], [126], [22], [8], [13]	9	3.5.5
Visual constraints creation and validation	[176], [152], [53], [138], [44], [78], [106], [146]	8	3.5.12
CL implementation	[62], [137], [151], [148], [69], [61], [46], [26]	8	3.5.3
Inference in validation	[25], [73], [43], [6], [129], [179]	6	3.5.15
CL satisfiability and containment	[2], [127], [128], [126], [159], [103]	6	3.5.6
CL analysis and review	[97], [126], [21], [167], [80], [169]	6	3.5.1
Query optimization	[2], [62], [3], [101], [1]	5	3.5.16
CL further characteristics	[38], [127], [22], [8]	4	3.5.7
Validation reports (explanation)	[43], [6], [56]	3	3.5.13

4

Integrated Restrictions Model

Contents

4.1 Introduction 63
4.1.1 Motivation 63
4.1.2 Preliminaries 65
4.2 Representation approach 68
4.2.1 Design of the Integrated Restriction Model 68
4.2.2 Pre-validation configuration 73
4.2.3 Source provenance representation 78
4.3 Conclusion 81

4.1 Introduction

This chapter addresses the lack of flexibility in a single SHACL shape graph when validating different contexts with varying needs. It introduces the problem and provides the motivation for a better understanding. Then, based on the SHACL standards, the design of an Integrated Restriction Model (IRM) is presented as a solution proposal for this problem. In addition, since this model could represent shapes which carry incoherencies between them, a pre-validation configuration process is presented which is able to set the model ready for validating RDF data. Finally, a lightweight ontology for the source provenance information of the IRM shapes is presented. This model is designed for representing the knowledge related to each specific integration and is able to represent a link between specific shapes of the IRM and the source shape graphs (SSGs) which were part of the integration process.

4.1.1 Motivation

Recalling some SHACL definitions, shape graphs are specific RDF graphs which are conformed by a *set of shapes S = {s1, s2, ..., sn}*. Each of these shapes represents a set of constraints which are targeted to a specific subgraph of the RDF data graph that will be validated. In other words, the *set of constraints C = {c1, c2, c3, .., cn}* of a specific *shape Sx* are defined for the validation of a specific *set of targets tx* (which could comprehend one or more targeted nodes). In addition, there exist two types of shapes: the *Node Shapes* which target a specific node (called *focus node, fn*) and the *Property Shapes* which target a *value node vn* to which is possible to arrive from

the *focus node fn* through the predicate *path px*, represented as the triple pattern *(fn,px,vn)*. To illustrate this, consider the shape in the listing 4.1. This shape is targeting all the focus nodes which are instances of the *class GroupLesson* and that holds a *property px has_students* which connects the focus nodes to a set of *value nodes vn*. This specific target definition could be represented as the *target (GroupLesson,has_students)*. Furthermore, what this example shape is validating is that the instances of *GroupLesson* must have a number of *students* comprehended *between 3 and 10*.

The aforementioned example sounds simple, and there shouldn't be any problem with its behavior. However, there are some limitations with respect to it. Let's imagine a scenario in which the SHACL cardinality restrictions (such as *sh:minCount* and *sh:maxCount*) applied to the number of *students'* instances that a *GroupLesson* could have, depends on the characteristics or requirements of a *specific external context*. Going forward, imagine that the *weather conditions (e.g. of an external context) influence directly the number of students' instances allowed*. If the weather conditions are good enough, then the amount of students should respect the cardinality interval [3,10]. However, if the weather conditions are bad, for safety reasons, the number of students desired changes to the cardinality interval [3,6]. Since the interval of values [3,6] = [3, 4, 5, 6] is contained in the interval [3,10] = [3, 4, 5, 6, 7, 8, 9, 10] *(*$[3, 6] \subseteq [3, 10]$*)*, the described shape wouldn't detect if the amount of students desired for the bad weather context is being respected. In order to clarify, if a *GroupLesson* has 7 *students*, this would be perfectly accepted by the example shape. So this scenario opens the following question: how is it possible to represent a SHACL shape graph *Sg* which represents *a set of shapes S = {s1, s2, ..., sn} aiming to the same set of targets t = (fn,px)*, e.g. *(GroupLesson, has_student)* with *different set of constraints C = {c1, c2, ..., cn}* which could present incoherencies between one shape and the other? The incoherencies between these shapes take place when they have constraints which wouldn't be compatible in a single shape (e.g. *sh:maxCount 6* and *sh:maxCount 10*). In other words, how is it possible to create a SHACL shape graph which is flexible enough to restrict a *target ti* in different ways, depending on an *external specific context*?

A straightforward solution would be to have a different shape graph for each specific context, and use the corresponding one depending on which context is currently active. However, this could be more and more complex as the number of contexts increases, leading to an important amount of individual shape graphs. In addition, even if they are defined on the same domain or environment, the set of *targets t1* aimed by the *set of shapes S* of an individual shape graph s1 could be different to the set of targets t2 aimed by a second shape graph s2. This would mean that when attempting to use an individual shape graph, the set of related targets would be incomplete if a set of instance knowledge from the whole domain or environment is desired to be validated. Moreover, this would be translated to the lose of certain valuable restricting knowledge when validating. In consequence, in these cases, it would be useful to have a shape graph which could represent the complete set of targets T, which includes the targets of the individual shape graphs and, at the same time, represents sets of shapes which aims to the same target t with a variety of sets of constraints.

In order to solve the aforementioned issues, the idea of an IRM follows the objective of having a unique and unified SHACL shape graph which contains a comprehensive integration of the individual source shape graphs (SSGs), keeping a variety of shapes that are grouped by their respective *target ti*. In other words, the objective is to cover not only all the possible context requirements, but also the complete set of *targets t* of the whole environment. Besides, this model would represent a comprehensive combination of the individual shape graphs that are part of a determined environment (e.g. an enterprise), including all the sets of constraints and the complete sets of targets. Solving this problems would lead to SHACL shapes graphs which

are more useful, since they would be able to cover more scenarios than usual. Nevertheless, having different contexts with different characteristics, is a scenario that reflects better what happens in the real world use cases.

4.1.2 Preliminaries

SHACL shapes equivalence

The equivalence of shapes in this thesis is inspired in the equivalence detection defined in ([50]) and is described as following: **Two shapes *S1* and *S2* are considered equivalent if and only if their defined *set of targets t* are the same.** Specifically, this thesis considers first two different cases: the first one is for two simple *Node Shapes NS1* and *NS2* ("simple" refers that these node shapes don't have another nested nor referred shape) and they are considered equivalent if they aim to the same *set of focus nodes or targets t* through the definition of the property *sh:targetClass classX* (the classes in both node shapes are the same). As an example, the shapes defined in the listings 4.2 and 4.3 are interpreted as equivalent, since both shapes have the *sh:targetClass* constraint with the value *ex:Instructor*. The second case is for two equivalent *Node Shapes NS1* and *NS2* which are not simple. This means that in addition, these node shapes have one or more nested property shapes. In these cases, the equivalences are defined for each pair of *(NodeShape, PropertyShape)*, being the first one identified by the set of focus nodes targeted as *t* and the second one identified by the specific value defined for the property *sh:path*. If there are two pair of *(NodeShape, PropertyShape)* which are equal, they are considered as equivalent shapes. In addition, if a Node Shape has more than one property shape nested, at the time of evaluating the equivalences, this shape will be separated into a set of pairs Eq_P = *{(NodeShape, PropertyShape1), (NodeShape, PropertyShape2),...}*.

As an example, recall the shape in the listing 4.1. This shape presents a sh:targetClass property which aims to the class ex:GroupLesson, and a nested property shape which refers to the sh:path ex:has_students. In consequence, this shape would be considered equivalent to any other shape which presents exactly the pair (GroupLesson, has_students) as the values for *sh:targetClass* and *sh:path* respectively. An example of an equivalent shape is shown in the listing 4.4. Besides to the first two presented cases, there are two more scenarios in which the definitions of the elements of interest occur in a referred shape. The first case is represented when a simple node shape NS1 has a sh:node element which refers to a second simple node shape NS2, in which there is a target definition. In this case, NS1 is directly linked to the set of focus nodes defined in NS2, and this relationship is involved when looking for equivalences. As an example, the listing 4.5 shows a node shape which defines its target in a referred shape. In this case, the target assigned to the shape ex:InstructorAbstractedShape is the class ex:Instructor and this is the id used for the equivalence procedure. The final scenario takes place when a node shape NS1 has a sh:property element which refers to a non-nested property shape PS1 (meaning not a blank node, but instead another defined shape). In this case, the path value defined in PS1 is linked to the focus nodes defined in NS1, and this combination conforms the respective target id used for the detection of equivalences. The listing 4.6 shows an example in which the node shape ex:GroupLessonPropertyAbstractedShape defines as the target the class ex:GroupLesson and refers to the property shape ex:hasStudentPropertyShape. Furthermore, the latter defines a set of constraints for the property ex:has_students. As a result, the equivalence id set for the node shape ex:GroupLessonPropertyAbstractedShape is (ex:GroupLesson, ex:has_students).

Listing 4.1: Simple example of a compound SHACL shape

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:Annecy_GroupLessonShape
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 3 ;
        sh:maxCount 10 ;
        sh:class ex:student ;
        ] .
```

Listing 4.2: Example of a simple node shape which targets the class ex:Instructor

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:InstructorShape
    a sh:NodeShape ;
    sh:targetClass ex:Instructor ;
    sh:nodeKind sh:IRI .
```

Listing 4.3: Example of a second simple node shape which targets the class ex:Instructor

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:InstructorShapeIRI
    a sh:NodeShape ;
    sh:targetClass ex:Instructor .
```

Listing 4.4: Example of a compound node shape which targets the class ex:GroupLesson and the value nodes reached through the property ex:has_students.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLessonShape
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 5 ;
        ] .
```

Listing 4.5: Example of a simple node shape which defines its target with the property sh:node in a second referred shape.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:InstructorAbstractedShape
    a sh:NodeShape ;
    sh:nodeKind sh:IRI ;
    sh:node ex:InstructorShape ;

ex:InstructorShape
    a sh:NodeShape ;
    sh:targetClass ex:Instructor .
```

Listing 4.6: Example of a node shape which refers to a property shape with the element sh:property.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLessonPropertyAbstractedShape
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:property ex:hasStudentPropertyShape .

ex:hasStudentPropertyShape
    a sh:propertyShape ;
    sh:path ex:has_students ;
    sh:minCount 3 ;
    sh:maxCount 10 .
```

SHACL containment

In this thesis, the SHACL containment characteristic in the domain of shapes graphs is particularly useful. **The SHACL containment is defined as follows: *"for all SHACL documents M1, M2, M1 is contained in M2, denoted*** $M1 \subseteq M2$ ***, iff for all graphs G, if G conforms to M1 then G conforms to M2. Deciding whether a SHACL document is contained in another is the decision problem of SHACL containment".*** Furthermore, the SHACL containment could be evaluated not only at a SHACL document level, but also at a shape and constraint levels. **The SHACL constraint containment is defined as follows: *"for all SHACL constraints c1 and c2, c1 is contained in c2, denoted by*** $c1 \subseteq c2$ ***iff for all graphs G and nodes n, if G conforms to c1(n) then G conforms to c2(n)"*** (both definitions can be found in [127] and [128]). As an example, consider the two shapes described in the listing 4.7, where the target is defined as the object nodes of those triples which have the property *ex:startsInPlce* and there are two constraints defined. *c1 = sh:nodeKind sh:IRI* defined in the shape *ex:MeetingPointShape1* and the constraint *c2 = sh:nodeKind sh:IRIOrLiteral* defined in the shape *ex:MeetingPointShape2*. In addition, take into account the RDF data graph *G* shown in the listing 4.8, where the object node of the triple in line 4 is taken as *n1 (an IRI)* and the object node of the triple in line 5 is taken as *n2 (a literal)*. Then *G* conforms to *c1(n1)* since the node *ex:MeetingPoint1* is of type *IRI* and *G* conforms to *c2(n1)* as well. In addition, with respect to *n2*, *G* conforms to *c2(n2)* but not to *c1(n2)*. This means that *c2* conforms to the *G* nodes to which *c1* conforms and additionally it conforms to extra nodes. *c1* is accepting all those nodes of type *IRI*, meanwhile *c2* accepts all those nodes of type *IRI* and all those nodes of type *Literal*. This means that *c2* accepts a space of nodes which includes the space of nodes accepted by *c1* ($c1 \subseteq c2$). Then, in this example, *sh:nodeKind sh:IRI* $\subseteq$ *sh:nodeKind sh:IRIOrLiteral*.

Listing 4.7: Example of two node shapes which define the type of node allowed (with sh:nodeKind) for those nodes which are the objects of those triples that have the property ex:startsInPlace.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:MeetingPointShape1
   a sh:NodeShape ;
   sh:targetObjectOf ex:startsInPlace ;
   sh:nodeKind sh:IRI .

ex:MeetingPointShape2
   a sh:NodeShape ;
```

```
11    sh:targetObjectOf ex:startsInPlace ;
12    sh:nodeKind sh:IRIOrLiteral .
```

Listing 4.8: Example of a RDF data graph

```
1  @prefix ex: <http://localhost/ontology/esf_ontology/> .
2  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3
4  ex:lesson1 ex:startsInPlace ex:MeetingPoint1 .
5  ex:lesson2 ex:startsInPlace 'MontBlanc main entrance' .
```

4.2 Representation approach

4.2.1 Design of the Integrated Restriction Model

The design of the Integrated Restriction Model (IRM) is first described as the result of an integration process which takes into account a set of individual SHACL shape graphs for the creation of the model. In this way, the definition relates the IRM with a set of sources represented as individual SHACL shape graphs which are desired to be integrated. These sources are defined as follows:

Sg is an input SHACL shape graph which has a *set of shapes S* associated *{s1, s2,..sN}*. Each of these shapes *Sx* has a defined target, namely *Ti* on the *data graph Dg*. These *targets Ti* are identified by the *focus node (Fn)* if it's a *simple NodeShape*. Furthermore, if the *shape Sx* has a *nested property shape*, then the *targets Ti* are identified by the *combination (Fn, Pv)*, being *Pv* the related *path value*. In addition, each *shape Sx* has a *set of constraints C = {c1, c2,..,cN}* that aims to restrict the representation of the defined *target Ti* in *Dg*. Moreover, a set of *SSGs = {Sg1, Sg2, .., Sgn}* represents every graph which was desired to be integrated in the IRM. Both of these definitions can be seen graphically in the figure 4.1.

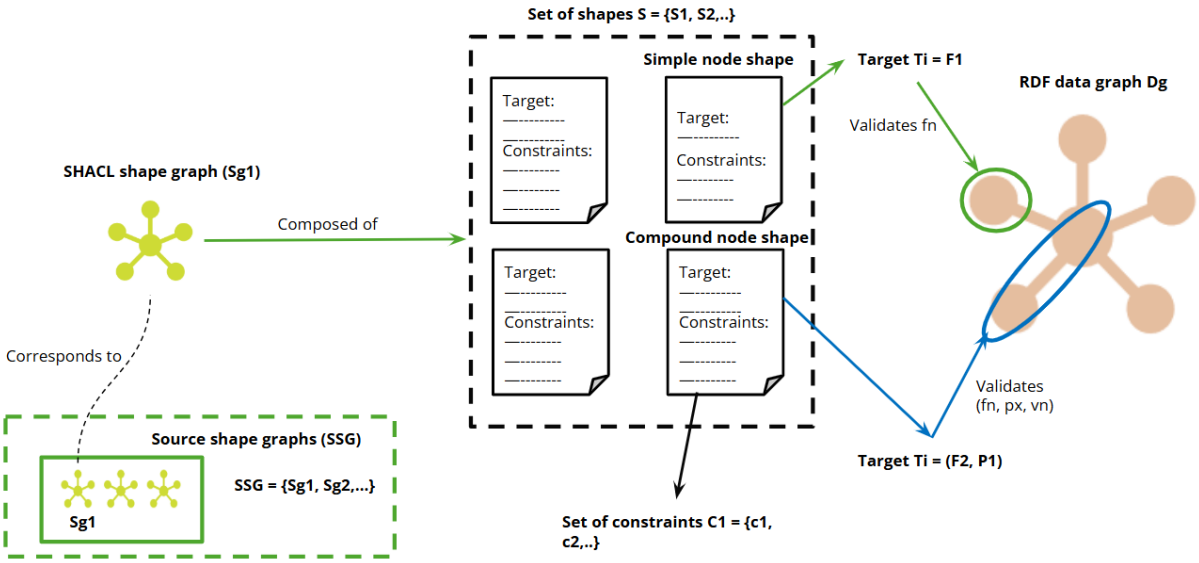


Figure 4.1: Description of a set of Source Shape Graphs (SSG) which are a set of Shape Graphs which are desired to be integrated in the IRM. Furthermore, the structure of a shape graph is detailed, highlighting the importance of the target Ti definitions.

Taking the aforementioned definitions into account, an IRM is a *SHACL shape graph Sg* which has a *set of shapes S = {s1, s2, .., sn}*. Each of these shapes *Sx* belonging to *S* could have been obtained from one of the *SSGs* that have been already processed or could be the result of an integration of more than one shape belonging to more than one Sg from the *set of source graphs SSG = {Sg1, Sg2, .., SgN}*. As a result, each *shape Sx* could have it provenance related to a single *Sgx* from *SSG* or to an integration of more than one *Sgx* from *SSG*. Furthermore, there could be more than one shape *Sx* which aims to the same *target Ti [(Fn) or (Fn,Pv)]* (namely equivalent shapes). The *group* of shapes *GSti* which aim to the same *target Ti* are identified by *sh:group Ti* and represented as a *sh:propertyGroup*, which respective *IRI* respects the pattern *:(fn)_(pv)*, being *fn* the targeted set of focus nodes and *pv* the related path value. Moreover, the shapes in these groups are organized in a hierarchical way. This organization respects a relationship in which a *super shape SupS* of the *GS1* contains its *sub shapes SubS = {SubS1, SubS2,..,SubSn}* which belong to the same group *(SubS ⊆ SupS)*. In addition, each *sub shape SubS* references its right after *super shape SupS* with a *shape-based constraint (SubS sh:node SupS)*. A graphical explicative figure of the IRM can be found in 4.2. This image first shows on the left, a set of *Source shape graphs (SSG)* composed by three shapes *Sg1, Sg2, Sg3*. In addition, the integration of these shape graphs generates an IRM, which is in the center of the figure. As a high-level explanation, it's shown that the IRM shapes could be related to one or more shape graphs from the source. The *"IRM shape 1"* is linked to the *Sg3*, meanwhile the *"IRM shape 2"* is related to *Sg1* and *Sg2*. The former states that the shape wasn't the result of an integration between two shapes from different graphs. The latter is interpreted as the result of an integration of two shapes, where one belonged to *Sg1* and the other one to *Sg2*. Moreover, on the right part of the figure, it's highlighted the possibility of having more than one shape which aims to the same *target t*. This is the case for the shapes *"IRM shape 3"* and *"IRM shape 4"*. Finally, the integration of these shapes is organized in a SHACL group where the shape members are structured in a hierarchical way. In the following subsection, a more practical example is detailed.

Furthermore, the IRM model design could be also analyzed independently of the integration of sources. The main contribution of this design holds in the capacity for representing groups of shapes *GSti* which present more than one shape *Sx* member with the same *target ti*, where a *shape member Sx* could have constraints that present conflicts with the constraints of another *shape member Sy* of the same group *GSti*. In addition, another important feature is the hierarchical relationship stated between the shape members of these groups.

Analyzing the IRM features through an example

In this section, the IRM is analyzed through a scenario example in which there is a definition of *group lessons* and the amount of *students* that these types of lessons could have, which could vary depending on the *current weather conditions*. The listing 4.9 represents an example of a shape that could apply to a good weather scenario, having as the *target t* the combination *(GroupLesson, has_student)*. In the IRM design, this shape would be a member of the sh:PropertyGroup defined with the IRI *:GroupLesson_has_student*. Going forward with the example, the scenario presents a second weather condition context (bad weather) in which the amount of students allowed for a group lesson changes to [3,6] for safety reasons. In order to detect this difference at the validation time, it's necessary to create a new shape which will be a member of the same group *:GroupLesson_has_student*. The listing 4.10 shows the two shapes members of this group. At this point, there are already the two shapes which belong to the same group *:GroupLesson_has_student*. However, there are two main problems that these shapes present with

respect to the IRM: first, the relationship between the two shapes is not clear, beyond that they share the *target ti*. Second, attempting to validate a graph with both shapes could generate a result not desired, since their constraints hold incoherencies. With respect to the former, the IRM design, as already mentioned, expects that the shapes of a group are organized hierarchically with respect to the containment relationship of their respective constraints and, in consequence, of the shapes itself. In the example, there's only two constraints which present a conflict: *sh:maxCount 10* from the first shape and *sh:maxCount 6* from the second one. Analyzing the containment relationship between them would mean to analyze the relationship between the intervals [0,10] and [0,6]. The analysis of the containment relationship between two constraints is done in different ways, depending on the specific SHACL constraint type that it's being evaluated and the type of values that it could hold. Some examples of these definitions (including *sh:maxCount*) are shown in the table 4.1. It's quite straightforward to define that $[0, 6] \subseteq [0, 10]$, and as a result, the IRM represents the second shape as a sub shape of the first shape. It's worth to highlight that the sub shape is stricter than the super shape, meaning that the space of values that it accepts is smaller than the once accepted by the super shape. Furthermore, the sub shape will refer to its super shape with the element *sh:node* which will have as a value the IRI of the super shape. Finally, in order to understand the hierarchy level of a shape, the IRM represents the IRIs of the shape with the pattern *:{fn}_{pv}_{hierarchyLevel_{branchNumber}}*. A correct representation of these shapes is presented in the listing 4.11. Besides, the figure 4.3 represents the group and the hierarchical relationship of the shapes in a graphical manner. This hierarchical organization that the IRM proposes allows to easily understand which is the relationship between the shapes of a group GSti, with respect to their containment and to their strictness. The given example represents a simple structure, however, in a real case, the hierarchical structure of a *group GSti* could be formed by multiple levels with different branches included.

Listing 4.9: A compound SHACL shape which targets the class ex:GroupLesson and the value nodes which are reached through the path has_students. In addition the shape defines an interval cardinality between 3 and 10. This restricts the value nodes to the class ex:student and defines the type of node expected as an IRI.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLessonShape
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 3 ;
        sh:maxCount 10 ;
        sh:class ex:student ;
        sh:nodeKind sh:IRI ;
        ] .
```

Listing 4.10: Intermediate example of two IRM shapes which belong to the same group. Both shapes target the class ex:GroupLesson and are part of the same group. The two shapes have a property shape with the property has_students. These property shapes present some differences from which the most important one is that both shapes have the constraint sh:maxCount with different values assigned which represents a constraints conflict.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLessonShape
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:group ex:GroupLesson_has_students ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 3 ;
        sh:maxCount 10 ;
        sh:class ex:student ;
        sh:nodeKind sh:IRI ;
        ] .

ex:GroupLessonShape2
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:group ex:GroupLesson_has_students ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 3 ;
        sh:maxCount 6 ;
        sh:class ex:student ;
        ] .

ex:GroupLesson_has_students a sh:propertyGroup .
```

Constraint type	Comparison used	Example
Value type – sh:nodeKind	Predefined values	sh:IRI is contained in sh:IRIorLiteral
Value type – sh:class	Specific defined classes	sh:class [class1] != sh:class [class2]
Cardinality - sh:minCount	Integer interval comprehended	sh:minCount 2 = [2,∞) is contained in sh:minCount 1 = [1,∞)
Cardinality - sh:maxCount	Integer interval comprehended	sh:maxCount 6 = [0,6] is contained in sh:maxCount 10 = [0,10]
Value Range - sh:minExclusive	Integer interval comprehended	sh:minExclusive 2 = (2,∞) is contained in sh:minExclusive 1 = (1,∞)
Value Range - sh:maxExclusive	Integer interval comprehended	sh:maxExclusive 6 = [0,6) is contained in sh:maxExclusive 10 = [0,10)
Value Range - sh:minInclusive	Integer interval comprehended	sh:minInclusive 2 = [2,∞) is contained in sh:minInclusive 1 = [1,∞)

Table 4.1: Examples of the comparison used for some SHACL constraints when evaluating their containment relationship. The column "Constraint type" defines the specific constraint that is being detailed. The column "Comparison used" describes what is used at the time of contrasting the values assigned to the constraints. Finally, the column "Example" provides a demonstration of how the relationship between the constraints is evaluated.

At this point, it's clear that the IRM is able to represent shapes with different characteristics

for validating the same *target ti*, which gives to the model the capability for representing the constraints related to different specific contexts. In other words, the validation could behave differently depending on which shape of the group is used. Although, as already mentioned, there is an issue that comes with the IRM when the moment of validation arrives. If a group has shapes which present conflicts between them, the behavior could not be the best one. For this reason, before the validation, some actions must be taken on the IRM, which are detailed in the following section.

Listing 4.11: Finished example of two IRM shapes which belong to the same group. The shapes' IRIs respect the IRM pattern defined. The conflict between the constraints sh:maxCount has been resolved putting the softer value in the super shape and the restricter value in the sub shape. The sub shape aims to its super shape through the property sh:node representing the hierarchy relationship between them.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLesson_has_students_sup
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:group ex:GroupLesson_has_students ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 3 ;
        sh:maxCount 10 ;
        sh:class ex:student ;
        ] .

ex:GroupLesson_has_students_sub1_1
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:node ex:GroupLesson_has_students_sup ;
    sh:group ex:GroupLesson_has_students ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 3 ;
        sh:maxCount 6 ;
        sh:class ex:student ;
        sh:nodeKind sh:IRI ;
        ] .

ex:GroupLesson_has_students a sh:propertyGroup .
```

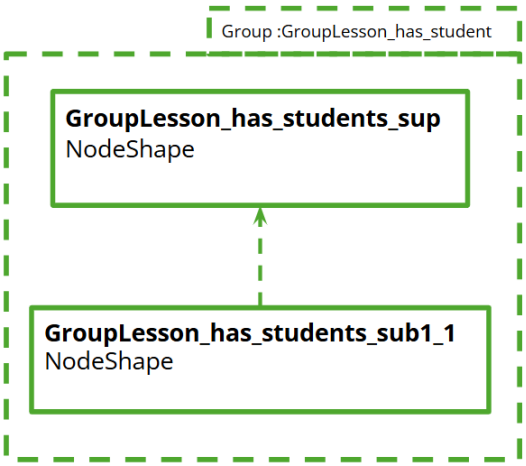


Figure 4.3: This figure represents two shapes which are part of the same group (GroupLesson_has_student). The hierarchical relationship if represented with the arrow which states an extension.

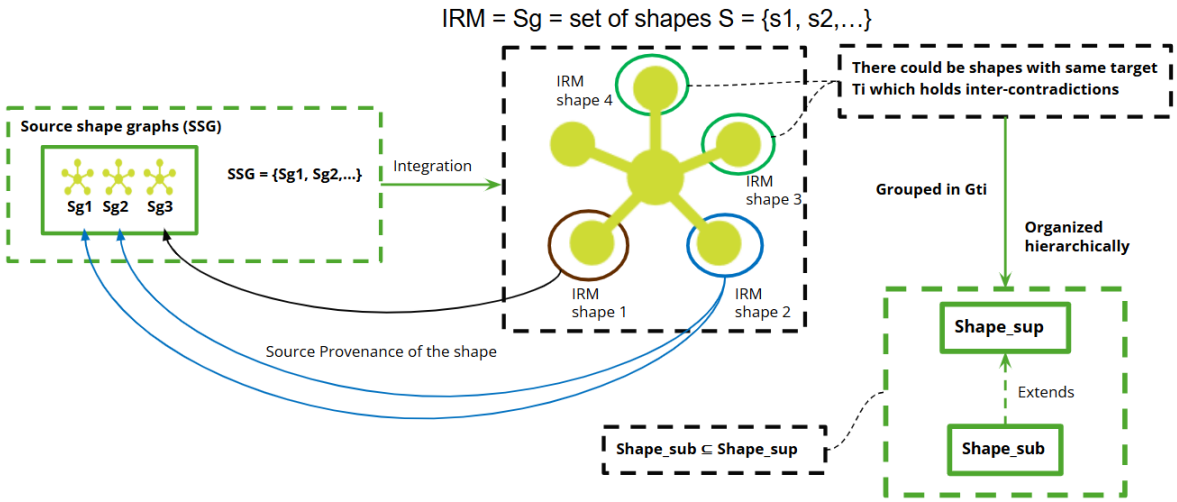


Figure 4.2: Brief graphical description of an IRM

4.2.2 Pre-validation configuration

Since the IRM could have groups of shapes which aim to the same target ti and could hold constraint incoherencies between them, it's necessary to take some actions before the use of the model for validating an RDF graph. To explain this with a simple example, let's go back to the IRM group example in the listing 4.11. This group is formed by a small hierarchy with two shapes which aim to the target *(GroupLesson, has_student)*. Each of them is able to respond to a different context which expects a different amount of students attached to this type of lessons. Now, let's imagine that both shapes are active in the IRM. If the current active context expects that the amount of students is in the interval [3,6], then the sub shape *:GroupLesson_has_students_sub1_1* is the expected one since it'll control that the amount of students respects this interval. In parallel, if the super shape of the group *:GroupLesson_has_students_sup*

is also active, since it contains the sub shape, it'll accept all the instances that were accepted by the sub shape and it'll possibly accept more instances if there exist group lessons with more than 6 students (up to 10). In other words, the super shape would be accepting redundant and not expected data which is being already rejected by the sub shape. Next, let's imagine that the current active context expects that the amount of students for a group lesson is in the interval [3,10]. In this case, the super shape *:GroupLesson_has_students_sup* would be perfect for validating this context. However, if the sub shape *:GroupLesson_has_students_sub1_1* is also active, it would be restricting an interval of students (from 7 to 10) which is accepted in the context. This happens because the sub shape is stricter than the super shape, and this would mean an incoherence between the group and the context. In consequence, in order to avoid the validation of redundant shapes or even to evade the validation of shapes which are stricter than expected, which would give unexpected validation results, it's necessary to do a configuration of the IRM before it's used for the actual validation of an RDF data graph.

In order to use the IRM in a correct and efficient way, only one shape Sx per group GSti should be active. As already mentioned in the preliminaries, the SHACL element sh:deactivated allows to deactivate a shape Sx by linking it with the boolean value True, so the configuration process manipulates this element in a group of shapes GSti, leaving only one shape Sx of the group with the value sh:deactivated False. Put differently, at the validation time, only a sub-graph of the IRM will remain active, having only one shape per *group GSti*.

In this thesis there's a proposal for a pre-validation configuration of the IRM which takes into account the characteristics of the current active context. The way in which the characteristics of the context (including their current status) are represented, in addition to the decisions which must be taken to the definition of the link between the current context and the final active IRM sub-graph, could be represented in several ways. However, in this thesis there's one specific representation of this workflow which is presented and involves different models and steps or tasks.

IRM workflow description

This workflow can be seen in the figure 4.4 and its description follows. The IRM Manager is the core system which interacts with all the parts of the workflow. Furthermore, in this environment, it's expected to have a domain ontology, defined by experts, which describes the related domain knowledge, including the existing contexts and the related characteristics. The second expected model is an RDF data graph which has instance data which follows the structure defined in the ontology. Included in these data, a few triples are expected to define which is the context that is currently active. Continuing with the example, the expected context RDF triples would be as detailed in the listing 4.12. These two triples state that the resource *ex:goodWeather* represents the context which is active at the moment. The instance *ex:goodWeather* is of type *ex:contextClass*, meaning a class which describes external contexts that could influence the behavior of a ski resort. Moreover, this instance is related through the property *ex:currentStatus* to the instance *ex:active*, which is of type *ex:Status*. These triple patterns are expected for the definitions of the contexts' status. The tasks of the pre-validation workflow are described as follows:

Task 1 (retrieve current active context): The first task that the IRM Manager does in this proposal, is to get all the current active contexts from the RDF data graph with a SPARQL query which follows the aforementioned context pattern and could be seen in listing 4.13. The execution of this query on the data detailed in the listing 4.12 would retrieve the instance *:goodWeather*. Moreover, it's expected that the ontology in the workflow represents, between other knowledge, some characteristics of the specific contexts. This could be represented in different

ways, while in this proposal, the ontology it's expected to hold property assertions between an individual of type context and an individual of another class which itself holds some characteristics. Following the example, the ontology could have triples such as *(GroupLesson, hasContext, goodWeather)*, *(GroupLesson, equivalentTo, (GroupLesson and (has_students [3,10])))*.

Task 2 (retrieve characteristics related to the current active context): The acquisition of the contextual characteristics is completed with a second SPARQL query executed by the IRM manager, which uses as an input the active context previously retrieved and consults the ontology with a query pattern that could be as the one shown in the listing 4.14. This query retrieves the related characteristics taking into account the ontology patterns in which the characteristics are defined with the owl:equivalentClass property as an intersection with the current class (as shown in the latter example). Furthermore, continuing withe the triples defined in the example, this query retrieves the following results: GroupLesson, has_students, owl:maxCardinality, 10 and GroupLesson, has_students, owl:minCardinality, 3. At this point, the IRM manager already has some characteristics (if there's any) related to the active context. Next, the configuration of the IRM starts (task 3).

Task 3 (IRM filtering, manipulation, and candidate shapes acquisition): The IRM manager has four objectives at this moment: first to find out if there are IRM shapes which accomplish with the characteristics retrieved from the ontology. In case there's one or more shapes which respect explicitly those filters, each of them are activated, meanwhile the rest of the shapes of the same group (if there is any) are deactivated. Examples of the SPARQL queries are shown respectively in the listings 4.15 (this query retrieves a list of node shape IRIs) and 4.16. Second, to activate all those shapes which are unique in their respective *groups GSti*. These shapes can be identified with SPARQL queries which use the *GROUP BY* clause, such as the one shown in the listing 4.17. Third, the active shapes of those groups which currently have more than one shape on this status are listed (namely contextualized candidate shapes). Going back to the example aforementioned, the contextual characteristics retrieved from the ontology states that a group lesson should have an amount of students between 3 and 10, which is translated into a query which searches in the group *(groupLesson, hasStudents)* those shapes which have an interval amount of students inside the expected interval [3,10]. If this is applied to the IRM example shown in the listing 4.11, then both shapes would stay as contextualized candidate shapes. Fourth and finally, the IRM retrieves the shapes from the groups which have all the shapes deactivated, meaning candidate shapes of a group which wasn't filtered by the context (namely non-contextualized candidate shapes).

Task 4 (check candidate IRM shapes against the RDF graph): Following, the task 4 takes place, where the candidate shapes are tested against the RDF data and the results for each shape validation are analyzed. By taking advantage of the containment structure relationship, it's possible to avoid validating each candidate shape of a group, since if the lower level accepts everything and don't generate any validation report, then the result will be the same for the super shapes of this branch (bottom-top manner).

Task 5 (results analysis): The workflow continues with the task 5, in which the IRM manager develops an analysis of the validation results and a creates a report based on them. This report is done by analyzing elements of the validation report such as the ones already described in the preliminaries. The results are grouped by *GSti group* and present a comprehensive report of information such as the number, kind and severity of the errors for each candidate shape of the group. Furthermore, this report details the containment relationship between the shapes of a group.

Task 6 (Report creation and sharing): Following, in the task 6, the report is presented to the user which is aiming to use the IRM for the validation. Based on the report presented, the

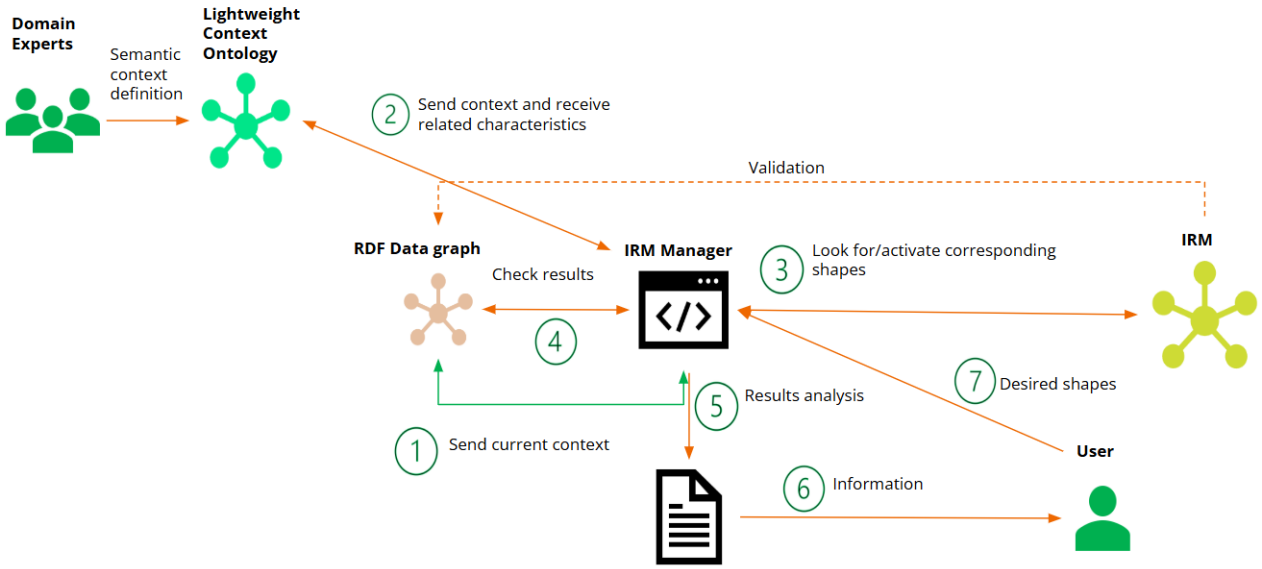


Figure 4.4: IRM pre-validation configuration workflow proposed.

user is expected to select one candidate shape per group and this list of shapes will be received by the IRM manager.

Task 7 (Acquisition of user selections): In this step, the IRM manager receives a list of node shape IRIs which represents the candidate shapes which have been selected by the user.

Task 3 second execution (selected candidate shapes activation): Finally, as the last step, a second execution of the task 3 is done, but this time only executing the activation of the selected candidate shapes and the deactivation of the discarded ones. At this point, the IRM is ready for validating the RDF data graph. An implementation of this pre-validation proposal and its evaluation are presented in the evaluation chapter.

Listing 4.12: RDF triples which define a context which is currently active.

```
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:goodWeather a ex:Context .
ex:goodWeather ex:currentStatus ex:active .
```

Listing 4.13: SPARQL query pattern proposed for retrieving the current active context. This query looks for all the resources of type "Context" which have the property "currentStatus" linking it to the resource "active".

```
PREFIX : <http://localhost/ontology/esf_ontology/>

SELECT ?subject
WHERE { ?subject a :Context ;
:currentStatus :active . }
```

Listing 4.14: SPARQL query pattern proposed for retrieving the characteristics of the current active context. This query first looks for those classes which are related to the current active context (lines 6-7-8-18). Then the query searches for the owl:equivalentClass definition for the retrieved classes and looks for the properties defined in the second part of this definition which are not standard vocabulary definitions (lines 9 to 17 and 19-20).

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX owl: <http://www.w3.org/2002/07/owl#>
3  PREFIX : <http://localhost/ontology/esf_ontology/>
4
5  SELECT ?class ?path ?property ?constraint2
6          WHERE { ?subject :related_to_context ?context .
7                  ?subject rdf:type ?class .
8                  ?context rdf:type :Context .
9                  ?class owl:equivalentClass ?characteristics .
10 ?characteristics owl:intersectionOf ?Bnode1 .
11 ?Bnode1 rdf:rest ?Bnode2 .
12 ?Bnode2 rdf:first ?constraint1 .
13 ?constraint1 ?property ?constraint2 .
14 ?constraint1 owl:onProperty ?path .
15 FILTER( regex(str(?class), "^(?!http://www.w3.org/2002/07/owl#).+")) .
16 FILTER( regex(str(?class), "^(?!http://www.w3.org/1999/02/22-rdf-syntax-ns#).+")) .
17 FILTER( regex(str(?class), "^(?!http://www.w3.org/2000/01/rdf-schema#).+")) .
18 FILTER( str(?context) != ?activeContext)  . # ?activeContext value comes from an input
       parameter.
19 FILTER NOT EXISTS { ?constraint1 rdf:type ?constraint2 . }
20 FILTER NOT EXISTS { ?constraint1 owl:onProperty ?constraint2 . }
21 }
```

Listing 4.15: SPARQL query pattern proposed for retrieving the IRM shape which accomplishes to the characteristics of the current active context. This query looks in the IRM those node shapes which are part of a particular group (lines 4 to 7 and 10-11). The query also restricts the related constraints' types and values to accomplish the ones retrieved from the context ontology (lines 8-9 and 12). The line 13 represents a filter which is generated dynamically depending on the specific constraint retrieved from the ontology.

```
1  PREFIX sh: <http://www.w3.org/ns/shacl#>
2
3  SELECT ?nodeShape
4  WHERE { ?nodeShape a sh:NodeShape .
5  ?nodeShape sh:targetClass ?focusClass .
6  ?nodeShape sh:property ?blankNodePath .
7  ?blankNodePath sh:path ?path .
8  ?blankNodePath ?constraintProperty ?constraintValue .
9  OPTIONAL{?constraintValue ?constraintTypeValueCompound ?constraintValueValueCompound}
10   FILTER( str(?focusClass) = str(?varInputClass) ) .  # ?varInputClass is an input
         parameter.
11   FILTER( str(?path) = str(?varInputPath)  ) .  # ?varInputPath is an input parameter.
12   FILTER( str(?constraintProperty) = ?irmConstraintType && ?constraintValue <= ?
         owlConstraintValue) # ?irmConstraintType and ?owlConstraintValue are input
         parameters.
13   }
```

Listing 4.16: SPARQL query pattern proposed for activating a specific IRM shape. This query looks for a specific IRM node shape. If this shape is deactivated then it's activated.

```
PREFIX sh: <http://www.w3.org/ns/shacl#>

DELETE { ?nodeShape sh:deactivated false . }
INSERT { ?nodeShape sh:deactivated true . }
WHERE
  { ?nodeShape sh:deactivated false .
    FILTER( str(?nodeShape) = str(?nodeShapeIRI) ) . # ?nodeShapeIRI is an input
        parameter
  }
```

Listing 4.17: SPARQL query pattern proposed for detecting those shapes which are unique in their respective groups GSti.

```
PREFIX sh: <http://www.w3.org/ns/shacl#>

SELECT ?nodeShape
WHERE { ?nodeShape a sh:NodeShape .
?nodeShape sh:targetClass ?focusClass .
?nodeShape sh:property ?blankNodePath .
?blankNodePath sh:path ?path .
}
GROUP BY ?focusClass ?path
HAVING(COUNT(?nodeShape) = 1)
```

4.2.3 Source provenance representation

The source provenance knowledge of each generated IRM shape, which was obtained through the integration of two shapes, allows to have the knowledge related to how the shape was obtained, meaning how was the specific integration process related to the specific shape. Furthermore, having a representation of this knowledge could give the possibility to rebuild the original source shapes related to an IRM shape. In order to keep the contribution of this thesis attached to the SHACL standards, the proposal for the representation of the source provenance knowledge related to the integrations, involves the creation of an external lightweight ontology, which is populated each time that an integration is finished (see figure 4.5), obtaining as a result, a related KG. More specifically, the provenance ontology has a structure for the semantic representation of the integration, having the classes: Integration, Status, IRMshape, SourceShape, Constraint, SourceShapeGraph and the object properties has_status, generatedShape, first_inputShape, second_inputShape, has_constraint and sourceGraph. The link between the Provenance Ontology and the IRM will be the IRI of a specific shape which in the ontology is an instance of the class IRMshape. Moreover, the instance of an integration has a current status which could be Active or Inactive. This status represents the last integration associated with the specific generated IRM shapes. When a new integration updates a shape IRI which had already a related integration in the provenance ontology, then before adding this new integration, the status of the previous one is changed to Inactive. A detail of each class and object property of the provenance ontology follows:

- **Classes:**
- *Constraint*: this class represents constraints combinations as (constraintType constraintValue). These constraints are represented in SHACL syntax.
- *Integration:* represents an integration done between two input SHACL shapes which belong to two different source shapes graphs.

- *SHACLshape*: represents a SHACL shape.
- *IRMshape*: represents the shapes generated as an outcome of an integration process. Furthermore, these shapes belong to the Integrated Restriction Model (IRM).
- *SourceShape*: represents a SHACL shape which is one of the inputs of an integration.
- *SourceShapeGraph*: this class represents a SHACL shapes graph which is the source of an Integration.
- *Status*: represents the status of an integration, which could be Active or Inactive. It allows to represent if a specific Integration is the last one related to the specific generated shapes (Active), or if it's a previous related integration (Inactive).
- **Object Properties:**
- *first_ inputShape*: relates an integration with a SHACL shape which takes the first place in the integration algorithm. This property has the class *Integration* as the domain and the class *SourceShape* as the range.
- *second_ inputShape*: relates an integration with a SHACL shape which takes the second place in the integration algorithm. This property has the class *Integration* as the domain and the class *SourceShape* as the range.
- *source_ shapeGraph*: relates a SHACL shape which is a source of an integration with the respective SHACL shape graph to which it belongs to. This property has the class *SourceShape* as the domain and the class *SourceShapeGraph* as the range.
- generated_shape: relates an integration with the IRMshapes that it generated as an outcome. This property has the class *Integration* as the domain and the class *IRMshape* as the range.
- *has_ constraint*: relates a SHACL shape with a representation of its respective constraints. This property has the class *SHACLshape* as the domain and the class *Constraint* as the range.
- *has_ status*: represents the status of an Integration. If an Integration is Active, this means that it's the integration which generated the latter version of the output shapes. If the status is Inactive, then this means that the output shapes have another more recent related Integration. This property has the class *Integration* as the domain and the class *Status* as the range.

In order to show through an example how this representation would look like, first the figure 4.6 describes the integration of two input shapes which resolves a conflict of unidirectional contained type for sh:maxCount SHACL constraint. This specific integration type is explained in the following chapter, however, in this chapter it's just used to exemplify how an integration is represented in the source provenance ontology. In addition, the knowledge related to the aforementioned integration is represented in the Provenance Ontology structure as shown in the figure 4.7. The Source Provenance lightweight ontology also openly shared in the general repository[16].

[16]https://github.com/marianojf89/Thesis/tree/main/Integration%20procedure/Source%20provenance (last accessed on 2/11/2025)

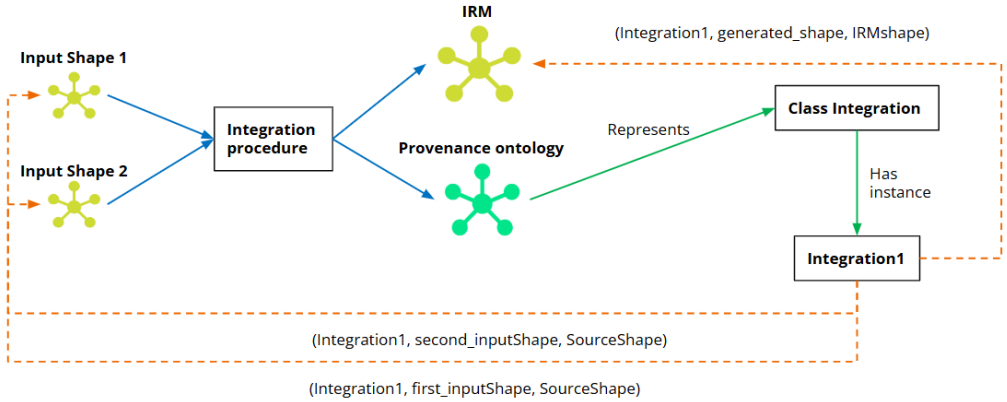


Figure 4.5: General workflow of the source provenance knowledge acquisition. The provenance ontology represents instances of integrations which links a shape from the IRM with the two shapes that were the respective sources of the specific integration.

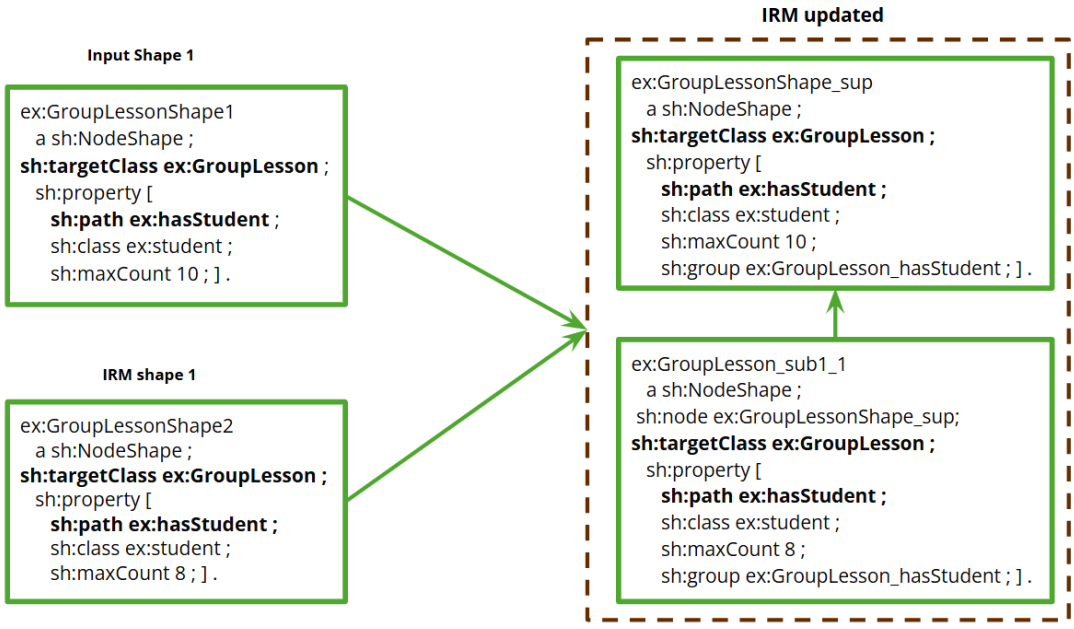


Figure 4.6: Integration of two input SHACL shapes which resolve a conflict of unidirectional contained type for sh:maxCount SHACL constraint.

```
:Integration1 a :Integration ;
:Integration1 :has_status :Active  ;
:Active a :Status ;
:Integration1 :generatedShape ex:GroupLessonShape_sup ;
:Integration1 :generatedShape ex:GroupLessonShape_sub1_1 ;
:Integration1 :first_inputShape :InShape1 ;
:Integration1 :second_inputShape :InShape2 ;
:InShape1 :has_constraint :[sh:class ex:student];
:InShape1 :has_constraint :[sh:maxCount 10] ;
:InShape2 :has_constraint :[sh:class ex:student] ;
:InShape2 :has_constraint :[sh:maxCount 8] ;
ex:GroupLessonShape_sup :has_constraint :[sh:class ex:student] ;
ex:GroupLessonShape_sup :has_constraint :[sh:maxCount 10] ;
ex:GroupLessonShape_sub1_1 :has_constraint :[sh:class ex:student] ;
ex:GroupLessonShape_sub1_1 :has_constraint :[sh:maxCount 8] ;
ex:GroupLessonShape_sup a :IRMshape ;
ex:GroupLessonShape_sub1_1 a :IRMshape ;
:InShape1 a :SourceShape ;
:InShape2 a :SourceShape ;
:[sh:class ex:student] a :Constraint ;
:[sh:maxCount 10] a :Constraint ;
:[sh:maxCount 8] a :Constraint ;
:InShape1 :sourceGraph :RDB1shapeGraph ;
:InShape2 :sourceGraph :RDB2shapeGraph ;
:RDB1shapeGraph a :SourceShapeGraph ;
:RDB2shapeGraph a :SourceShapeGraph ;
```

Figure 4.7: Example of the knowledge related to an integration in the source provenance ontology proposed. The resources in green represent classes, the resources in black represent properties and the resources in violet represent individuals.

4.3 Conclusion

In this chapter, the importance of a SHACL shape graph design which is able to validate multiple contexts, has been discussed. Aligned with this need, **the Integrated Restriction Model (IRM) has been proposed based on the SHACL standards.** This shape graph design is **capable of representing groups of shapes which aim to the same target** (shapes taken as equivalent) **with different restrictions that could hold inter-shape conflicts**. Furthermore, **the presented model organizes the equivalent shapes in a hierarchical structure that follows the containment relationship between them**. In addition, **a proposal for a pre-validation configuration process of the IRM model has been presented**. This proposal includes a set of steps which **involves a set of SPARQL queries used for consulting and manipulating the models** of the environment, having as the final objective, the activation of a sub-graph of the IRM which best answers to the current context characteristics and to the user's desires. Besides, **the Provenance Ontology structure has been presented, which is designed to represent the knowledge related to each specific integration that generated as an outcome a specific IRM shape**. Through the population of this ontology, it's possible to link the IRM specific shapes with their respective source provenance knowledge. Finally, the IRM model has been also defined as the result of the integration of a set of individual SHACL shape graphs related to a specific environment. In the following chapter, a technique for the integration of multiple SHACL shape graphs which resolves the conflicts by shapes specialization and generates as an outcome the creation of an IRM model is presented.

5

Integration of SHACL shape graphs with shapes specialization

Contents

5.1 Introduction 83
5.1.1 Motivation 83
5.1.2 Preliminaries 85
5.2 Integration procedure 86
5.2.1 Conflict Resolution and Integration Process 88
5.3 Conclusion 92

5.1 Introduction

This chapter is focused on the design of an approach for the guidance on the integration of more than one SHACL shape graph. The proposal includes a strategy for the resolution of constraint conflicts which are part of shapes that are aimed to be integrated. In order to define the required action, this strategy includes an analysis of the containment relationship for those shapes which are being compared. Based on this, the proposed strategy resolves the conflicts by the generalization and specialization of shapes, generating as an outcome, an Integrated Restrictions Model, which structure has been detailed in the previous chapter.

5.1.1 Motivation

The creation of SHACL shape graphs could be a highly time-consuming task and easily error-prone. Some semi-automatic and automatic generation techniques, which aim to automatize the creation of these graphs and increase their adoption, can be found in the literature (see 3.5.11). Furthermore, these techniques cover different types of sources (such as ontology, RDB, between others) from which the shape graphs are extracted. These sources which are taken into account as inputs of the shapes extraction techniques, frequently contribute to the construction of an RDF KG by providing with schema or instance data. In this scenario, it would be very valuable if the generated shape graph takes into account the characteristics that are represented in each of the heterogeneous sources. However, one important limitation of the aforementioned extraction techniques is that they take as an input, only sources of one specific type (e.g. the extraction

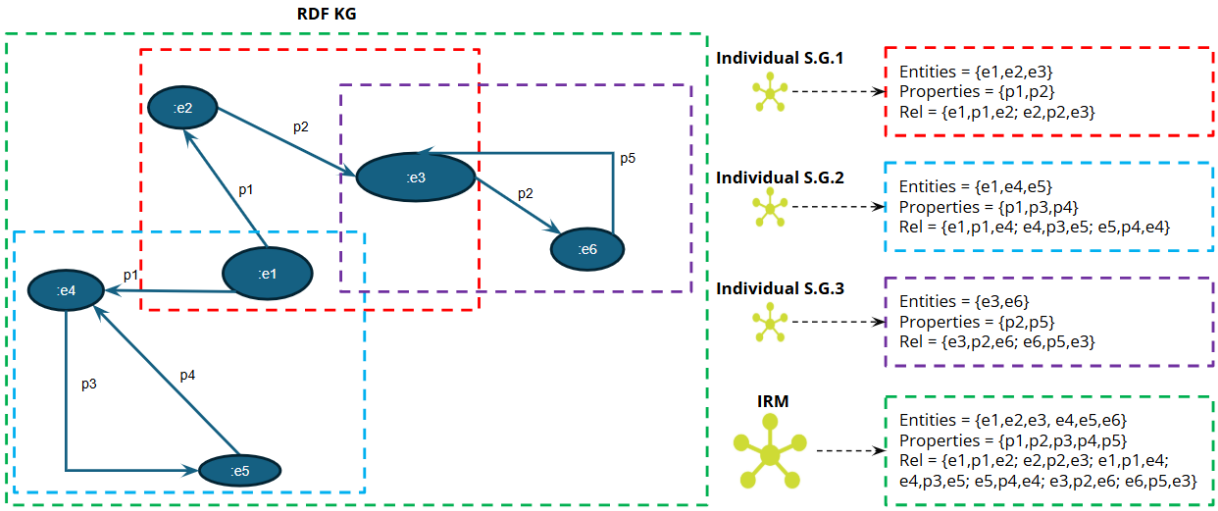


Figure 5.1: Motivation for the integration of individual shape graphs which aim to a general RDF knowledge graph. *"S.G."* states for *"Shape graph"*. This image illustrates the target of each shape graph.

of a shape graph from an ontology). In consequence, it's very likely that when working on the introduction of the SHACL shape graphs into an existing system which presents heterogeneous sources, then more than one individual SHACL shape graph will be generated. For example, if the existing architecture presents a relational database and an ontology, then two different specific techniques could be applied respectively to each of these sources, generating two different SHACL shape graphs. This could create as an outcome, the possibility of having several separated individual SHACL shape graphs. Moreover, each of these individual shape graphs will represent a set of shapes with a set of constraints which aims to a specific subgraph of the general RDF graph that comprehends the whole environment. In contrast, having a unified SHACL shape graph would allow to target the complete set of entities and properties at once (see an example comparison in figure 5.1). In order to move forward with the generation of one unified SHACL shape graph from an existing environment with heterogeneous sources, it's necessary to propose techniques for the integration of the individual shape graphs which are extracted from each source.

The integration of shapes presents some interesting challenges such as the definition of the characteristics that are taken into account in defining the equivalence between shapes and the design of an effective procedure that is able to detect them. Furthermore, once two equivalent shapes are aimed to be integrated, the compatibility between the constraints that belong to each shape must be analyzed, since it's expected that a number of conflicts could appear, creating a scenario in which decisions must be taken. As a consequence, in order to manage the conflict resolution, the definition of a strategy which contemplates every possible scenario is necessary. Finally, the definition of this technique involves also the desired characteristics for the integrated SHACL shape graph that is obtained as the outcome. The integration technique is expected to generate an IRM, which allows to validate specific parts of the RDF graph in different ways, reflecting the variety of characteristics that were found on the individual shapes graphs for a specific part of the graph.

As previously mentioned, there is only one proposal which includes a technique for the integration of multiple SHACL shape graphs ([50]). However, the richest integration of this proposal

includes the resolution of conflicts by disjunction, which generates softer shapes in the outcome. In addition, the generated model is not able to answer to contexts with different characteristics as the IRM does.

5.1.2 Preliminaries

The integration approach presented in this chapter has some common points with the IRM presented in the previous chapter. For this reason, the preliminaries already detailed in the section 4.1.2, hold for this chapter.

Constraints integration and conflict resolution

The *constraint integration and conflict resolution* involves the integration of those constraints which are related to the shapes already stated as equivalent. This task includes some cases in which a conflict could be found, meaning that two constraints which are meant to be integrated can't be included in the same shape, since having both constraints would result in an ill-formed shape. Furthermore, these constraints could have a relationship such as contradictory, not compatible, or they could even have some kind of relationship that creates space for taking design decisions. Those constraints which are contradictory present a target space that is the opposite to the other. For example, a shape *s1* could present a determined property *p1* as mandatory, meanwhile the equivalent shape *s2* could present as a definition that it's not allowed to have the property *p1* related (*not(p1)*). Furthermore, two equivalent shapes could present two constraints which are not compatible, meaning that it's not possible to have both constraints at the same time. An example of this could be presented if both equivalent shapes *s1* and *s2* have the same property *p1* attached but with the cardinality constraint *minCount* with two different values. In some cases, the constraints present also a containment relationship, meaning that the space of values allowed by one constraint is included and extended into the space of values allowed by a second constraint. For example, the amount of relationships that a cardinality constraint *minCount = 2* allows is the space *[2,∞[*, meanwhile another constraint defined on the same relationship *minCount = 1* would allow the space *[1,∞[* which contains the first one. In order to resolve these conflicts and decide which will be the result of the integration, some strategies could be followed in order to define a number of possible actions depending on the specific scenario presented. As already stated in the state of the art (3.7.1), in [173] and [28], a number of these strategies are defined and described. It's worth to highlight that these works are outside the domain of Semantic Web, and that they are related to other types of databases. More precisely, the conflict resolution strategy that has more relevance for this thesis contribution is related to the integration of constraints in incremental object-oriented database design. These conflict resolution strategies vary from leaving the decision to an expert, keeping the less restrictive constraints (creating a softer model), the creation of a restriction model with specialization and more. Specifically, the conflict resolution strategy that has more relevance for this thesis contribution is the *"Resolution by Specialization"*, presented in [173]. This technique resolves the constraint conflicts by specializing and generalizing the object on which the conflict takes place. It's worth to highlight that this technique has been defined taking into account constraints in general, but it doesn't present an application or specific definition for SHACL. Furthermore, the strategy is defined as follows:

Having two constraints *C1* and *C2* which aim to the same *target object O*, the conflict resolution by specialization comprehends the following four scenarios' resolutions:

1. *C1* is *identical* to *C2*, meaning *C1 = C2*. Then, *C1* is added to *O*.

2. *C1* and *C2* are *contradictory*, meaning that *C1 = not(C2)*. Then *O* is *specialized* into *two sub-classes O1 and O2*, with the corresponding *C1* and *C2* constraints associated.

3. *C1* and *C2* are *independent*, meaning *C1 != C2* and *C1 != not(C2)*. In this case, both constraints are added and associated with *O*.

4. *C1* and *C2* are *compatible*, meaning that one constraint is *stronger* than the other. *C1 > C2* if *C1* is *stronger* than *C2*. Then, the *object O* is *specialized* into one *sub-class O1*. The stronger constraint *(C1)* is associated to the *sub-class O1*. The other constraint, *C2*, is added to *O*.

The aforementioned strategy has been used for inspiring the creation of the constraint conflict resolution strategy that will be presented in the following section as a part of this contribution of the thesis.

5.2 Integration procedure

As a result of this procedure, the IRM will contain the integration of the characteristics found in multiple shape graphs related to a determined domain. In addition, for those shapes that aim to the same target Ti, instead of having a unique generated shape as in the state of the art, this approach allows to create a set of shapes that target Ti with conflicting constraints included, which are organized in a hierarchical way. Moreover, these hierarchies follow a structure in which a sub shape is contained in its super shape. The IRM representation allows to describe a set of different combinations (shapes) of source constraints which aims to the same target Ti, keeping the possibility to activate the desired combination and level of hardness before validating Ti in a Dg. The IRM is fed sequentially, iteration by iteration, each time that a new shape graph is processed. The first ever iteration will be done with an empty IRM, meaning that in this case, the input shape graph is directly added to the IRM with the required IRM data to respect the defined structure (e.g. having the sh:group added). In this way, the IRM knowledge will be probably extended with each new iteration. A representative image of this process can be seen in the figure 5.2. Moreover, the integration procedure can be described as follows: The *set of shape graphs SSG* is desired to be integrated into the *IRM*. Sequentially, *one shape graph at a time* is treated. Starting by *Sg1*, an *input shape graph* from *SSG* with a *set of shapes S1*. If there's no coincidence between the *target identifiers Ti1* of each *shape* belonging to *S1* and the *target identifiers Ti2* found in the *set of shapes S2* belonging to the *IRM*, then the shapes from *S1* and *S2* are directly added to the *updated IRM model* (there's no intersection or no equivalence between the shapes). It's worth to mention that the shapes from *S1* will need a simple treatment for adapting it to the *IRM* structure. Otherwise, those shapes from *S1* and *S2* that have an *intersection* in the *target identifiers* (meaning that they are taken as equivalent shapes) will be sent to the *Conflict Resolution and Integration Process (CRIP)*. A simple picture of a workflow which describes the integration procedure is shown in the figure 5.3.

The equivalence detection task has been already described in the previous chapter (4.1.2). As a simple reminder, two shapes are considered equivalent if both of them have the same targets defined, meaning focus node, and in case that the node shape has a nested property shape, the path value is also taken into account. As already mentioned, those shapes which are part of an input shape graph and don't have an equivalence in the current version of the IRM, would need a slight adaptation to the IRM structure, which includes the adaptation of the respective node shape IRI to the pattern defined *(:{fn}_{pv}_{hierarchyLevel_{branchNumber}})*, the

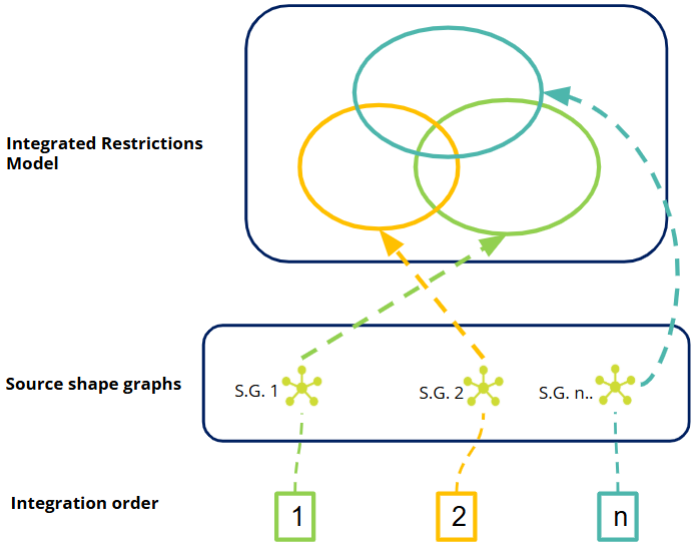


Figure 5.2: The IRM knowledge is extended sequentially with each new integration of each SHACL shape graph. The bottom part of the figure shows the order in which the shape graphs are integrated with the IRM (e.g. the shape graph 1 goes first). The section in the middle of the image shows a set of source shape graphs which are desired to be integrated with the IRM. Finally, the top of the picture shows a Venn diagram which represents the universe of restrictions held by the IRM. The intersections in the diagram represent those restrictions found in equivalent shapes which were integrated.

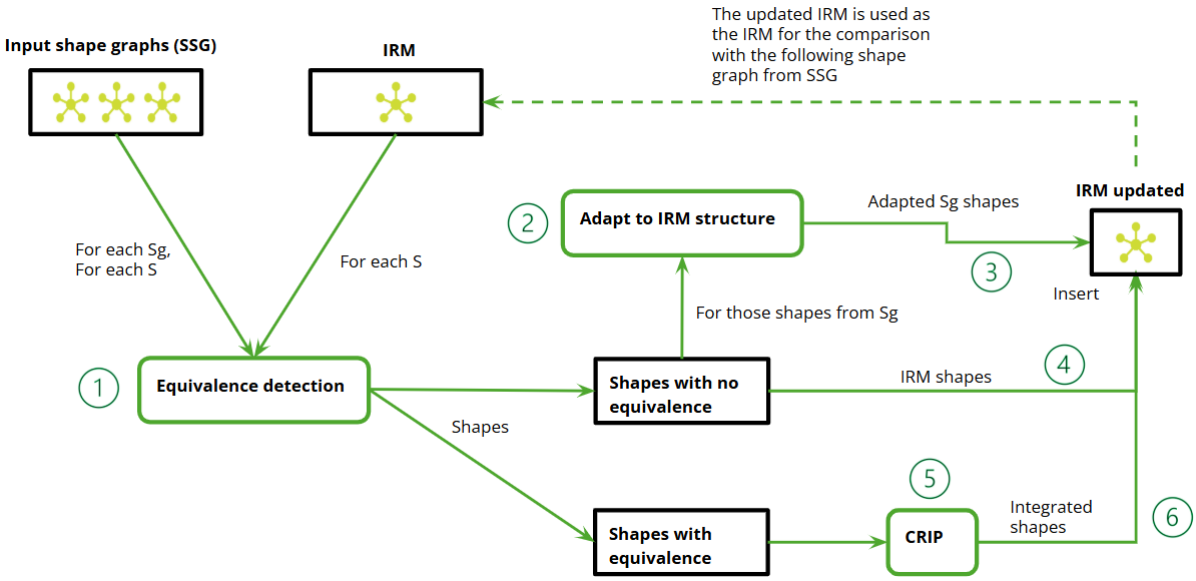


Figure 5.3: The integration procedure's main workflow. The black rectangles represent the sources. The green rounded rectangles represent procedures. The numbers in green circles indicate the order in which the workflow is executed. CRIP stands for "Conflict Resolution and Integration Process".

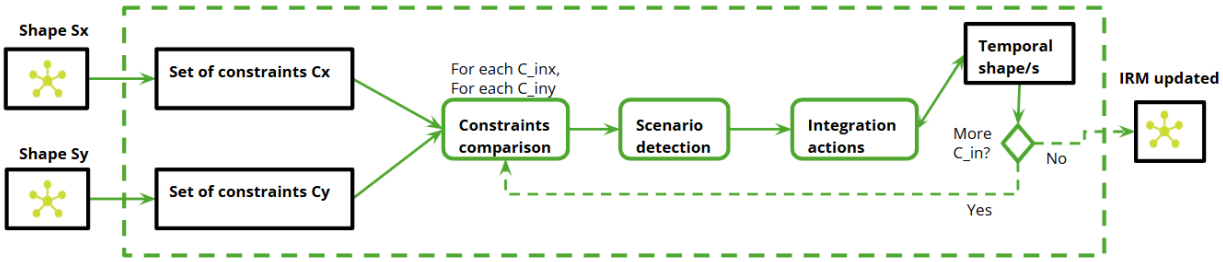


Figure 5.4: The "Conflict Resolution and Integration Process" main workflow.

introduction of the property *sh:deactivated* with value *true*, the definition of the new related *sh:PropertyGroup* and the link of the nested property shape to the new group with the element *sh:group* (the last two points are done only if the shape is a compound one, meaning that it's a node shape with a nested or referenced property shape). On the other hand, if two shapes are found equivalent, they are sent to the *"CRIP"* which is described in the following section.

5.2.1 Conflict Resolution and Integration Process

The CRIP is a procedure which objective is to integrate two equivalent shapes. More specifically, the process receives two well-formed shapes, Sx and Sy. The *shape Sx* belongs to a specific *shape graph Sg* from the input *set of shape graphs SSG* where *Sx* has a *target identifier Ti*. In addition, the *shape Sy* belongs to the current version of the *IRM* and has the same *target identifier Ti*. In addition, *Sx* has the *set of constraints Cx* meanwhile *Sy* has the *set of constraints Cy*. Going forward, the CRIP compares each constraint of *Sx (C_ inx)* against each constraint of *Sy (C_ iny)*. This comparison takes into account the types of constraints that are *C_ inx* and *C_ iny*, their respective *values* and the *containment relationship* between them. The containment relationship interpretation between shapes is further detailed later in this chapter. Furthermore, the specific relationship that the two constraints hold, is linked to a set of previously defined scenarios which have attached a set of rules or actions for the integration resolution. Based on these resolutions, a temporal group of shapes which respects the IRM structure is constructed iteration by iteration, until the complete set of constraints of both shapes Sx and Sy is treated. It's worth to mention that the actions defined take also into account the current temporal structure of shapes. Finally, the process finishes by inserting the final constructed group of shapes in the updated version of the IRM. The CRIP workflow is shown in the figure 5.4.

Behavior of specific SHACL constraint types in the containment evaluation

The behavior of the integration and conflict resolution procedure presented in this thesis depends also on the specific SHACL constraints which are involved, such as sh:nodeKind, sh:minCount, sh:maxLength, sh:hasValue, etc. Moreover, as already mentioned, some of these elements could hold specific predefined values, which present a containment relationship between them, meanwhile other types of constraints hold integer values and the containment evaluates the relationship between integer intervals. Furthermore, there are other types of constraints which accepts other types of values. In other words, the evaluation of the containment relationship depends on each specific type of constraint and on the type of values that this constraint holds. A related detail which shows which types of values are taken for the containment relationship evaluation for each type of constraint, along with a simple example for each of them, is shown in the tables 5.1, 5.2,

Constraint type	Compares	Example
Value type – sh:datatype	xsd definitions (types derivations)	xsd:integer is derived from xsd:float = xsd:float is softer than xsd:integer.
Value type – sh:nodeKind	Predefined values	sh:IRI is contained in sh:IRIorLiteral
Value type – sh:class	Specific defined classes	sh:class [class1] != sh:class [class2]
Cardinality - sh:minCount	Integer interval comprehended	sh:minCount 2 = [2,∞) is contained in sh:minCount 1 = [1,∞)
Cardinality - sh:maxCount	Integer interval comprehended	sh:maxCount 6 = [0,6] is contained in sh:maxCount 10 = [0,10]
Value Range - sh:minExclusive	Integer interval comprehended	sh:minExclusive 2 = (2,∞) is contained in sh:minExclusive 1 = (1,∞)
Value Range - sh:maxExclusive	Integer interval comprehended	sh:maxExclusive 6 = [0,6) is contained in sh:maxExclusive 10 = [0,10)
Value Range - sh:minInclusive	Integer interval comprehended	sh:minInclusive 2 = [2,∞) is contained in sh:minInclusive 1 = [1,∞)

Table 5.1: A detail of the type of values that are taken into account for the containment relationship evaluation in different specific constraints.

5.3 and 5.4. In these tables, the column "Constraint type" defines the specific constraint that is being detailed. The column "Compares" describes what is used at the time of contrasting the values assigned to the constraints. Finally, the column "Example" provides a demonstration of how the relationship between the constraints is evaluated.

Constraint conflict resolution by shapes specialization scenarios definition

The definition of the scenarios that could be found in the constraints comparison task, as well as the corresponding resolution actions that are linked to each scenario, is inspired in the constraint conflict resolution strategy "Resolution by specialization", which was already mentioned in the preliminaries of this chapter. Based on this technique, the following adaptation to shapes is defined:

Having a constraint C_inx which belongs to the shape Sx from the input shape graph and the constraint C_iny which belongs to the shape Sy which belong to the current version of the IRM. The integration and conflict resolution procedure presents the following general actions depending on the five possible expected general scenarios:

1. **Identical:** C_inx and C_iny are the same type of constraint and their value spaces are contained in each other, meaning that $C_inx \subseteq C_iny$ and vice-versa (C_inx = C_iny).

   a. Then C_inx is added to the output shape S.

2. **Contradictory:** a *sh:not* constraint is detected in the input shape Sx, more precisely in the constraint C_inx (which is inside the *sh:property*), then the referenced shape *s1* (as *sh:not s1*) is taken and compared against any shape referenced in the current IRM shape Sy (again, each C_iny inside the corresponding nested sh:property). If the same shape is referenced first in C_inx as (*sh:not s1*) and then in C_iny as (*sh:node s1*), then there's a *contradiction*. In this case, the containment relationship is defined as $C_inx \nsubseteq C_iny$ and $C_iny \nsubseteq C_inx$. Both references to the shape *s1* should be linked with an *sh:or* in

Constraint type	Compares	Example
Value Range - sh:maxInclusive	Integer interval comprehended	sh:maxInclusive 6 = [0,6] is contained in sh:maxInclusive 10 = [0,10]
String Based - sh:minLength	Integer interval comprehended	sh:minLength 2 = [2,∞) is contained in sh:minLength 1 = [1,∞)
String Based - sh:maxLength	Integer interval comprehended	sh:maxCount 6 = [0,6] is contained in sh:maxCount 10 = [0,10]
String Based - sh:pattern	Regular expressions values (equality)	sh:pattern "^B" != sh:pattern "^A"
String Based - sh:pattern - sh:flags	List of specific defined options, depending on the meaning of each of them.	sh:flags ¨i¨ contains a pattern which doesn't have a flag defined. This is because the ¨i¨generates a softer pattern.
String Based - sh:languageIn	List of values	sh:languageIn (¨en¨) is contained in sh:languageIn (¨es¨, ¨en¨)
String Based - sh:uniqueLang	Boolean values where true adds hardness	sh:uniqueLang true is contained in sh:uniqueLang false

Table 5.2: A detail of the type of values that are taken into account for the containment relationship evaluation in different specific constraints.

Constraint type	Compares	Example
Property Pair - sh:equals	Specific IRI defined	sh:equals [p1] != sh:equals [p2]
Property Pair - sh:disjoint	List of specific IRIs. More values means harder	sh:disjoint [p1, p2] is contained in (harder) sh:disjoint [p2]
Property Pair - sh:lessThan	Specific IRI defined	sh:lessThan [p1] != sh:lessThan [p2]
Property Pair - sh:lessThanOrEquals	Specific IRI defined	sh:lessThanOrEquals [p1] != sh:lessThanOrEquals [p2]
Shape based - sh:qualifiedValueShape, sh:qualifiedMinCount, sh:qualifiedMaxCount	1- sh:qualifiedValueShape: specific shape defined. 2- sh:qualifiedMinCount, sh:qualifiedMaxCount as Integer interval comprehended (min and max respectively).	- sh:qualifiedValueShape [shape1] != sh:qualifiedValueShape [shape2] - sh:qualifiedMinCount 3 = [3,∞) is contained in sh:qualifiedMinCount 1 = [1,∞). - sh:qualifiedMaxCount 10 = [0,10] is contained in sh:qualifiedMinCount 12 = [0,12].

Table 5.3: A detail of the type of values that are taken into account for the containment relationship evaluation in different specific constraints.

Constraint type	Compares	Example
Other - sh:closed	Boolean values where true adds hardness	sh:closed true is contained in (harder) sh:closed false
Other - sh:closed - sh:ignoredProperties	List of properties where adding a property creates a softer constraint	sh:ignoredProperties (p1, p2) contains sh:ignoredProperties (p1)
Other - sh:hasValue	Specific values defined	- sh:hasValue (value1) != sh:hasValue (value2) - sh:hasValue (value1) is contained in [sh:hasValue (value1) or sh:hasValue (value2)]
Other - sh:in	List of specific values defined	- sh:in (value1) != sh:in (value2) - sh:in (value1) is contained in sh:in (value1,value2)

Table 5.4: A detail of the type of values that are taken into account for the containment relationship evaluation in different specific constraints.

the outcome super shape *S_sup*. Since *(s1 U not(s1))* represents the *universal set*, then the super shape won't have neither of these constraints and they are directly represented in the sub shapes. The same scenario applies if the constraints are found in the inverse order (sh:not found in the current IRM shape *Sy*).

a. Then the new outcome super shape *S_sup* is created and structured without the constraints.

b. *C_inx* is added to a new outcome sub shape (*sub1_2*).

c. *C_iny* is added to a new outcome sub shape (*sub1_1*).

3. **Inconsistent:** *C_inx* and *C_iny* are the same type of constraint, *C_inx* $\nsubseteq$ *C_iny*, *C_iny* $\nsubseteq$ *C_inx* and there's no super constraint value *C3* between the *predefined values* for this specific type of SHACL constraint (the latter applies only for sh:nodeKind) which contains *C_inx* and *C_iny* at the same time. The same resolution actions apply also to the scenario in which *C_inx* and *C_iny* are different types of constraints and are defined as not compatible with the SHACL syntax rules. Some of these have been detailed in the figure 5.5 ([50]).

   a. The constraints *C_inx* and *C_iny* are linked through an *sh:or* and assigned to the new outcome super shape *S_sup*.

   b. *C_inx* is added to a new outcome sub shape (*sub1_2*).

   c. *C_iny* is added to a new outcome sub shape (*sub1_1*).

4. **Independent:** *C_inx* and *C_iny* are not the same type of constraint and their value spaces are compatible (*C_inx != C_iny* and *C_inx != not(C_iny)*). In addition, *C_inx* accomplishes this for each constraint of *C_iny*.

   a. An outcome shape S is created.

   b. *C_inx* and *C_iny* are added to S.

5. **Unidirectional contained:** *C_inx* and *C_iny* are the same type of constraint and their value spaces present a containment relationship where *C_inx* $\subseteq$ *C_iny* or the inverse.

   a. IF *C_inx* $\subseteq$ *C_iny*, means also that *C_inx* is stronger than *C_iny*.

      1. A new super shape S is created. *C_iny* is added to S.
      2. *C_inx* is added to a new sub shape (sub1_1).

   b. IF *C_iny* $\subseteq$ *C_inx*.

      1. A new super shape S is created. *C_inx* is added to S.
      2. *C_iny* is added to a new sub shape (sub1_1).

6. **Compatible with super predefined value:** *C_inx* and *C_iny* are the same type of constraint and they are not compatible, meaning that they have values that target different spaces. Furthermore, there exists a super constraint value *C3* between the *predefined values* for this specific SHACL constraint (applies for *sh:nodeKind*), which contains *C_inx* and *C_iny*, such as *C_inx* $\subseteq$ C3 and *C_iny* $\subseteq$ *C3*.

   a. A new outcome super shape S is created. *C3* is added to S.

   b. *C_inx* is added to a new sub shape *(sub1_1)*.

   c. *C_iny* is added to a new sub shape *(sub1_2)*.

It's worth to mention that for those cases in which the resolution action links the integrated constraints with a sh:or, the addition of the latter property depends also on the type of constraint that is being integrated. There are some specific constraints for which this element is not necessary, such as with sh:in, in which case just adding the values to the assigned list (sh:in value1,value2,...) is enough, or as in sh:hasValue, in which case just adding one sh:hasValue for each different value is interpreted as a sh:or between the different definitions for this specific constraint. The detail of the scenarios is summarized in the figure 5.6.

For a matter of clarity, the scenario definitions presented in this chapter comprehend those cases in which the integration of two shapes is starting, meaning that there's still no temporal group of shapes created for the outcome. A complete definition of the actions defined for each scenario, which takes into account the different structures that could be found in the temporal group of shapes of the group, can be consulted in the appendix B.

5.3 Conclusion

In this chapter, **the integration procedure of multiple SHACL shape graphs, which are part of a common environment, has been discussed**. Furthermore, **a related integration procedure has been proposed**, which takes as inputs a SHACL shape graph (which is desired to be integrated) and an IRM SHACL shape graph (which structure has been presented in the previous chapter), which represents the current version of the IRM at the moment. Based on those inputs, the procedure generates as an outcome, an updated version of the IRM SHACL shape graph which could include new knowledge related to the integration of the previous IRM version with the new input processed. As part of this procedure, **a process for the equivalence detection of those shapes which aim to the same *target ti*, has been proposed**. In addition, as part of the integration of those shapes which are detected as equivalent, **a conflict resolution and integration strategy have been described**. In this procedure, the comparison of the constraints takes into account the type of each SHACL constraint, the specific values

Constraints to be added	Inconsistent constraints
sh:datatype	sh:nodeKind *isnot* sh:Literal
sh:nodeKind *is* (sh:IRIOrLiteral *or* sh:IRI *or* sh:BlankNodeOrLiteral *or* sh:BlankNode *or* sh:BlankNodeOrIRI)	sh:datatype *or* sh:languageIn *or* sh:minExclusive *or* sh:minInclusive *or* sh:maxExclusive *or* sh:maxInclusive
sh:minCount	$>$ sh:maxCount
sh:maxCount	$<$ sh:minCount
sh:minExclusive	sh:nodeKind *isnot* sh:Literal *or* $\geq$ sh:maxExclusive *or* $\geq$ sh:maxInclusive
sh:minInclusive	sh:nodeKind *isnot* sh:Literal *or* $\geq$ sh:maxExclusive *or* $>$ sh:maxInclusive
sh:maxExclusive	sh:nodeKind *isnot* sh:Literal *or* $\leq$ sh:minExclusive *or* $\leq$ sh:minInclusive
sh:maxInclusive	sh:nodeKind *isnot* sh:Literal *or* $\leq$ sh:minExclusive *or* $<$ sh:minInclusive
sh:minLength	sh:nodeKind *isnot* sh:Literal *or* $>$ sh:maxLength
sh:maxLength	sh:nodeKind *isnot* sh:Literal *or* $<$ sh:minLength

Figure 5.5: Inconsistency among SHACL constraints defined in [50]

Scenario	Same type of C?	Containment	Compatible	Super pred. C
Identical	Yes	C_in1 ⊆ C_in2 and C_in2 ⊆ C_in1	Yes	N/A
Contradictory	Yes	C_in1 ⊈ C_in2 and C_in2 ⊈ C_in1	No	N/A
Inconsistent	N/A	C_in1 ⊈ C_in2 and C_in2 ⊈ C_in1	No	No
Independent	No	N/A	Yes	N/A
Unidirectional contained – (a)	Yes	C_in1 ⊆ C_in2 and C_in2 ⊈ C_in1	No	N/A
Unidirectional contained – (b)	Yes	C_in2 ⊆ C_in1 and C_in1 ⊈ C_in2	No	N/A
Compatible with sup. pred. value	Yes	C_in1 ⊆ C3 and C_in2 ⊆ C3	No	Yes (C3)

Figure 5.6: A detail of the scenarios defined, which can appear in the comparison of two constraints which belong to equivalent shapes and are aimed to be integrated.

that they have assigned and the containment relationship that holds between them. The detail of how each SHACL type constraint is evaluated with respect to its containment relationship has been also detailed. Based on the results of the three comparisons mentioned, **a set of scenarios has been defined, which in addition have a set of resolution actions attached**. Moreover, these resolution actions construct step by step the related group of shapes which aim to the same *target ti* and are organized in the IRM's hierarchical structure.

6

Experiments and evaluations

Contents

6.1 Introduction 95
6.2 Ski Teaching school use case 95
6.2.1 Integration of the resorts' characteristics 96
6.2.2 Validation of the general system data for multiple contexts 98
6.3 Implementation 98
6.3.1 ESF knowledge representation models 99
6.3.2 IRM pre-validation configuration 100
6.3.3 SHACL shape graphs integration procedure 101
6.4 Experiments and evaluation 102
6.4.1 Integration of two SHACL shape graphs corresponding to two different resorts 102
6.4.2 Activation of the generated ESF IRM related to the weather context . . 104
6.5 Conclusion 107

6.1 Introduction

This section first presents two use cases related to the French Ski School. Furthermore, the implementations of the contributions of the thesis are mentioned. Finally, a set of experiments is carried out in order to evaluate some aspects of the models and procedures proposed in the thesis.

6.2 Ski Teaching school use case

As already mentioned in chapter 1, this thesis is a contribution to the *French Ski School (stands for "École du Ski Français" in french)*[17], which is a French company dedicated to the management of ski resorts in France. The goal of the related project was first to construct a Semantic Web environment which represents the specific domain business with knowledge representation models that involve the instance, conceptual and restriction levels of representation based on

[17] https://www.esf-uk.co.uk/ (last accessed on 2/11/2025)

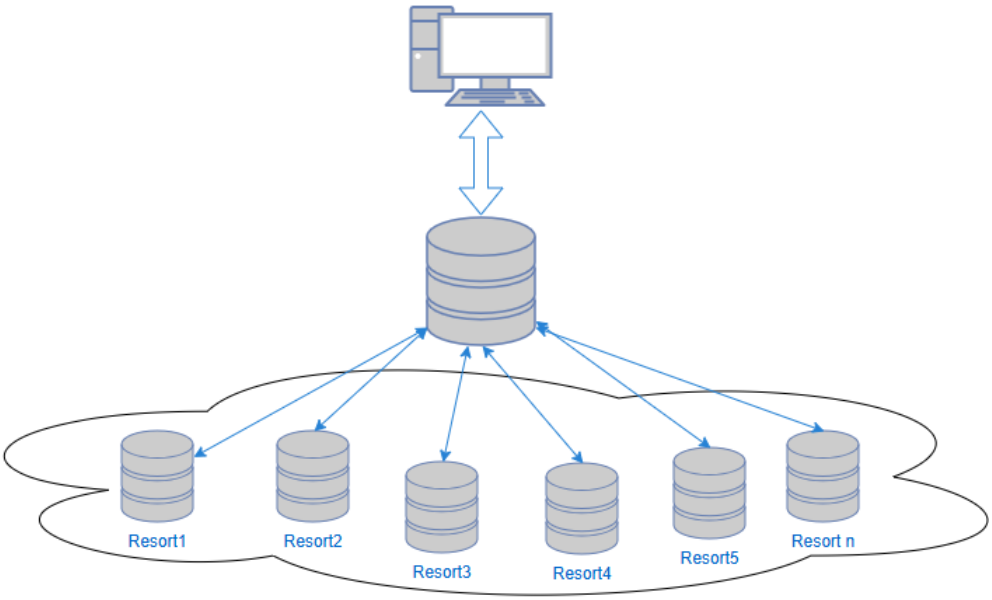


Figure 6.1: A representative image of a simplified ESF architecture which shows how data from all the resorts are centralized.

existing heterogeneous sources. In consequence, Semantic Web models such as an ontology and a knowledge graph (KG) were created.

Moreover, the current main goal of this company consists in providing to their clients with a number of different products, with which it's possible to cover all the facilities and services that they could need for their stay in the mountains. This means that the company offers ski lessons to book, but also other products such as the accommodation, restaurant reservations, ski equipment and more. Furthermore, these offers are done for a number of ski resorts which are placed all over France, and each of these resorts have their own data and their own systems. Additionally, a system centralizes information about the various schools located across all resorts (see the representative image 6.1). While ESF provides products for all these resorts, each resort simultaneously offers its own products independently. Resorts throughout France can exhibit diverse characteristics, many of which may be concealed within the associated data. This diversity extends to the characteristics of the products offered by each resort, leading to potential differences. Nevertheless, these resorts share an underlying formal semantic framework related to the domain. Using the ESF ontology as the foundational semantic structure, an RDF KG could be populated with data from all the resorts.

6.2.1 Integration of the resorts' characteristics

In order to create a shape graph which is able to describe the structure of the knowledge which represents the specific characteristics of each resort, an individual shape graph could be obtained for each of them. However, since ESF manages all the resorts and the RDF KG is fed with data from all of them, the validation of the data through the use of the individual shape graphs could lead to incomplete validations that don't take into account the whole related universe. Additionally, if the shape graph is generated from the general RDF KG, then the possibility of having options as the outcome of the integration of all the differences in the characteristics of the

individual resorts would be lost. At this point, the integration procedure presented in chapter 5 comes to the scene.

By processing the integration of the shape graphs which represent the individual resorts' characteristics, the creation of an Integrated Restriction Model will first ensure to ESF, that this model will cover the whole picture by the representation of restrictions that aim to the complete set of classes and properties. In addition, through the creation of hierarchies, a representative set of different combinations of constraints will be kept available for the validation.

In summary, ESF obtains as a benefit, a restriction model which has a combination of the restrictions of each resort, keeping all those shapes which aim for different targets and integrating in a comprehensive way those shapes which aims to the same targets, creating and keeping more than one option if there was a conflict between the respective constraints. For example, if the Annecy and Chamonix resorts shape graphs presented a difference in the amount of students that a group lesson could have, then the IRM will have as a result, more than one option for the amount of students allowed in group lessons. Furthermore, these options won't be literally mapped, and in contrary, the intervals of students are integrated in a way that the shapes will reflect a shape with the most restrictive interval combination and another shape with the less restrictive interval combination. In order to present this with a practical example, two input shapes and the result of their integration are shown in the listings 6.1 and 6.2, respectively.

Listing 6.1: Two SHACL shapes which belong to two different shape graphs. These shapes restrict the amount of students that a group lesson could have.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLesson_has_students_source1
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 5 ;
        sh:maxCount 10 ;
        ] .

ex:GroupLesson_has_students_source2
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 4 ;
        sh:maxCount 8 ;
        ] .
```

Listing 6.2: Integration of the shapes presented in the previous listing. The conflict between the constraints sh:maxCount has been resolved putting the softer value in the super shape and the restricter value in the sub shape.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLesson_has_students_sup
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:group ex:GroupLesson_has_students ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 4 ;
        sh:maxCount 10 ;
        ] .

ex:GroupLesson_has_students_sub1_1
    a sh:NodeShape ;
    sh:targetClass ex:GroupLesson ;
    sh:node ex:GroupLesson_has_students_sup ;
    sh:group ex:GroupLesson_has_students ;
    sh:property [
        sh:path ex:has_students ;
        sh:minCount 5 ;
        sh:maxCount 8 ;
        ] .

ex:GroupLesson_has_students a sh:propertyGroup .
```

6.2.2 Validation of the general system data for multiple contexts

The ESF company has to take into account some external conditions that could influence the behavior of the ski resorts which are managed by them. One of these conditions is the weather, which depending on its current characteristics, could change drastically the restrictions recommended for a wide number of activities that are normally done in the resorts. This represents a good scenario for the exploitation of the IRM, which once generated, gives to ESF the possibility to validate the RDF KG differently, depending first on the characteristics of a current external context, and second in the decision of the user which is aiming to do the validation. The variety of options will depend on the specific input restrictions processed and it'll be presented in those groups of shapes which were the result of the resolution of constraint conflicts during the integration, leading to more than one shape aiming for the same target. In addition, if some shape which is unique in its group, generates undesired results, it can be deactivated.

6.3 Implementation

In this section, the implementations and the experiments related to the thesis' contributions are analyzed and described. Furthermore, the procedures and models developed are deposited in subfolders of the main repository[18]. In a matter of clarity, each path of the repository which is mentioned in this section should be taken as a continuation of the main path.

[18]https://github.com/marianojf89/Thesis/tree/main (last accessed on 2/11/2025)

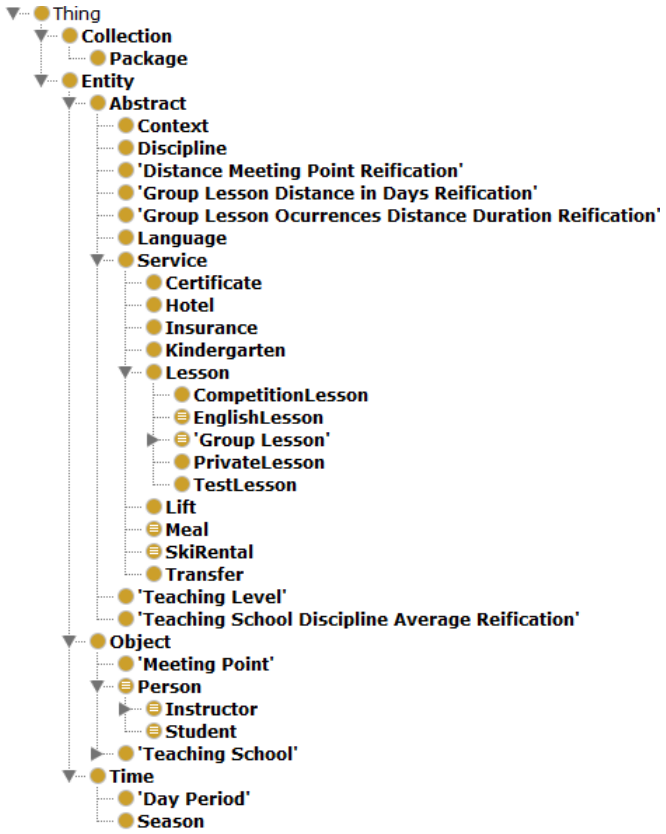


Figure 6.2: Main classes of the ESF ontology.

6.3.1 ESF knowledge representation models

ESF ontology

In order to accomplish the *acquisition of the enterprise domain related knowledge* necessary for the creation of the ontology, knowledge related to the ski resorts domain was obtained from three main sources: interviews with experts that have some position in the company, the existing business models and domain knowledge documents such as ski guides and rules. This ontology aims to represent the knowledge related to ski resorts, the services that they could offer to their tourists and the schools that operate on them. Furthermore, *enterprise ontology creation* was done following the NeOn (Networked Ontologies) methodology[19], which is a scenario-based methodology that provides guidance for many aspects of the ontology engineering process. However, it's worth to mention that not all the detailed steps of the NeON methodology were applicable in the development of the current version. The implementation of the ontology was done using the Protégé tool[20] (version 5.5.0), which is a well-known and open source ontology editor. The main classes of the ontology are shown in the figure 6.2, meanwhile the complete ontology can be found in the "Ontology" repository folder.

[19]http://neon-project.org/ (last accessed on 2/11/2025)
[20]https://protege.stanford.edu/ (last accessed on 2/11/2025)

ESF RDF knowledge graph

Having constructed the ontology, the following step on the project was the acquisition of the enterprise heterogeneous data, which included analyzing, understanding and integrating data from heterogeneous sources. The main data source consisted in an RDB which contains interesting information such as the teaching schools profiles, services, scheduling and more. In addition, extra data was obtained in CSV format representing more information such as a history of the packages ordered. In an attempt to integrate the data in a simple manner, the CSV files were first loaded as tables in the RDB. Once all the data was integrated, the following objective was to feed the ontological schema, by mapping the data contained in the RDB to the defined ontology through the definition of mapping rules. In order to create these mappings, the specific source data is obtained through the creation of SQL queries of different complexities and then related to the corresponding classes and relationships by using the RDB to RDF Mapping Language (R2RML[21]). As a result of the mapping execution and enterprise knowledge graph creation, an enterprise virtual KG is created representing instance data which is structured by the aforementioned ontology. This KG could be kept as a virtual one. Otherwise, the mappings could be materialized and the creation of the KG could be completely explicit. The defined R2RML mappings can be found in the corresponding repository's folder[22].

6.3.2 IRM pre-validation configuration

In order to recall the models and procedures that were defined as part of the IRM pre-validation configuration, it's recommended to consult the figure 4.4. First, the domain ontology recently described, represents the base ontology in the workflow. Through the Context class (shown in the class hierarchy picture 6.2) and the object property *related_to_context*, the representation of specific contexts and their links to specific classes can be represented by simply populating these relationships with the respective individuals. The Context is also related through *has_status* to its current status. This knowledge is provided in a small RDF graph. Moreover, the weather context in which there is a storm is represented with its characteristics in a specific context ontology which only adds these characteristics to the aforementioned base ontology. In addition, this context ontology is used for the experiment that will be described later in this chapter. Next, the RDF Data graph is created as a set of RDF data which follows the ontology structure. With respect to the IRM manager and the different actions that it executes, the implementation of a Python[23] program has been done, in order to connect all the interactions. Furthermore, the rdflib[24] library has been widely used for the manipulation of RDF graphs, which includes the RDF Data graph, the ontology and the IRM. By an adaptation and execution of the SPARQL base queries described in the section 4.2.2, the following tasks have been implemented:

1. Retrieve the active context from the RDF Data graph.

2. Retrieve the ontology sub-graph which has the characteristics related to the active context.

3. Retrieve and activate those shapes from the IRM which accomplish the contextual characteristics. This task includes the translation of elements from OWL to SHACL, more specifically of the properties owl:minCount and owl:maxCount.

[21] https://www.w3.org/TR/r2rml/ (last accessed on 2/11/2025)
[22] https://github.com/marianojf89/Thesis/tree/main/Ontology/Mappings (last accessed on 2/12/2025)
[23] https://www.python.org/ (last accessed on 2/11/2025)
[24] https://rdflib.readthedocs.io/en/stable/# (last accessed on 2/11/2025)

4. Detect and activate those shapes from the IRM which are unique in their groups.
5. Retrieve those candidate node shapes which need a decision from the user, meaning those shapes which are active and are not the unique active shapes in their respective group.
6. Retrieve the shapes from the groups which have all the shapes deactivated, meaning candidate shapes of a group which wasn't filtered by the context.
7. Once the user selected which candidate shapes are decided to have active, process this list of shapes and activate them.

In summary, the implementation comprehends the interaction of the IRM manager with all the involved models, and includes an interesting set of actions for the configuration of the IRM. However, there are parts of the workflow which development is still in progress, such as adding the detection and translation of more constraint types (OWL-SPARQL), the generation of the validation reports, the construction of the reports based on them and finally the interaction with the user through an interface. The implementation's code can be found in the respective repository.[25]

6.3.3 SHACL shape graphs integration procedure

The implementation of the integration procedure has been done by using the same programming language used for the pre-validation configuration (Python[26]). In addition, the rdflib[27] library has provided important functionalities for the manipulation of the IRM. Recalling the integration procedure defined in chapter 5, the following features have been implemented so far:

1. The shapes equivalence detection procedure, including the detection of simple and compound shapes equivalence. It's worth to mention that the implementation recognizes those property shapes which are defined as *sh:property blankNode* and expects a sh:targetClass definition in the defined node shapes.
2. The insertion of those shapes which don't have equivalence in the updated IRM.
3. Integration of constraints for all the expected scenarios related to the structure of the temporal shapes.
    a. Integration of simple shapes for the *sh:nodeKind* constraint.
    b. Integration of compound shapes for all the defined scenarios: *Identical*, *Contradictory*, *Inconsistent*, *Independent*, *Unidirectional contained* and *Compatible with super predefined value*, including the following SHACL constraints: *sh:nodeKind*, *sh:minCount*, *sh:maxCount*, *sh:minInclusive*, *sh:maxInclusive*, *sh:minLength*, *sh:maxLength*, *sh:hasValue*, *sh:in* and *sh:class*.

Finally, the code related to the features which have been implemented is shared in the respective repository's folder[28]. The implementation of the rest of the SHACL constraints is in progress as well as the integration of equivalent compound shapes where the current IRM graph has already more than one shape in the respective group.

[25] https://github.com/marianojf89/Thesis/tree/main/Pre-validation%20configuration (last accessed on 2/12/2025)
[26] https://www.python.org/ (last accessed on 2/11/2025)
[27] https://rdflib.readthedocs.io/en/stable/# (last accessed on 2/11/2025)
[28] https://github.com/marianojf89/Thesis/tree/main/Integration%20procedure (last accessed on 2/12/2025)

6.4 Experiments and evaluation

Experiments were carried out on an Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz with 20 GB RAM. Furthermore, the execution of the Python programs was done using Python 3.13.1 and Virtual Studio Code IDE version 1.96.2. In addition, the library rdflib was used in its version 7.1.1.

6.4.1 Integration of two SHACL shape graphs corresponding to two different resorts

As already described, ESF manages different resorts all over France. Furthermore, each of these resorts has its own systems and data. For this specific experiment, two shape graphs which respectively represent two different resorts (Annecy and Chamonix) are desired to be integrated into one. The knowledge represented in each of these shape graphs is representative of the real data but not exact. Moreover, the shape graph related to Chamonix has been adapted to respect some of the restrictions recommended for the resort in case that there is a storm. Having all that said, ESF desires to have an IRM which represents a combination of the restrictions that were found in both shape graphs. In order to accomplish this, the Integration Procedure is executed to create a new IRM and to feed it with the combination of the aforementioned shape graphs. Moreover, both shape graphs and everything related to these experiments are deposited in the respective repository's folder[29]. The Annecy shape graph has a size of 7KB, contains 18 node shapes which aim to 18 different classes and a total of 39 property shapes which refers to the same amount of different properties. With respect to the Chamonix shape graph, it has a size of 8 KB, which comprehends 19 node shapes (targeting to 19 classes) and 50 property shapes each of them referring to a specific path value. Both shape graphs were reduced to using only those constraints which are already included in the current implementation of the integration procedure.

First experiment order

The first experiment comprehends first the integration of the Annecy shape graph with an empty IRM. The execution of this integration took 0.0672 seconds, and generated as an outcome the conversion of the Annecy shapes to the IRM structure. As a result, the IRM presented a size of 28 KB and generated 41 node shapes from which 2 are simple node shapes and 39 are compound shapes corresponding to the 39 properties described in the input. In addition, the IRM respects the IRM structure, including the definition of a group for each compound shape (39 in total) and a deactivated property with value True. Besides, the syntax has been checked to be correct through the use of an online syntax checker[30]. Furthermore, the SHACL consistency of the IRM has been checked by using the Python library pyshacl[31] which confirmed that the structure conforms to the SHACL standards. Next, the Chamonix SHACL shape is integrated with the current version of the IRM, which already has the Annecy restrictions. The execution of the integration took 0.2469 seconds and generated as an outcome a new version of the IRM which has a size of 66 KB. This updated IRM presents a total of 94 shapes from which only 2 are simple node shapes and the other 92 are compound shapes which have a property nested. Moreover, 53 groups of shapes are found, which means that this is the amount of different targets that

[29] github.com/marianojf89/Thesis/tree/main/Integration%20procedure/Experiments/Integration%20of%20two%20resorts (last accessed on 2/12/2025)

[30] http://ttl.summerofcode.be/ (last accessed on 2/11/2025)

[31] https://github.com/RDFLib/pySHACL (last accessed on 2/11/2025)



Figure 6.3: A graphical representation of the IRM generated as the integration of the Annecy SHACL shape graph with an empty IRM.

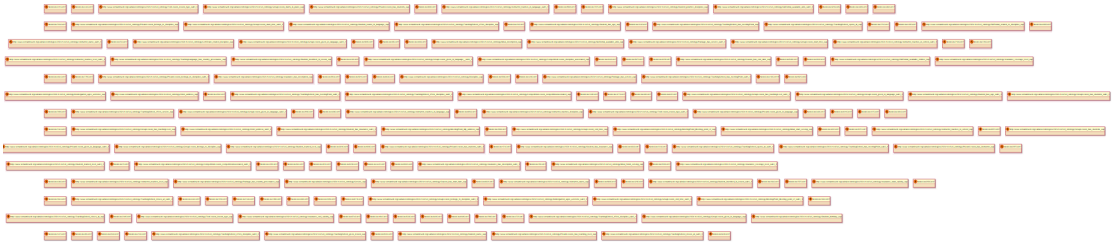


Figure 6.4: A graphical representation of the updated IRM generated as the integration of the Chamonix SHACL shape graph with a previous version of the IRM which already contained the Annecy shape graph.

the shape graph comprehends. Besides, 31 of these groups are formed by a hierarchy which has more than one shape. The correctness of the integrations has been checked manually and corresponds to the expected behavior. The syntax of the integrated SHACL shape graph has been tested with the aforementioned online checker which stated that the syntax is correct. In addition, the consistency of the integrated graph has been checked with *pyshacl*, and it was confirmed that the generated graph conforms to the SHACL standards. This experiment showed that the integration procedure is quite fast, generates the IRM structure in a correct way, creates hierarchies of shapes as expected and generates a SHACL shape graph which is consistent and respects the SHACL standards. Moreover, in this experiment, the integration generates a SHACL shape graph which offers more than one option for 31 targets. A graphic representation of the first generated version of the IRM, which includes only the shapes from the Annecy graph, and a second graphic representation which includes the last integration of the Chamonix graph shapes with the former graph, have been generated by an online tool[32]. Both representations are shown in the figures 6.3 and 6.4, respectively. Taking into account the size of these graphs, the aforementioned figures are shared in the thesis for representing the growth of the amount of shapes created when comparing the first outcome against the second one. If it's desired to look at these representations with more detail, both files are shared in the integration procedure experiments folder of the repository[33].

Second experiment: changing the processing order of the sources

Having into account that the behavior of the integration procedure proposed in this thesis is influenced by the order in which the shape graphs are processed, it's interesting to analyze the integration of the aforementioned SHACL shape graphs in the other possible order, meaning

[32] https://rdfshape.weso.es/ (last accessed on 2/11/2025)

[33] github.com/marianojf89/Thesis/tree/main/Integration%20procedure/Experiments/Integration%20of%20two%20resorts - /Annecy%20first (last accessed on 2/12/2025)

the Chamonix restrictions first, and then the Annecy ones. The integration of the Chamonix shape graph took 0.1316 seconds, which shows that the size of the graph affects slightly the execution time of the procedure. The generated IRM is composed by 52 shapes, from which only two are simple node shapes, meanwhile the rest has a nested property. In addition, 50 groups were defined, one per compound shape. Going forward, the integration of the Annecy shape graph was done successfully in a total time of 0.1667 seconds. The final version of the IRM in this experiment represents 92 shapes from which 2 are simple node shapes and 90 are compound node shapes. Moreover, 53 different groups were defined, from which 30 present a hierarchical structure with at least two shapes. The influence of the order in the integration of the shape graphs is shown in the amount of generated shapes, having two compound shapes less than in the previous experiment. However, it's shown that the amount of groups stays the same, which means that all the simple and compound shape targets are represented. As done in the previous experiment, the syntax and the consistency of the generated IRMs have been checked successfully.

6.4.2 Activation of the generated ESF IRM related to the weather context

This experiment aims to cover the second ESF use case described at the beginning of this chapter, in which it's needed that the IRM is set with the desired restrictions, depending on a current external context.

One of the specific weather conditions that change the normal behavior of the resorts and could have place often, is when there is a storm or there are high probabilities that there will be one. In these weather conditions, it's recommended to respect the following conditions (between others): the amount of students per lesson is reduced, the disciplines available should be the safest ones such as Nordic, BackCountry and SnowShoeing, an insurance other than the basic one is mandatory for every student that desires to do an activity, the types of lessons available are also reduced to the private and group common lessons, the equipment available for rental should be restricted to those which are related to the available disciplines, the instructors should be as experts as possible, the students shouldn't be beginners and they should be adults, and finally, the period of time and the part of the day in which the resorts are open should be smaller than usual and restricted to the morning and early afternoon if possible. However, this experiment is limited to the implementation completed so far, which could include those restrictions that could be represented with cardinality properties. This includes the amount of students per lesson, and the mandatory insurance.

First, these weather characteristics for the stormy days are represented in the ESF ontology. It's desired that in these conditions, the private lessons are reduced to two students and the group lessons to six students. The figures 6.5 and 6.6 show the definition of the characteristics for the group lessons and their relationship with the stormy weather context respectively.

Then, based on this ontology and the IRM generated in the previous experiment (first order), the pre-validation configuration should asses the ESF experts in the activation of the sub-IRM graph which best fits to a mix of the current IRM capabilities, the related recommendations and the experts preferences. The version of the IRM which is going to be configured presents groups of shapes with two different options for each of the targets which are going to be mainly evaluated, which are the following: (groupLesson, has_students), (privateLesson, has_students) and (student, has_insurance). As an example, the available shapes for targeting the first combination are shown in the listing 6.3.

The execution of the pre-validation configuration procedure took 4.3057 seconds and included the acquisition of the current active context from the RDF context graph, retrieving from the



Figure 6.5: The representation of the desired restrictions for the group lessons when there is a storm.

ESF context ontology the characteristics related to the active context, filtering the IRM shapes taking into account those characteristics, activating those shapes which accomplished to the characteristics, retrieving the active shapes from those groups that present more than one candidate shape, activating those shapes which are unique in their respective groups, activating the simple node shapes, and retrieving those shapes which weren't related to the targets influenced by the context (groups that weren't filtered). The behavior of the procedure is as expected and resulted in the following actions: the 22 compound shapes which were unique in their groups have been activated. Furthermore, the two simple node shapes have also changed their status to active.

With respect to the groups which were filtered by the contextual characteristics, the following actions have been taken by the procedure: first, for the (groupLesson, has_students), there's only one shape in IRM which has a maximum cardinality lower or equal than 6 and in consequence, only this shape has been activated and the group is all set. Second, the group (privateLesson, has_students) has only one shape that conforms to the highest cardinality recommended by the experts (2) and this shape was correctly activated. Lastly, the group (student, has_insurance) presents two shapes which accomplish with the restriction defined in the context, which expected a minimum cardinality of one, ensuring that each student has an insurance. An image of these candidate shapes is shown in the listing 6.4. Moreover, in the complete workflow of the pre-validation configuration process, these two shapes are tested against the RDF KG and a report would be presented to the user asking him to decide which of them should be activated. The implementation of this part of the process is still in progress, so the generation of the report is skipped in this experiment. Finally, the same process of report generation would be done for the rest of the 28 groups which haven't been filtered by the contextual characteristics. In this experiment, the list of the respective IRIs has been obtained successfully. In order to finish the experiment, the decision of the user is simulated as if the desired would be to activate the super shapes of those groups with candidate shapes. The complete execution of the pre-validation configuration took in total 5.0740 seconds and generated an IRM activated which respects the syntax and standards of SHACL and which is ready for its use for the validation of the RDF KG. The files processed and generated by this experiment are available in the thesis repository[34].

This experiment has shown the usefulness of the IRM and the flexibility that it offers for being adapted to different desired behaviors. However, it would be very valuable if more decisions for the activations of shapes could be taken automatically by the procedure. There are some interesting ideas for improving this, but they will be detailed in the further work since it's out of the scope of this thesis.

[34]https://github.com/marianojf89/Thesis/tree/main/Pre-validation%20configuration/Experiments/Storm%20weather (last accessed on 2/12/2025)



Figure 6.6: Link of a class and its associated characteristics with a specific context.

Listing 6.3: The IRM's available shapes for restricting the students of the group lessons.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:GroupLesson_has_students_sup
    a sh:NodeShape ;
    sh:deactivated true ;
    sh:targetClass ex:GroupLesson ;
    sh:property [
        sh:path ex:has_students
        sh:maxCount 10 ;
        sh:minCount 3 ;
        sh:nodeKind sh:IRI ;
        sh:group ex:GroupLesson_has_students ; ] .

ex:GroupLesson_has_students_sub1_1
    a sh:NodeShape ;
    sh:deactivated true ;
    sh:node ex:GroupLesson_has_students_sup ;
    sh:targetClass ex:GroupLesson ;
    sh:property [
        sh:path ex:has_students ;
        sh:maxCount 6 ;
        sh:minCount 5 ;
        sh:nodeKind sh:IRI ;
        sh:group ex:GroupLesson_has_students ; ] .

ex:GroupLesson_has_students a sh:propertyGroup .
```

Listing 6.4: The IRM's candidate shapes for (student - has_insurance) which accomplishes with the requirements defined in the context of a storm.

```
@prefix sh:   <http://www.w3.org/ns/shacl#> .
@prefix ex: <http://localhost/ontology/esf_ontology/> .

ex:Student_has_insurance_sup
    a sh:NodeShape ;
    sh:deactivated false ;
    sh:targetClass ex:Student ;
    sh:property [
        sh:path ex:has_insurance ;
        sh:in ( "ex:high" "ex:low" "ex:mid" ) ;
        sh:minCount 1 ;
        sh:group ex:Student_has_insurance ; ] .

ex:Student_has_insurance_sub1_1
    a sh:NodeShape ;
    sh:deactivated false ;
    sh:node ex:Student_has_insurance_sup ;
    sh:targetClass ex:Student ;
    sh:property [
        sh:path ex:has_insurance ;
        sh:class ex:Insurance ;
        sh:in ( "ex:high" "ex:mid" ) ;
        sh:maxCount 2 ;
        sh:minCount 1 ;
        sh:group ex:Student_has_insurance ; ] .
```

```
26
27 ex:Student_has_insurance a sh:PropertyGroup .
```

6.5 Conclusion

In this section, the ESF ski teaching school use case has been described in a general way and then, two specific related use cases have been also introduced for presenting a scenario useful for the evaluation of the thesis' contributions. Furthermore, two knowledge representation models created for representing the use case domain have been mentioned. Moreover, a detail of the implementations done so far has been shown. Next, two experiments were done for the evaluation of the integration procedure, which includes the integration of two SHACL shape graphs related to two ski resorts. The experiments have shown that the procedure has a good performance and that generates an IRM structure which respects the SHACL syntax, is consistent and respects the SHACL standards. Besides, another experiment has been described which aimed to the pre-validation configuration of one of the IRMs generated in the previous experiment. In this case, it was possible to activate a sub-graph of the IRM which was partly decided by the user, and partly derived automatically from the characteristics that a storm context presents on a ski resort. This use case shows on one side that the IRM is useful for behaving in different ways depending on the current context characteristics and on the desires of the user. Moreover, on the other hand, the experiment has shown that it's necessary or highly recommended to provide the procedure with the capability to take more decisions automatically and to reduce the amount of decisions that are asked to the user.

7

Conclusions and future work

Contents

7.1 Summary 109
7.2 Perspectives and future directions 110

7.1 Summary

In this thesis, the introduction of SHACL shape graphs in semantic environments which present heterogeneous sources and different context requirements has been addressed. This research presents a proposal for the enrichment of Semantic Web based systems, providing benefits for several domains which exploits the knowledge representation models, such as the Industry 4.0.

Firstly, in chapter 3, a systematic literature review (SLR) has been carried out in order to provide a description of the domain related to the validation of semantic constraints in knowledge representation models. As an outcome of this study, the definition of a related taxonomy which includes 18 categories of different types of related works, has been presented. In addition, the articles that were selected in the review have been classified in these categories and the associated current open challenges have been enumerated. Moreover, research gaps which involve the methodologies for the creation of SHACL shape graphs were highlighted among the other open challenges found. After going deeper in the latter research gap, the need for a SHACL shape graph interpretation which is able to attend different contexts, and the need for the integration of multiple SHACL shape graphs have been described and conform the other two research gaps that have been addressed in this thesis (in addition to the aforementioned taxonomy).

Secondly, in chapter 4, the Integrated Restrictions Model (IRM) has been proposed to help the SHACL shape graphs to validate multiple contexts. Based on the SHACL standards, this interpretation of the model is capable of representing groups of shapes which aim to the same target with different restrictions that could hold inter-shape conflicts. Furthermore, these groups of shapes are organized hierarchically in a structure in which a shape of the group is contained in its super shape. Additionally, this chapter presented an IRM's pre-validation configuration process which helps to activate the IRM sub-graph that best fits the characteristics of the current context and the user's desires. Moreover, a possible representation of the context knowledge is thought, which includes an ontology and an RDF graph. Besides, as part of this process, a set of SPARQL queries have been proposed for the acquisition of the context characteristics and for the manipulation of the IRM, respecting the characteristics retrieved. Finally, this chapter has presented the Provenance Ontology structure which is designed to represent the knowledge

related to each specific integration that generated as an outcome a specific IRM shape. Through the population of this ontology, it's possible to link the IRM specific shapes with their respective source provenance knowledge.

Thirdly, in chapter 5, a procedure for the integration of multiple SHACL shape graphs which are part of a common environment has been proposed. This procedure takes as inputs a SHACL shape graph and an IRM (which has been presented in chapter 4), and generates an updated IRM as the outcome, which could include new knowledge related to the integration of the previous IRM version with the new input processed. As part of the integration procedure, a process for the equivalence detection of those shapes which aim to the same target ti has been proposed. Furthermore, for the integration of the equivalent shapes, a conflict resolution and integration strategy has been presented. This strategy defines a set of scenarios, which are related to certain parameters that could be found when comparing the constraints that are aimed to be integrated. In addition, each scenario has a set of related actions, which resolves the constraints conflicts by the specialization of shapes, generating step by step, an IRM structure as an outcome. Moreover, the parameters that are taken into account for the definition of the scenarios are the type of each SHACL constraint, the specific values that they have assigned and the containment relationship that holds between them. In addition, the scenarios' actions are also influenced by the temporal structure of shapes that the current integration is generating.

Fourthly, in chapter 6, two practical use cases related to the French Ski School have been described in order to show the benefits that the contributions presented in this thesis offer. The first one detailed how the company processes knowledge from different resorts, which respects an underlying formal context but could have differences related to the specific characteristics of each of them. Having different shape graphs which represent the restrictions of individual resorts, this use case is useful for testing the integration procedure presented in this thesis. Through the generation of an IRM, ESF could obtain a semantic restriction model which aims to the whole ESF universe of targets and that presents options for some of them. The second use case outlined the existence of different external contexts, such as the weather, which could influence the behavior of a ski resort. Taking into account the IRM generated during the experiments applied to the first use case, and the recommended restrictions for a specific weather context in which there is a stormy day, the pre-validation procedure could be evaluated to set the IRM sub-graph that best fits the specific context and the user's preferences. In addition, the implementation details of the contributions have been detailed, including the created knowledge representation models, the pre-validation configuration procedure (with the involved models), and the SHACL shape graphs integration process. Finally, based on the described implementations and on the presented use cases, a number of experiments have been done in order to evaluate the contributions presented. The integration procedure has shown to perform quite well by integrating the shape graphs in less than a second. In addition, the generated IRM accomplished with the SHACL standards and approved consistency, syntax and correctness checks. Regarding the pre-validation procedure, the activation has shown to be correct and confirmed the usefulness of the IRM design. The activated model passed as well the consistency checks. The performance of this procedure could be improved, since it took 5 seconds to complete the tasks of the workflow that were possible to cover with the current implementation.

7.2 Perspectives and future directions

Several directions for further development came out of each of the contributions of this thesis. Regarding the taxonomy defined in chapter 3, as one of the outcomes of the SLR, a number

of research gaps related to the validation of semantic constraints on knowledge representation models have been described in the section 3.6.2.

With respect to the Integrated Restrictions Model presented in chapter 4, apart from its use for RDF validation, more exploitations could be researched. More specifically, the IRM model could be involved in processes related to schema description, query support, RDF sub-graph extraction, user interface construction and RDF graph evolution guidance.

Going forward, the pre-validation configuration workflow presented in 4.2.2 gives space for several improvement proposals. Firstly, adding more constraint types to the implementation, the development of the report generation related to the candidate shapes and its presentation to the user are necessary for having an implementation of the complete workflow. In addition, as mentioned in the evaluation chapter, the performance of the current implementation should be improved, for example through the reduction of the amount of queries that are executed. Secondly, alternatives for the activation of the IRM could be researched based in other parameters such as the level of shape hardness desired or even through a certain previous analysis of the RDF data that is aimed to be validated. In addition, it would be interesting to have the possibility of representing the IRM graphically in a nice and intuitive visual interface. This would give the user an easier understanding of the general restriction model and an interactive way for selecting the desired shapes by just clicking between options, or even on individual shapes. Moreover, an automatic activation procedure which doesn't need the user's intervention would be very interesting for the model's acceptance. The procedure could be fed with the capability for taking decisions over the candidate shapes, taking into account knowledge such as previous executions of the specific context, or the containment relationships of the candidate shapes. As an example, if the most restrictive level of shapes doesn't generate a validation report, it wouldn't be necessary to check the shapes of the higher levels. Thirdly, alternatives for the representation of the context characteristics could be researched, even in the OWL constraints used (other than equivalent_to) or directly in the model used in its interpretation. In this sense, it would be interesting to simplify as much as possible this task to the domain experts. Fourthly, the enrichment of the report presented to the user could be further developed. An interesting option would be to translate each shape to a SPARQL query, and instead of generating a validation report (which only informs the errors), the execution of the query and some slight modifications of it would give more flexibility for the acquisition of extra valuable information. For example, the complement (meaning those shapes which passed the validation) and inter-hierarchy relationships could be analyzed. Finally, another interesting feature that could be researched is saving the knowledge once a specific sub-graph of the IRM is activated for a specific context. Then, this knowledge could be used as part of the workflow.

Next, the integration procedure presented in chapter 5 presents also space for further development. Firstly, the completion of the implementation is desired, including the missing SHACL constraints and the integration procedure of a shape graph against a version of the IRM which already has hierarchies with more than one shape. Secondly, the current equivalence detection procedure is quite limited, since it defines the shapes equivalences based on the equality of their focus nodes' (in this case, classes) and path values' IRIs associated. However, there exists the possibility that those IRIs are referring to different things in each of the sources, mostly if the integration is done between shape graphs that are extracted from independent systems. In order to improve this and avoid possible confusions, it would be quite interesting to develop a new equivalence detection procedure which analyzes some characteristics of the class in each of the source graphs. Moreover, if a source shape graph (SSG) is extracted from another source such as an ontology or an RDF KG, the procedure could involve the analysis of the neighborhood by looking into the inbound/outbound relationships and the values of the related nodes

that are present in these original structures. Thirdly, the inclusion of a priority classification, based on the type of SHACL constraint that is being compared, could be researched for the improvement of the conflict resolution strategy. Fourthly, the minimization of the anchorage of the integrated shapes with specific structures of the data could be investigated, generating as a result, an IRM which could be reused without depending on the presence of specific classes in the data. This could be done by the definition of the classes in shape-based constraints, meaning that a node shape would have the sh:targetClass definition in another shape, which is referenced with sh:node. Moreover, during the pre-validation configuration, if the referenced shape is deactivated, then the node shape won't be attached to a specific class. In addition, this wouldn't affect the equivalence detection process, since currently it takes into account the class definitions in referenced shapes and, as an alternative, the IRM sh:group definition could be also used for the equivalence detection. Finally, other integration strategies could be developed and optionally offered to the users depending on their objectives. An example would be the creation of an alternative integration based on the 'loose constraining' strategy which always keeps the softer constraint and doesn't involve the design of hierarchies.

In addition, further exploitations of the Source Provenance ontology presented in 4.2.3 could be researched. This representation could be involved in a pre-validation configuration of the IRM, in which the user desires to activate a sub-graph of shapes which must be related to a specific source. In addition, the ontology could be used to rebuild original shapes which could then be used for different purposes such as the enrichment of the report which is presented to the user.

Finally, another interesting feature that could be studied is the inclusion of the integration technique in a bigger Semantic Web architecture, which involves as previous steps, the automatic extraction of SHACL shape graphs from heterogeneous sources such as RDBs, ontologies and RDF KGs.

Résumé étendu en français

Introduction

Le World Wide Web actuel peut être considéré comme un énorme réseau de documents et de programmes connectés, où la plupart du contenu représentatif est du texte brut qui ne peut être interprété que par les ordinateurs afin de l'afficher dans un format agréable pour les humains qui le consommeront. Ainsi, une quantité importante d'informations précieuses ne peut être traitée par les machines. Afin d'amorcer un changement dans ce sens, un sous-domaine de *Artificial Intelligence*, le *Knowledge Representation*, a fourni les fondations nécessaires. Sur la base de ce domaine, le *Semantic Web* a été proposé par Tim Berners Lee en 2001 ([18]) comme une nouvelle version de l'internet, dans laquelle les machines pourraient *« comprendre »* ce qu'elles stockent et représentent. Cette compréhension est possible si le web est doté d'un contenu structuré et de règles d'inférence qui permettent de raisonner automatiquement sur le contenu représenté. Si cette vision était révolutionnaire à l'époque, le web sémantique a depuis évolué de manière significative, sous l'impulsion des progrès réalisés dans le domaine des données liées, de l'intelligence artificielle et des technologies d'intégration des données. Des avancées récentes, telles que l'adoption de langages sous-jacents dans la construction des graphes du web sémantique et l'intégration de langages de contraintes (CL) pour la validation des contraintes sémantiques, démontrent la pertinence continue des technologies du web sémantique ([54]). En outre, la principale technologie liée au web sémantique est le *Resource Description Framework (RDF)*, qui représente les concepts du monde réel sous la forme de *triples (s,p,o)*. En outre, les graphes du web sémantique qui représentent les triples RDF sont également connus et interprétés de nos jours comme des *Knowledge Graphs* (KGs). Ces représentations sont définies comme des graphes formels, étiquetés et dirigés, où les nœuds représentent des concepts du monde réel, tandis que les arêtes représentent les relations existantes entre les concepts. Normalement, ces modèles représentent les connaissances au niveau de l'instance, avec une capacité d'expression plus faible (par rapport aux ontologies) et sont souvent alimentés par des sources hétérogènes ([86]). Une autre famille de technologies, comme le *Web Ontology Language (OWL)*, est utilisée pour la création de *ontologies*, qui sont des représentations conceptuelles, formelles et partagées d'une partie du monde, dotées d'une riche expressivité qui permet de représenter un grand nombre de caractéristiques ([111]). En outre, les KG et les ontologies sont normalement conçues pour être interprétées avec les hypothèses *Open World Assumption (OWA)* et *Non-Unique Name Assumption (NUNA)*. La première stipule qu'un fait qui ne figure pas dans les définitions explicites ne peut être défini comme faux, mais plutôt comme inconnu. En outre, la seconde stipule qu'il est possible d'avoir deux entités avec des noms différents qui se réfèrent au même concept du monde réel. Ces hypothèses définissent un cadre pour l'inférence de nouvelles connaissances basées sur les faits explicites ([164]).

En outre, il est courant que les systèmes concernés présentent une architecture web sémantique, qui implique l'intégration et le traitement de données provenant de sources hétérogènes

dans un KG RDF. Dans ces scénarios, les données peuvent se présenter sous différents formats tels que le langage de balisage extensible (XML), la notation d'objets JavaScript (JSON), les valeurs séparées par des virgules (CSV), les bases de données relationnelles (RDB) ou les triples RDF. En outre, ces systèmes pourraient également impliquer différents modèles tels que des ontologies, des graphes de forme ou même des schémas XML. Toutefois, les modèles présentés et les systèmes mentionnés présentent encore certains inconvénients, tels que la complexité liée à l'intégration de sources hétérogènes, les difficultés liées à la représentation et à la conservation efficaces des connaissances, les problèmes d'évolutivité lors de la représentation de données volumineuses, le traitement des connaissances incomplètes et la validation des données d'instance représentées ([85]).

Afin d'accroître l'adoption de ces modèles, il est essentiel de poursuivre la recherche dans ce domaine et de parvenir à des propositions de solutions. En particulier, en ce qui concerne la validation des modèles, il existe de nombreux domaines dans lesquels la cohérence des modèles revêt une importance vitale en raison de la sensibilité des résultats, ou même du désir de compter sur un modèle aussi fiable, efficace et efficient que possible. Ces exigences ont mis en évidence la nécessité d'avoir la possibilité d'ajouter de la cohérence aux modèles par la création d'un processus qui pourrait valider les modèles de manière fermée, comme c'est le cas dans les BDR. En conséquence, la recherche connexe s'est concentrée sur le développement de modèles de représentation des connaissances qui adhèrent au *Closed World Assumption* (CWA), où les faits qui ne sont pas explicitement définis sont considérés comme faux, et au *Unique Name Assumption* (UNA), qui garantit que chaque concept du monde réel est désigné par un nom unique. Ces modèles ont été proposés pour résoudre les problèmes de restriction et de validation des modèles du Web sémantique, qui sont généralement fondés sur l'OWA et l'UNA ([154], [108]).

Au cours des dernières années, certaines approches ont été proposées afin de pouvoir valider les modèles de représentation des connaissances de manière fermée. Plus précisément, l'accent a été mis sur la validation des graphes RDF. Ces approches ont évolué, passant de méthodes centrées sur la génération de requêtes *SPARQL*[35] *(qui signifie « SPARQL Protocol and RDF Query Language »)* (le langage de requête standard pour RDF) ou l'augmentation des modèles de langage existants, tels que OWL, avec de nouvelles caractéristiques connexes, au développement de langages entièrement nouveaux capables de représenter les contraintes d'un modèle. Ce dernier type d'approche a donné lieu à la création de quelques CL, tels que *Shape Expressions (ShEX), SPARQL Inferencing Notation (SPIN) et Shapes Constraint Language (SHACL)* ([98]), ce dernier étant défini comme la norme par le World Wide Web Consortium (W3C[36]). Les modèles SHACL sont également connus sous le nom de *« SHACL shape graphs »*, car ce langage respecte une structure de « formes » qui représentent des groupes de contraintes exprimées sous la forme d'un graphe RDF qui cible et restreint un ou plusieurs nœuds d'un graphe de données, qui est lui-même un autre graphe RDF représentant la connaissance de l'instance.

Portée et objectifs de la thèse

Cette thèse est une contribution à l'École du Ski Français (ESF), qui est une société française dédiée à la gestion des stations de ski en France. L'objectif de ce projet était de construire un environnement Web sémantique qui représente le domaine spécifique de l'entreprise avec des modèles de représentation des connaissances qui impliquent les niveaux de représentation d'instance, de concept et de restriction, sur la base de sources hétérogènes existantes. En outre,

[35] https://www.w3.org/TR/sparql11-query/ (last accessed on 2/11/2025)
[36] https://www.w3.org/ (last accessed on 2/11/2025)

il y a un intérêt particulier dans la validation des instances de connaissance liées, ayant un scénario dans lequel il existe plus d'un graphe de forme et où les restrictions définies nécessitent une flexibilité pour être adaptées à différents contextes possibles.

En termes généraux, l'objectif de cette thèse est de développer des approches pour construire un graphe de forme unifié dans un environnement sémantique caractérisé par des sources hétérogènes. De plus, ces sources conduisent souvent à la génération de plusieurs graphes de forme, et le scénario peut impliquer diverses exigences contextuelles. Dans ce contexte, les objectifs de la thèse sont les suivants :

1. Analyser et décrire les travaux liés à la validation qui sont directement ou indirectement impliqués dans la résolution du problème de cohérence dans les modèles de représentation des connaissances.

2. Sur la base des normes SHACL, fournir une conception de graphe de formes qui puisse être adaptée à différents contextes.

3. Proposer une méthode d'intégration des graphes de formes SHACL qui génère un graphe de formes SHACL unifié respectant la conception présentée dans l'objectif précédent.

4. Montrer son utilité potentielle dans un scénario pratique.

Résultats

Tout d'abord, dans le chapitre 3, une revue systématique de la littérature (SLR) a été effectuée afin de fournir une description du domaine lié à la validation des contraintes sémantiques dans les modèles de représentation des connaissances. Cette étude a débouché sur la définition d'une taxonomie connexe qui comprend 18 catégories de différents types de travaux connexes. En outre, les articles sélectionnés dans l'étude ont été classés dans ces catégories et les défis actuels associés ont été énumérés. En outre, les lacunes de la recherche qui impliquent les méthodologies pour la création de graphes de forme SHACL ont été mises en évidence parmi les autres défis ouverts trouvés. Après avoir approfondi cette dernière lacune de recherche, le besoin d'une interprétation des graphes de forme SHACL capable de s'adapter à différents contextes et le besoin d'intégration de multiples graphes de forme SHACL ont été décrits et constituent les deux autres lacunes de recherche qui ont été abordées dans cette thèse (en plus de la taxonomie susmentionnée).

Deuxièmement, au chapitre 4, le modèle de restrictions intégré (IRM) a été proposé pour aider les graphiques de formes SHACL à valider des contextes multiples. Basée sur les normes SHACL, cette interprétation du modèle est capable de représenter des groupes de formes qui visent la même cible avec des restrictions différentes qui pourraient entraîner des conflits entre les formes. En outre, ces groupes de formes sont organisés hiérarchiquement dans une structure dans laquelle une forme du groupe est contenue dans sa super-forme. En outre, ce chapitre a présenté le processus de configuration de prévalidation d'un MRI qui permet d'activer le sous-graphe du MRI qui correspond le mieux aux caractéristiques du contexte actuel et aux souhaits de l'utilisateur. En outre, une représentation possible de la connaissance du contexte est envisagée, qui comprend une ontologie et un graphe RDF. De plus, dans le cadre de ce processus, un ensemble de requêtes SPARQL a été proposé pour l'acquisition des caractéristiques du contexte et pour la manipulation du MRI, en respectant les caractéristiques récupérées. Enfin, ce chapitre a présenté la structure de l'ontologie de provenance, conçue pour représenter les connaissances relatives à chaque intégration spécifique ayant abouti à une forme IRM particulière. Grâce à la

population de cette ontologie, il est possible de lier les formes spécifiques de l'IRM avec leurs connaissances de provenance respectives.

Troisièmement, au chapitre 5, une procédure d'intégration de plusieurs graphes de forme SHACL faisant partie d'un environnement commun a été proposée. Cette procédure prend en entrée un graphique de formes SHACL et un MRI (qui a été présenté au chapitre 4), et génère un MRI mis à jour comme résultat, qui pourrait inclure de nouvelles connaissances liées à l'intégration de la version précédente du MRI avec la nouvelle entrée traitée. Dans le cadre de la procédure d'intégration, un processus de détection de l'équivalence des formes qui visent la même cible a été proposé. En outre, pour l'intégration des formes équivalentes, une stratégie de résolution des conflits et d'intégration a été présentée. Cette stratégie définit un ensemble de scénarios, qui sont liés à certains paramètres susceptibles d'être trouvés lors de la comparaison des contraintes à intégrer. En outre, chaque scénario comporte un ensemble d'actions connexes qui résolvent les conflits de contraintes par la spécialisation des formes, générant ainsi, étape par étape, une structure IRM comme résultat. En outre, les paramètres pris en compte pour la définition des scénarios sont le type de chaque contrainte SHACL, les valeurs spécifiques qui leur sont attribuées et la relation de confinement qui existe entre elles. En outre, les actions des scénarios sont également influencées par la structure temporelle des formes générées par l'intégration actuelle. Quatrièmement, dans le chapitre 6, deux cas d'utilisation pratiques liés à l'École de Ski Française ont été décrits afin de montrer les avantages offerts par les contributions présentées dans cette thèse. Le premier cas détaille comment l'entreprise traite les connaissances provenant de différentes stations de ski, respectant un contexte formel sous-jacent tout en présentant des différences liées aux caractéristiques spécifiques de chacune d'elles. Disposant de différents graphes de formes représentant les restrictions des stations individuelles, ce cas d'utilisation est utile pour tester la procédure d'intégration présentée dans cette thèse. Grâce à la génération d'un IRM, l'ESF a pu obtenir un modèle de restriction sémantique visant l'ensemble de l'univers des cibles de l'ESF et offrant des options pour certaines d'entre elles. Le second cas d'utilisation met en évidence l'existence de différents contextes externes, tels que la météo, qui peuvent influencer le fonctionnement d'une station de ski. En prenant en compte l'IRM généré lors des expériences appliquées au premier cas d'utilisation, ainsi que les restrictions recommandées pour un contexte météorologique spécifique, comme une journée de tempête, la procédure de pré-validation pourrait être évaluée afin de sélectionner le sous-graphe IRM le mieux adapté au contexte particulier et aux préférences de l'utilisateur. De plus, les détails de l'implémentation des contributions ont été décrits, y compris les modèles de représentation des connaissances créés, la procédure de configuration de pré-validation (avec les modèles impliqués) et le processus d'intégration des graphes de formes SHACL. Enfin, sur la base des implémentations décrites et des cas d'utilisation présentés, un certain nombre d'expériences ont été menées afin d'évaluer les contributions proposées. La procédure d'intégration a démontré de bonnes performances en intégrant les graphes de formes en moins d'une seconde. De plus, l'IRM généré était conforme aux standards SHACL et a passé avec succès les vérifications de cohérence, de syntaxe et de correction. En ce qui concerne la procédure de pré-validation, l'activation s'est révélée correcte et a confirmé l'utilité de la conception de l'IRM. Le modèle activé a également passé les tests de cohérence. Les performances de cette procédure pourraient être améliorées, car l'exécution des tâches du workflow couvertes par l'implémentation actuelle a pris 5 secondes.

Perspectives et directions futures

Chacune des contributions de cette thèse a donné lieu à plusieurs orientations pour un développement ultérieur. En ce qui concerne la taxonomie définie au chapitre 3, comme l'un des résultats du SLR, un certain nombre de lacunes de recherche liées à la validation des contraintes sémantiques sur les modèles de représentation des connaissances ont été décrites dans la section 3.6.2.

En ce qui concerne le modèle intégré de restrictions présenté au chapitre 4, outre son utilisation pour la validation RDF, d'autres exploitations pourraient être étudiées. Plus précisément, le modèle IRM pourrait être impliqué dans les processus liés à la description des schémas, au soutien des requêtes, à l'extraction des sous-graphes RDF, à la construction de l'interface utilisateur et à l'orientation de l'évolution des graphes RDF.

Pour aller de l'avant, le processus de configuration de la prévalidation présenté dans 4.2.2 offre un espace pour plusieurs propositions d'amélioration. Tout d'abord, l'ajout d'autres types de contraintes à la mise en œuvre, le développement de la génération de rapports relatifs aux formes candidates et leur présentation à l'utilisateur sont nécessaires pour disposer d'une mise en œuvre du flux de travail complet. En outre, comme mentionné dans le chapitre sur l'évaluation, les performances de l'implémentation actuelle devraient être améliorées, par exemple en réduisant le nombre de requêtes exécutées. Deuxièmement, des alternatives pour l'activation du MRI pourraient être recherchées sur la base d'autres paramètres tels que le niveau de dureté de la forme souhaité ou même par le biais d'une certaine analyse préalable des données RDF que l'on souhaite valider. En outre, il serait intéressant d'avoir la possibilité de représenter le MRI graphiquement dans une interface visuelle agréable et intuitive. L'utilisateur comprendrait ainsi plus facilement le modèle de restriction général et disposerait d'un moyen interactif de sélectionner les formes souhaitées en cliquant simplement sur les options, voire sur des formes individuelles. En outre, une procédure d'activation automatique ne nécessitant pas l'intervention de l'utilisateur serait très intéressante pour l'acceptation du modèle. La procédure pourrait être dotée de la capacité de prendre des décisions sur les formes candidates, en tenant compte de connaissances telles que les exécutions précédentes du contexte spécifique ou les relations de confinement des formes candidates. Par exemple, si le niveau de formes le plus restrictif ne génère pas de rapport de validation, il ne serait pas nécessaire de vérifier les formes des niveaux supérieurs. Troisièmement, des alternatives pour la représentation des caractéristiques du contexte pourraient être recherchées, même dans les contraintes OWL utilisées (autres que equivalent_to) ou directement dans le modèle utilisé pour son interprétation. En ce sens, il serait intéressant de simplifier au maximum cette tâche aux experts du domaine. Quatrièmement, l'enrichissement du rapport présenté à l'utilisateur pourrait être développé. Une option intéressante consisterait à traduire chaque forme en une requête SPARQL et, au lieu de générer un rapport de validation (qui n'informe que des erreurs), l'exécution de la requête et quelques légères modifications de celle-ci offriraient plus de souplesse pour l'acquisition d'informations supplémentaires précieuses. Par exemple, le complément (c'est-à-dire les formes qui ont passé la validation) et les relations inter-hiérarchiques pourraient être analysés. Enfin, une autre caractéristique intéressante qui pourrait être étudiée est la sauvegarde des connaissances une fois qu'un sous-graphe spécifique de l'IRM est activé dans un contexte spécifique. Ces connaissances pourraient ensuite être utilisées dans le cadre du flux de travail.

Ensuite, la procédure d'intégration présentée au chapitre 5 offre également des possibilités de développement. Tout d'abord, il est souhaitable de compléter la mise en œuvre, y compris les contraintes SHACL manquantes et la procédure d'intégration d'un graphe de formes par rapport à une version de la MRI qui comporte déjà des hiérarchies avec plus d'une forme. Deuxièmement, la procédure actuelle de détection des équivalences est assez limitée, puisqu'elle définit les

équivalences de formes sur la base de l'égalité des IRI associés à leurs nœuds de focalisation (dans ce cas, les classes) et à leurs valeurs de chemin. Cependant, il est possible que ces IRI fassent référence à des éléments différents dans chacune des sources, surtout si l'intégration est effectuée entre des graphes de formes extraits de systèmes indépendants. Afin d'améliorer cette situation et d'éviter les confusions possibles, il serait très intéressant de développer une nouvelle procédure de détection d'équivalence qui analyse certaines caractéristiques de la classe dans chacun des graphes sources. En outre, si un graphe de forme source (SSG) est extrait d'une autre source telle qu'une ontologie ou un RDF KG, la procédure pourrait impliquer l'analyse du voisinage en examinant les relations d'entrée/sortie et les valeurs des nœuds apparentés qui sont présents dans ces structures d'origine. Troisièmement, l'inclusion d'une classification prioritaire, basée sur le type de contrainte SHACL comparée, pourrait être étudiée pour améliorer la stratégie de résolution des conflits. Quatrièmement, la minimisation de l'ancrage des formes intégrées avec des structures spécifiques des données pourrait être étudiée, générant ainsi un MRI qui pourrait être réutilisé sans dépendre de la présence de classes spécifiques dans les données. Cela pourrait se faire par la définition des classes dans des contraintes basées sur la forme, ce qui signifie qu'une forme de nœud aurait la définition sh:targetClass dans une autre forme, qui est référencée avec sh:node. De plus, lors de la configuration de pré-validation, si la forme référencée est désactivée, la forme du nœud ne sera pas attachée à une classe spécifique. En outre, cela n'affecterait pas le processus de détection d'équivalence, puisqu'il prend actuellement en compte les définitions de classe dans les formes référencées et, comme alternative, la définition sh:group du MRI pourrait également être utilisée pour la détection d'équivalence. Enfin, d'autres stratégies d'intégration pourraient être développées et proposées aux utilisateurs en fonction de leurs objectifs. Un exemple serait la création d'une intégration alternative basée sur la stratégie « loose constraining » qui conserve toujours la contrainte la plus douce et n'implique pas la conception de hiérarchies.

En outre, d'autres exploitations de l'ontologie Source Provenance présentée dans 4.2.3 pourraient être étudiées. Cette représentation pourrait être utilisée dans une configuration de pré-validation de la GIR, dans laquelle l'utilisateur souhaite activer un sous-graphe de formes qui doivent être liées à une source spécifique. En outre, l'ontologie pourrait être utilisée pour reconstruire des formes originales qui pourraient ensuite être utilisées à différentes fins, telles que l'enrichissement du rapport présenté à l'utilisateur.

Enfin, une autre caractéristique intéressante qui pourrait être étudiée est l'inclusion de la technique d'intégration dans une architecture Web sémantique plus large, qui implique, comme les étapes précédentes, l'extraction automatique des graphes de forme SHACL à partir de sources hétérogènes telles que les BDR, les ontologies et les KG RDF.

Resumen extendido en español

Introducción

La web mundial actual puede considerarse como una enorme red de documentos y programas conectados, donde la mayor parte del contenido representado es texto plano que solo puede ser interpretado por las computadoras para mostrarse en un formato agradable para los humanos que lo consumirán. Así, una cantidad importante de información valiosa no puede ser procesada por las máquinas. Para iniciar un cambio en este sentido, un subcampo de la Inteligencia Artificial, la Representación del Conocimiento, ha proporcionado las bases necesarias. Sobre la base de este campo, Tim Berners-Lee propuso en 2001 la Web Semántica ([18]) como una nueva versión de Internet, en la que las máquinas podrían "comprender" lo que almacenan y representan. Esta comprensión es posible si la web cuenta con contenido estructurado y reglas de inferencia que permitan razonar automáticamente sobre el contenido representado. Si bien esta visión fue revolucionaria en su época, la web semántica ha evolucionado de manera significativa desde entonces, impulsada por los avances realizados en el campo de los datos vinculados, la inteligencia artificial y las tecnologías de integración de datos. Avances recientes, como la adopción de lenguajes subyacentes en la construcción de los grafos de la web semántica y la integración de lenguajes de restricciones para la validación de restricciones semánticas, demuestran la relevancia continua de las tecnologías de la web semántica ([54]).

Además, la principal tecnología relacionada con la web semántica es el "Resource Description Framework" (RDF), que representa los conceptos del mundo real en forma de tripletas (s,p,o). Los grafos de la web semántica que representan las tripletas RDF son también conocidos e interpretados en la actualidad como Grafos de Conocimiento. Estas representaciones se definen como grafos formales, etiquetados y dirigidos, donde los nodos representan conceptos del mundo real, mientras que las aristas representan las relaciones existentes entre los conceptos. Normalmente, estos modelos representan el conocimiento a nivel de instancia, con una capacidad de expresión más débil (en comparación con las ontologías) y a menudo se alimentan de fuentes heterogéneas ([86]). Otra familia de tecnologías, como el "Web Ontology Language" (OWL), se utiliza para la creación de ontologías, que son representaciones conceptuales, formales y compartidas de una parte del mundo, con una rica expresividad que permite representar una gran cantidad de características ([111]). Por otra parte, los grafos de conocimiento y las ontologías están normalmente diseñados para ser interpretados con las hipótesis de "Open World Assumption" (OWA) y "Non-Unique Name Assumption" (NUNA). La primera estipula que un hecho que no aparece en las definiciones explícitas no puede definirse como falso, sino como desconocido. Además, la segunda estipula que es posible tener dos entidades con nombres diferentes que se refieren al mismo concepto del mundo real. Estas hipótesis definen un marco para la inferencia de nuevos conocimientos basados en los hechos explícitos ([164]).

Adicionalmente, es común que los sistemas involucrados presenten una arquitectura de web semántica, que implica la integración y el procesamiento de datos provenientes de fuentes het-

erogéneas en un grafo de conocimiento RDF. En estos escenarios, los datos pueden presentarse en diferentes formatos, como el lenguaje de marcado extensible (XML), la notación de objetos de JavaScript (JSON), los valores separados por comas (CSV), las bases de datos relacionales (RDB) o las tripletas RDF. Además, estos sistemas podrían involucrar diferentes modelos, como ontologías, grafos de Shapes o incluso esquemas XML. Sin embargo, los modelos presentados y los sistemas mencionados todavía presentan algunos inconvenientes, como la complejidad relacionada con la integración de fuentes heterogéneas, las dificultades para representar y almacenar eficazmente los conocimientos, los problemas de escalabilidad al representar datos voluminosos, el tratamiento de conocimientos incompletos y la validación de los datos de instancia representados ([85]).

Para aumentar la adopción de estos modelos, es esencial continuar la investigación en este campo y llegar a propuestas de soluciones. En particular, en lo que respecta a la validación de los modelos, existen muchos campos en los que la coherencia de los modelos tiene una importancia vital debido a la sensibilidad de los resultados, o incluso por el simple deseo de contar con un modelo lo más fiable, eficiente y eficaz posible. Estos requisitos han puesto en evidencia la necesidad de poder añadir coherencia a los modelos mediante la creación de un proceso que pueda validar los modelos de manera cerrada, como ocurre en las bases de datos relacionales (RDB). En consecuencia, la investigación relacionada se ha centrado en el desarrollo de modelos de representación del conocimiento que adhieran a la "Closed World Assumption" (CWA), donde los hechos que no están definidos explícitamente se consideran falsos, y a la "Unique Name Assumption" (UNA), que garantiza que cada concepto del mundo real esté designado por un nombre único. Estos modelos se han propuesto para resolver los problemas de restricción y validación de los modelos de la web semántica, que generalmente se basan en el OWA y el UNA ([154], [108]).

En los últimos años, se han propuesto algunos enfoques para poder validar los modelos de representación del conocimiento de manera cerrada. Más concretamente, se ha puesto el foco en la validación de los grafos RDF. Estos enfoques han evolucionado, pasando de métodos centrados en la generación de consultas SPARQL ("SPARQL Protocol and RDF Query Language", el lenguaje de consulta estándar para RDF) o la ampliación de los modelos de lenguaje existentes, como OWL, con nuevas características relacionadas, al desarrollo de lenguajes completamente nuevos capaces de representar las restricciones de un modelo. Este último tipo de enfoque ha dado lugar a la creación de algunos lenguajes de restricciones, como "Shape Expressions" (ShEX), "SPARQL Inferencing Notation" (SPIN) y "Shapes Constraint Language" (SHACL) ([98]), siendo este último definido como el estándar por el "World Wide Web Consortium" (W3C). Los modelos SHACL también se conocen como "grafos de Shapes SHACL", ya que este lenguaje sigue una estructura de "Shapes" que representan grupos de restricciones expresadas en forma de un grafo RDF que apunta y restringe uno o varios nodos de un grafo de datos, que es en sí mismo otro grafo RDF que representa el conocimiento de la instancia.

Marco teórico y objetivos

Esta tesis es una contribución a la Escuela Francesa de Esquí (ESF), que es una empresa francesa dedicada a la gestión de estaciones de esquí en Francia. El objetivo de este proyecto era construir un entorno relacionado con la Web Semántica, que representara el dominio específico de la empresa con modelos de representación del conocimiento que involucraran los niveles de representación de instancia, conceptual y de restricción, sobre la base de fuentes heterogéneas existentes. Además, hay un interés particular en la validación de las instancias de conocimiento

vinculadas, teniendo un escenario en el que existen más de un grafo de Shapes y donde las restricciones definidas requieren flexibilidad para ser adaptadas a diferentes contextos posibles.

En términos generales, el objetivo de esta tesis es desarrollar enfoques para construir un grafo de Shapes unificado en un entorno semántico caracterizado por fuentes heterogéneas. Además, estas fuentes a menudo conducen a la generación de varios grafos de Shapes, y el escenario puede implicar diversos requisitos contextuales. En este contexto, los objetivos de la tesis son los siguientes:

1. Analizar y describir los trabajos relacionados con la validación que están directa o indirectamente implicados en la resolución del problema de coherencia en los modelos de representación del conocimiento.

2. Sobre la base de las normas SHACL, proporcionar un diseño de grafo de Shapes que pueda adaptarse a diferentes contextos.

3. Proponer un método de integración de los grafos de Shapes SHACL que genere un grafo de Shapes SHACL unificado que respete el diseño presentado en el objetivo anterior.

4. Mostrar su potencial utilidad en un escenario práctico.

Resultados

Primero, en el capítulo 3, se realizó una revisión sistemática de la literatura con el fin de proporcionar una descripción del ámbito relacionado con la validación de las restricciones semánticas en los modelos de representación del conocimiento. Este estudio resultó en la definición de una taxonomía relacionada que incluye 18 categorías de diferentes tipos de trabajos conexos. Además, los artículos seleccionados en el estudio fueron clasificados en estas categorías y se enumeraron los desafíos actuales asociados. Asimismo, se destacaron los desafíos abiertos de investigación que implican las metodologías para la creación de grafos de Shapes SHACL entre los demás desafíos abiertos encontrados. Después de profundizar en esta última, se describió la necesidad de una interpretación de los grafos de Shapes SHACL capaz de adaptarse a diferentes contextos, así como la necesidad de integración de múltiples grafos de Shapes SHACL. Estas constituyen los dos principales desafíos de investigación adicionales que se abordaron en esta tesis (además de la taxonomía mencionada anteriormente).

En segundo lugar, en el capítulo 4, se propuso el modelo de restricciones integradas (MRI) para ayudar a los grafos de Shapes SHACL a validar múltiples contextos. Basada en los estándares SHACL, esta interpretación del modelo es capaz de representar grupos de Shapes que tienen el mismo objetivo pero con diferentes restricciones, lo que podría generar conflictos entre ellas. Además, estos grupos de Shapes están organizados jerárquicamente en una estructura en la que una Shape dentro del grupo está contenida en su super-Shape. Asimismo, este capítulo presentó el proceso de configuración de pre-validación de un MRI, que permite activar el subgrafo del MRI que mejor se ajusta a las características del contexto actual y a las preferencias del usuario. Por otro lado, se consideró una posible representación del conocimiento del contexto, que incluye una ontología y un grafo RDF. Adicionalmente, en el marco de este proceso, se propuso un conjunto de consultas SPARQL para la adquisición de las características del contexto y para la manipulación del MRI, respetando las características recuperadas. Finalmente, este capítulo presentó la estructura de la ontología de procedencia, diseñada para representar el conocimiento relacionado con cada integración específica que ha resultado en una Shape MRI

particular. Gracias a la población de esta ontología, es posible vincular las Shapes específicas del MRI con su respectivo grafo de procedencia.

Tercero, en el capítulo 5, se propuso un procedimiento de integración de múltiples grafos de Shapes SHACL que forman parte de un mismo entorno. Este procedimiento toma como entrada un grafo de Shapes SHACL y un MRI (modelo presentado en el capítulo 4) y genera un MRI actualizado como resultado, que podría incluir nuevos conocimientos derivados de la integración de la versión anterior del MRI con la nueva entrada procesada. Como parte del procedimiento de integración, se propuso un proceso de detección de equivalencia entre Shapes que tienen el mismo objetivo. Además, para la integración de Shapes equivalentes, se presentó una estrategia de resolución de conflictos e integración. Esta estrategia define un conjunto de escenarios relacionados con ciertos parámetros que pueden encontrarse al comparar las restricciones a integrar. Cada escenario incluye un conjunto de acciones específicas que resuelven los conflictos de restricciones mediante la especialización de las Shapes, generando así, paso a paso, una estructura MRI como resultado. Los parámetros considerados en la definición de los escenarios incluyen el tipo de cada restricción SHACL, los valores específicos asignados a ellas y la relación de contención existente entre ellas. Asimismo, las acciones de los escenarios también están influenciadas por la estructura temporal de las Shapes generadas en la integración actual.

En cuarto lugar, en el capítulo 6, se describieron dos casos de uso prácticos relacionados con la Escuela Francesa de Esquí para demostrar los beneficios de las contribuciones presentadas en esta tesis. El primer caso detalla cómo la empresa gestiona el conocimiento proveniente de diferentes estaciones de esquí, respetando un contexto formal subyacente mientras presenta diferencias relacionadas con las características específicas de cada una. Al contar con diferentes grafos de Shapes que representan las restricciones de las estaciones individuales, este caso de uso es útil para probar el procedimiento de integración presentado en esta tesis. A través de la generación de un MRI, la empresa pudo obtener un modelo de restricción semántica que abarca todo el universo de objetivos de la ESF, ofreciendo opciones específicas para algunos de ellos. El segundo caso de uso destaca la existencia de diferentes contextos externos, como el clima, que pueden influir en el funcionamiento de una estación de esquí. Tomando en cuenta el MRI generado durante los experimentos aplicados al primer caso de uso, así como las restricciones recomendadas para un contexto meteorológico específico, como un día de tormenta, se pudo evaluar el procedimiento de pre-validación para seleccionar el subgrafo MRI más adecuado al contexto particular y a las preferencias del usuario. Además, se describieron los detalles de la implementación de las contribuciones, incluyendo los modelos de representación del conocimiento creados, el procedimiento de configuración de pre-validación (junto con los modelos involucrados) y el proceso de integración de los grafos de Shapes SHACL. Finalmente, con base en las implementaciones descritas y los casos de uso presentados, se llevaron a cabo una serie de experimentos para evaluar las contribuciones propuestas. El procedimiento de integración mostró un buen rendimiento al integrar los grafos de Shapes en menos de un segundo. Asimismo, el MRI generado cumplió con los estándares SHACL y pasó con éxito las verificaciones de coherencia, sintaxis y corrección. En cuanto al procedimiento de pre-validación, la activación se realizó correctamente, confirmando la utilidad del diseño del MRI. El modelo activado también superó las pruebas de coherencia. Sin embargo, se identificó que el rendimiento de este procedimiento podría mejorarse, ya que la ejecución de las tareas del flujo de trabajo cubiertas por la implementación actual tomó 5 segundos.

Perspectivas y direcciones futuras

Cada una de las contribuciones de esta tesis ha dado lugar a varias oportunidades para un desarrollo futuro. En lo que respecta a la taxonomía definida en el capítulo 3, como uno de los resultados de la revisión sistemática de la literatura, se han descrito una serie de oportunidades de investigación relacionadas con la validación de restricciones semánticas en los modelos de representación del conocimiento en la sección 3.6.2.

En lo que respecta al modelo integrado de restricciones presentado en el capítulo 4, además de su uso para la validación RDF, podrían estudiarse otras aplicaciones. Más específicamente, el modelo MRI podría estar involucrado en procesos relacionados con la descripción de esquemas, el soporte de consultas, la extracción de subgrafos RDF, la construcción de la interfaz de usuario y la orientación de la evolución de los grafos RDF.

Además, el proceso de configuración pre-validación presentado en 4.2.2 ofrece un espacio para varias propuestas de mejora. En primer lugar, es necesario agregar otros tipos de restricciones a la implementación, desarrollar la generación de informes relacionados con las Shapes candidatas y su presentación al usuario para disponer de una implementación completa del flujo de trabajo. Además, como se mencionó en el capítulo de evaluación, las prestaciones de la implementación actual deben mejorarse, por ejemplo, reduciendo el número de consultas ejecutadas. En segundo lugar, se podrían investigar alternativas para la activación del MRI basadas en otros parámetros, como el nivel deseado de dureza de la Shape o incluso mediante un análisis previo de los datos RDF que se deseen validar. Además, sería interesante tener la posibilidad de representar gráficamente el MRI en una interfaz visual atractiva e intuitiva. Así, el usuario comprendería más fácilmente el modelo de restricciones general y contaría con una forma interactiva de seleccionar las Shapes deseadas, simplemente haciendo clic en las opciones o incluso en Shapes individuales. Además, sería muy interesante contar con un procedimiento de activación automática que no requiera intervención del usuario para aceptar el modelo. El procedimiento podría tener la capacidad de tomar decisiones sobre las Shapes candidatas, considerando conocimientos como las ejecuciones anteriores del contexto específico o las relaciones de jerarquía de las Shapes candidatas. Por ejemplo, si el nivel de Shapes más restrictivas no genera un informe de validación, no sería necesario verificar las Shapes de niveles superiores. En tercer lugar, se podrían investigar alternativas para la representación de las características del contexto, incluso en las restricciones OWL utilizadas (otras que equivalent_to) o directamente en el modelo utilizado para su interpretación. En este sentido, sería interesante simplificar al máximo esta tarea para los expertos en el tema. En cuarto lugar, podría desarrollarse el enriquecimiento del informe presentado al usuario. Una opción interesante sería traducir cada Shape a una consulta SPARQL y, en lugar de generar un informe de validación (que solo informa de los errores), ejecutar la consulta y realizar algunas modificaciones leves en ella, lo que ofrecería más flexibilidad para obtener información adicional valiosa. Por ejemplo, podrían analizarse los complementos (es decir, las Shapes que pasaron la validación) y las relaciones jerárquicas. Finalmente, otra característica interesante que podría estudiarse es el almacenamiento de conocimientos una vez que un subgrafo específico del MRI se activa en un contexto específico. Estos conocimientos podrían luego ser utilizados dentro del flujo de trabajo.

A continuación, la procedencia de integración presentada en el capítulo 5 también ofrece posibilidades de desarrollo. En primer lugar, sería deseable completar la implementación, incluyendo las restricciones SHACL faltantes y el procedimiento de integración de un grafo de Shapes en relación con una versión de la MRI que ya contenga jerarquías con más de una Shape. En segundo lugar, el procedimiento actual de detección de equivalencias es bastante limitado, ya que define las equivalencias de Shapes basándose en la igualdad de los IRI asociados a sus

nodos de focalización (en este caso, las clases) y sus valores de propiedades. Sin embargo, es posible que estos IRI se refieran a elementos diferentes en cada una de las fuentes, especialmente si la integración se realiza entre grafos de Shapes extraídos de sistemas independientes. Para mejorar esta situación y evitar posibles confusiones, sería muy interesante desarrollar un nuevo procedimiento de detección de equivalencia que analice ciertas características de la clase en cada uno de los grafos fuente. Además, si un grafo de Shapes origen se extrae de otra fuente, como una ontología o un grafo RDF, el procedimiento podría implicar el análisis del vecindario examinando las relaciones de entrada/salida y los valores de los nodos relacionados presentes en esas estructuras originales. En tercer lugar, podría estudiarse la inclusión de una clasificación de prioridad, basada en el tipo de restricción SHACL comparada, para mejorar la estrategia de resolución de conflictos. En cuarto lugar, se podría estudiar la minimización del anclaje de las Shapes integradas con estructuras específicas de los datos, generando así una MRI que podría reutilizarse sin depender de la presencia de clases específicas en los datos. Esto se podría lograr mediante la definición de las clases en restricciones basadas en la Shape, lo que significaría que una Shape de nodo tendría la definición de sh:targetClass en otra Shape, que se referenciaría con sh:node. Además, al realizar la configuración pre-validación, si la Shape referenciada está desactivada, la Shape del nodo no se vincularía a una clase específica. Esto no afectaría el proceso de detección de equivalencias, ya que actualmente se tienen en cuenta las definiciones de clase en las Shapes referenciadas y, como alternativa, la definición de sh:group de la MRI también podría usarse para la detección de equivalencias. Por último, podrían desarrollarse otras estrategias de integración y ofrecerse a los usuarios en función de sus objetivos. Un ejemplo sería la creación de una integración alternativa basada en la estrategia de "loose constraining" (restricción flexible) que siempre conserve la restricción más suave y no implique la creación de jerarquías.

Además, podrían estudiarse otras aplicaciones de la ontología de Proveniencia de Fuente presentada en 4.2.3. Esta representación podría utilizarse en una configuración de pre-validación del MRI, en la que el usuario desee activar un subgrafo de Shapes que deban estar relacionadas con una fuente específica. Además, la ontología podría usarse para reconstruir Shapes originales que luego podrían ser utilizadas para diversos fines, como el enriquecimiento del informe presentado al usuario.

Finalmente, otra característica interesante que podría ser estudiada es la inclusión de la técnica de integración en una arquitectura de web semántica más amplia, que implique, como pasos previos, la extracción automática de los grafos de Shapes SHACL a partir de fuentes heterogéneas como las bases de datos relacionales, las ontologías y los grafos RDF.

A

SLR final queries

The tables A.1, A.2 and A.3 present the final queries executed on each scientific database selected for the SLR described in the chapter 3. The use of * in the queries responds to the need in some databases for retrieving all the possibles word terminations after it.

Table A.1: Queries with filters per database (Scopus and Web of Science).

Database	Query
Scopus	TITLE-ABS-KEY(("semantic* constrain*" OR "semantic restriction" OR "ontological constraints" OR shapes OR shacl OR shex) AND ("knowledge formali*" OR "knowledge representation" OR "semantic* network" OR rdf OR owl OR ontology OR "knowledge graph")) AND (LIMIT-TO (PUBYEAR, 2023) OR LIMIT-TO(PUBYEAR, 2022) OR LIMIT-TO(PUBYEAR, 2021) OR LIMIT-TO(PUBYEAR, 2020) OR LIMIT-TO(PUBYEAR, 2019) OR LIMIT-TO(PUBYEAR, 2018) OR LIMIT-TO(PUBYEAR, 2017) OR LIMIT-TO(PUBYEAR, 2016) OR LIMIT-TO(PUBYEAR, 2015) OR LIMIT-TO(PUBYEAR, 2014) OR LIMIT-TO(PUBYEAR, 2013) OR LIMIT-TO(PUBYEAR, 2012) OR LIMIT-TO(PUBYEAR, 2011) OR LIMIT-TO(PUBYEAR, 2010) OR LIMIT-TO(PUBYEAR, 2009) OR LIMIT-TO(PUBYEAR, 2008) OR LIMIT-TO(PUBYEAR, 2007) OR LIMIT-TO(PUBYEAR, 2006) OR LIMIT-TO(PUBYEAR, 2005) OR LIMIT-TO(PUBYEAR, 2004) OR LIMIT-TO(PUBYEAR, 2003) OR LIMIT-TO(PUBYEAR, 2002) OR LIMIT-TO(PUBYEAR, 2001)) AND (LIMIT-TO(DOCTYPE, "ar") OR LIMIT-TO(DOCTYPE, "cp") OR LIMIT-TO(DOCTYPE, "ch") OR LIMIT-TO(DOCTYPE, "re")) AND (LIMIT-TO(SUBJAREA, "COMP")) AND (LIMIT-TO(LANGUAGE, "english"))
WoS	TS=(("Semantic* Constrain*" OR "semantic restriction" OR "Ontological constraints" OR Shapes OR SHACL OR ShEx) AND ("Knowledge Formali*" OR "knowledge representation" OR "semantic* network" OR RDF OR OWL OR ONTOLOGY OR "Knowledge Graph*")) **Filters added**: by year (2001-2023), type of document (article, proceeding paper, review article, book chapter), categories (every category that has the word "Computer Science"), language (English)

Table A.2: Queries with filters per database (IEEE Xplore).

Database	Query
IEEE Xplore	(("Document Title":"semantic* constrain*" OR "Abstract":"semantic* constrain*" OR "Author Keywords":"semantic* constrain*") OR ("Document Title":"semantic restriction*" OR "Abstract":"semantic restriction*" OR "Author Keywords":"semantic restriction*") OR ("Document Title":"ontological constraint*" OR "Abstract":"ontological constraint*" OR "Author Keywords":"ontological constraint*") OR ("Document Title":shapes OR "Abstract":shapes OR "Author Keywords":shapes) OR ("Document Title":shacl OR "Abstract":shacl OR "Author Keywords":shacl) OR ("Document Title":shex OR "Abstract":shex OR "Author Keywords":shex)) AND (("Document Title":"knowledge formali*" OR "Abstract":"knowledge formali*" OR "Author Keywords":"knowledge formali*") OR ("Document Title":"knowledge representation*" OR "Abstract":"knowledge representation*" OR "Author Keywords":"knowledge representation*") OR ("Document Title":"semantic* network*" OR "Abstract":"semantic* network*" OR "Author Keywords":"semantic* network*") OR ("Document Title":rdf* OR "Abstract":rdf* OR "Author Keywords":rdf*) OR ("Document Title":owl* OR "Abstract":owl* OR "Author Keywords":owl*) OR ("Document Title":ontolog* OR "Abstract":ontolog* OR "Author Keywords":ontolog*) OR ("Document Title":"knowledge graph*" OR "Abstract":"knowledge graph*" OR "Author Keywords":"knowledge graph*")) Filters Applied: Conferences,Journals & Magazines, 2010 - 2023

Table A.3: Queries with filters per database (ACM).

Database	Query
ACM	[[Title: "semantic constraint"] OR [Title: "semantic constraints"] OR [Title: "semantic restriction"] OR [Title: "semantic restrictions"] OR [Title: "ontological constraints"] OR [Title: "ontological constraint"] OR [Title: shapes] OR [Title: shacl] OR [Title: shex] OR [Abstract: "semantic constraint"] OR [Abstract: "semantic constraints"] OR [Abstract: "semantic restriction"] OR [Abstract: "semantic restrictions"] OR [Abstract: "ontological constraints"] OR [Abstract: "ontological constraint"] OR [Abstract: shapes] OR [Abstract: shacl] OR [Abstract: shex] OR [Keywords: "semantic constraint"] OR [Keywords: "semantic constraints"] OR [Keywords: "semantic restriction"] OR [Keywords: "semantic restrictions"] OR [Keywords: "ontological constraints"] OR [Keywords: "ontological constraint"] OR [Keywords: shapes] OR [Keywords: shacl] OR [Keywords: shex]] AND [[Title: "knowledge formalisation"] OR [Title: "knowledge representation"] OR [Title: "semantic network"] OR [Title: rdf] OR [Title: owl] OR [Title: ontology] OR [Title: "knowledge graph"] OR [Title: "knowledge graphs"] OR [Abstract: "knowledge formalisation"] OR [Abstract: "knowledge representation"] OR [Abstract: "semantic network"] OR [Abstract: rdf] OR [Abstract: owl] OR [Abstract: ontology] OR [Abstract: "knowledge graph"] OR [Abstract: "knowledge graphs"] OR [Keywords: "knowledge formalisation"] OR [Keywords: "knowledge representation"] OR [Keywords: "semantic network"] OR [Keywords: rdf] OR [Keywords: owl] OR [Keywords: ontology] OR [Keywords: "knowledge graph"] OR [Keywords: "knowledge graphs"]] AND [Publication Date: (01/01/2001 TO 12/31/2023)]

B

Complete constraint conflict resolution by shapes specialization scenarios

The definition of the scenarios that could be found in the constraints comparison task, as well as the corresponding resolution actions that are linked to each scenario, is inspired in the constraint conflict resolution strategy "Resolution by specialization", which was already mentioned in the preliminaries of this chapter. Based on this technique, the following adaptation to shapes is defined:

First, for a matter of clarity, a set of procedures which are used in the scenarios' resolution are described as follows:

The *hierarchy_ reorganization(level)* is an extra procedure which sums up one level of hierarchy to each shape of a specific group, starting by the hierarchy level received in the parameter (level). This change is made by updating the IRI of the respective node shape. As an example, a shape with hierarchy level *sub1* will be updated to the hierarchy level *sub2*. In addition, the name of the shape that is being referenced by *sh:node* is also updated to a hierarchy level which is one level less than the one being previously referenced. As an example, if the shape is referring with *sh:node* to the super shape of the group *(sup)*, then it'll be updated in order to refer to the shape *sub1_ 1*.

The *hierarchy_ simplification* process is another extra procedure, which aims to identify if there are two shapes with a direct relationship sub-sup in the temporal integrated shapes that could be integrated into one without conflict resolution. If this is possible, the sub shape is integrated into its super shape, and then the former is deleted.

The *add_ IRMstructure* procedures have the objectives of setting a new temporal shape of *TSti* with a correct IRI which must respect the *groupID* of the respective group and the correct hierarchy position in which this shape should be added. In addition, these procedures aim to set the correct IRI of the shape that should be referenced with the property sh:node. There are three main procedures of this type which are called depending on the specific temporal structure of shapes that is being generated as the resolution of a conflict. They are detailed as follows:

1. *add_ IRMstructure_ first* sets the *IRI* of the shape as *groupID_ sub1_ 1* and creates the sh:node element referencing to the shape *:groupID_ sup*.

2. *add_ IRMstructure_ second* sets the *IRI* of the shape as *groupID_ sub1_ 2* and creates the sh:node element referencing to the shape *groupID_ sup*.

3. *add_ IRMstructure_ third(level,sibling)* first finds the last sibling of the hierarchy level *(level+1)* namely *((last_ sibling))*. Then, the process updates the *IRI* of the shape as

groupID_sub(level+1)_((last_sibling)+1). Finally, an update of the *IRI* of the shape that is being referenced by sh:node is done as *groupID_sub((level)_(sibling))* being the sibling the one of the parent shape that was divided.

Having presented the procedures, the definition of the scenarios and the related actions follows. Having a constraint *C_inx* which belongs to the shape Sx from the input shape graph and the constraint C_iny which belongs to the shape Sy which belong to the current version of the IRM. In addition, having the temporal structure of integrated shapes TSti (which is empty in the first integration) corresponding to the integration of the current shapes. TSti has the shapes that are generated iteration by iteration as the result of the resolution actions. The integration and conflict resolution procedure presents the following general actions depending on the five possible expected general scenarios.

1. **Identical:** C_inx and C_iny are the same type of constraint and their value spaces are contained in each other, meaning that $C_inx \subseteq C_iny$ and vice-versa (C_inx = C_iny).
   a. If *TSti* has shapes, then *C_inx* is added to the existing temporal shapes.
   b. If *TSti* has no shapes, Then *C_inx* is added to a new temporal shape.
2. **Contradictory:** a *sh:not* constraint is detected in the input shape *Sx*, more precisely in the constraint *C_inx* (which is inside the *sh:property*), then the referenced shape *s1* (as *sh:not s1*) is taken and compared against any shape referenced in the current IRM shape *Sy* (again, each *C_iny* inside the corresponding nested sh:property). If the same shape is referenced first in *C_inx* as (*sh:not s1*) and then in *C_iny* as (*sh:node s1*), then there's a *contradiction*. In this case, the containment relationship is defined as $C_inx \nsubseteq C_iny$ and $C_iny \nsubseteq C_inx$. Both references to the shape *s1* should be linked with an *sh:or* in the super shape. Since *(s1 U not(s1))* represents the *universal set*, then the super shape won't have neither of these constraints and they are directly represented in the sub shapes.
   a. If *TSti* has shapes, then the current super shape of the *TSti* structure is left as it is.
      1. If there exist already sub shapes in the *TSti* structure, then the *hierarchy_reorganization(sub1)* takes place, where *"sub1"* represents the first level of sub shapes. Moreover, the constraint *C_iny* is also added to the shapes which are moved with the hierarchy reorganization process. *C_inx* is added to a new sub shape (*sub1_2*) which is a copy of the previous super shape of *TSti*. This temporal shape is configured with *add_IRMstructure_second*. *C_iny* is added to a new sub shape (*sub1_1*) which is a copy of the previous super shape of *TSti*. This temporal shape is configured with *add_IRMstructure_first*. Finally, the *hierarchy_simplification* process is executed.
      2. If the temporal structure *TSti* has no sub shapes: *C_inx* is added to a new sub shape (*sub1_2*) which is a copy of the previous super shape of *TSti*. This temporal shape is configured with *add_IRMstructure_second*. *C_iny* is added to a new sub shape (*sub1_1*) which is a copy of the previous super shape of *TSti*. This temporal shape is configured with *add_IRMstructure_first*.
   b. If *TSti* has no shapes, then the new temporal super shape of the *TSti* structure is created without the constraints. *C_inx* is added to a new temporal sub shape (*sub1_2*) which is a copy of the previous super shape of *TSti*. This temporal shape is configured with *add_IRMstructure_second*. *C_iny* is added to a new temporal sub

shape (*sub1_1*) which is a copy of the previous super shape of *TSti*. This shape is configured with *add_IRMstructure_first*.

3. **Inconsistent:** *C_inx* and *C_iny* are the same type of constraint, *C_inx* $\nsubseteq$ *C_iny*, *C_iny* $\nsubseteq$ *C_inx* and there's no super constraint value *C3* between the *predefined values* for this specific type of SHACL constraint (the latter applies only for sh:datatype and sh:nodeKind) which contains *C_inx* and *C_iny* at the same time. The same resolution actions apply also to the scenario in which *C_inx* and *C_iny* are different types of constraints and are defined as not compatible with the SHACL syntax rules. Some of these have been detailed in the figure 5.5 ([50]).

   a. If *TSti* has shapes, the constraints *C_inx* and *C_iny* are linked through an *sh:or* and assigned to the current super shape of the *TSti* structure.

      1. If there exist already sub shapes in the *TSti* structure, then the *hierarchy_reorganization(sub1)* takes place, where *"sub1"* represents the first level of sub shapes. Moreover, the constraint *C_iny* is also added to the shapes which are moved with the hierarchy reorganization process. *C_inx* is added to a new sub shape (*sub1_2*) which is a copy of the previous super shape of *TSti* (without the modifications done on this integration). This temporal shape is configured with *add_IRMstructure_second*. *C_iny* is added to a new sub shape (*sub1_1*) which is a copy of the previous super shape of *TSti* (without the modifications done on this integration). This temporal shape is configured with *add_IRMstructure_first*. Finally, the *hierarchy_simplification* process is executed.
      2. If the temporal structure *TSti* has no sub shapes: *C_inx* is added to a new sub shape (*sub1_2*) which is a copy of the previous super shape of *TSti* (without the modifications done on this integration). This temporal shape is configured with *add_IRMstructure_second*. *C_iny* is added to a new sub shape (*sub1_1*) which is a copy of the previous super shape of *TSti* (without the modifications done on this integration). This temporal shape is configured with *add_IRMstructure_first*.

   b. If *TSti* has no shapes, then the constraints *C_inx* and *C_iny* are linked through an *sh:or* and assigned to the new temporal super shape of the *TSti* structure. *C_inx* is added to a new temporal sub shape (*sub1_2*) which is a copy of the previous super shape of *TSti* (without the *sh:or* added). This temporal shape is configured with *add_IRMstructure_second*. *C_iny* is added to a new temporal sub shape (*sub1_1*) which is a copy of the previous super shape of *TSti* (without the *sh:or* added). This shape is configured with *add_IRMstructure_first*.

4. **Independent:** *C_inx* and *C_iny* are not the same type of constraint and their value spaces are compatible (*C_inx != C_iny* and *C_inx != not(C_iny)*). In addition, *C_inx* accomplishes this for each constraint of *C_iny*.

   a. If *TSti* exists and has only one shape, *C_inx* is attempted to be added to the super shape of *TSti* without conflict, then it's possible to add *C_inx* in a new sub shape, that has everything from the original shape + *C_inx*. In addition, this new sub shape is configured with the *add_IRMstructure_first*.

b. If *TSti* exists, has more than one shape and *C_inx* has no conflict with neither of the *TSti* super shape's constraints, then *C_inx* is attempted to be added to the lower hierarchy level of *TSti*.

1. If *C_inx* has no conflict with at least one shape of the lower hierarchy level of *TSti*, then *C_inx* is added to those shapes in which there's no conflict.
2. If *C_inx* has one or more conflicts with all the shapes of this hierarchy level of *TSti*, then *C_inx* is added through an *sh:or* to the super shape of *TSti*. If there's a complementary type of constraint (e.g. *C_inx* = *maxLength* and *C_iny* is *minLength*), the *sh:or* involves only this constraint. Otherwise, the *sh:or* involves all the constraints of the shape.

c. If *TSti* has no shapes, a super shape for *TSti* is created as a copy of the shape Sy, including the structure of the shape (meaning shape name and definition, targets of the shape, property definition with path and sh:group). Then *C_inx* is added to this new shape.

5. **Unidirectional contained:** *C_inx* and *C_iny* are the same type of constraint and their value spaces present a containment relationship where *C_inx* $\subseteq$ *C_iny* or the inverse.

a. IF *C_inx* $\subseteq$ *C_iny*, means also that *C_inx* is stronger than *C_iny*.

1. If *TSti* exists and has only one shape, then *C_iny* is added to the current super shape of *TSti*. *C_inx* is added to a new temporal sub shape (sub1_1) which is a copy of the previous super shape of *TSti* (without the last *C_iny*). This shape is configured with *add_IRMstructure_first*.
2. If *TSti* exists and has more than one shape, *C_iny* is added to the current super shape of *TSti*. *C_inx* is added to all the temporal sub shapes (only if there's no conflict with their constraints).
3. If *TSti* has no shapes, a super shape for *TSti* is created as a copy of the shape Sy, including the structure of the shape (meaning shape name and definition, targets of the shape, property definition with path and sh:group). *C_iny* is added to the new super shape of *TSti*. *C_inx* is added to a new sub shape (sub1_1) which is a copy of the super shape of *TSti* (without the last *C_iny*). This shape is configured with *add_IRMstructure_first*.

b. IF *C_iny* $\subseteq$ *C_inx*.

1. If *TSti* exists and has only one shape, then *C_inx* is added to the current super shape of *TSti*. *C_iny* is added to a new temporal sub shape (sub1_1) which is a copy of the previous super shape of *TSti* (without the last *C_inx*). This shape is configured with*add_IRMstructure_first*.
2. If *TSti* exists and has more than one shape, *C_inx* is added to the current super shape of *TSti*. *C_iny* is added to all the temporal sub shapes (only if there's no conflict with their constraints).
3. If *TSti* has no shapes, a super shape for *TSti* is created as a copy of the shape Sy, including the structure of the shape (meaning shape name and definition, targets of the shape, property definition with path and sh:group). *C_inx* is added to the new super shape of *TSti*. *C_iny* is added to a new sub shape (sub1_1) which is a copy of the super shape of *TSti* (without the last *C_inx*). This shape is configured with *add_IRMstructure_first*.

6. **Compatible with super predefined value:** *C_inx* and *C_iny* are the same type of constraint and they are not compatible, meaning that they have values that target different spaces. Furthermore, there exists a super constraint value *C3* between the *predefined values* for this specific SHACL constraint (applies for *sh:datatype* and *sh:nodeKind*), which contains *C_inx* and *C_iny*, such as *C_inx* $\subseteq$ C3 and *C_iny* $\subseteq$ *C3*.

    a. If *TSti* exists and has only one shape, then C3 is assigned to the super shape of *TSti*. *C_inx* is added to a new sub shape *(sub1_1)* which is a copy of the previous super shape of *TSti* (without the *C3*). This shape is configured with *add_IRMstructure_first*. *C_iny* is added to a new sub shape *(sub1_2)* which is a copy of the previous super shape of *TSti* (without the *C3*). This shape is configured with *add_IRMstructure_second*.

    b. If *TSti* exists and has more than one shape, *C3* is added to all the sub shapes of the lower hierarchy level of *TSti* (only if *C3* doesn't have a conflict with any of the constraints of the specific sub shape). Each shape of the lower hierarchy level of each branch that was updated is taken and divided into two new sub shapes (copying every element except *C3*), where the first one will have *C_inx* and the second one *C_iny*. The organizational data will be updated with the *add_IRMstructure_third(level,sibling)*.

    c. If *TSti* has no shapes, a super temporal shape for *TSti* is created as a copy of the Sy, including the structure of the shape (meaning shape name and definition, targets of the shape, property definition with path and *sh:group*). *C3* is added to the new super shape of *TSti*. *C_inx* is added to a new sub shape *(sub1_1)* which is a copy of the new super shape of *TSti* (without the *C3* added). This shape is configured with *add_IRMstructure_first*. *C_iny* is added to a new sub shape *(sub1_2)* which is a copy of the new super shape of *TSti* (without the *C3* added). This shape is configured with *add_IRMstructure_second*.

It's worth to mention that for those cases in which the resolution action links the integrated constraints with a sh:or, the addition of the latter property depends also on the type of constraint that is being integrated. There are some specific constraints for which this element is not necessary, such as with sh:in, in which case just adding the values to the assigned list (sh:in value1,value2,...) is enough, or as in sh:hasValue, in which case just adding one sh:hasValue for each different value is interpreted as a sh:or between the different definitions for this specific constraint.

Finally, when a shape from the input shape graph finds an equivalence with an existing group of the current IRM graph which has already a hierarchy of shapes, then some slight differences appear in the process. The integration algorithm will always execute the option in which there existed already a temporal *TSti* structure with sub shapes. The comparison is done against the super shape of the IRM group. The super shape of the IRM will be generated in a new version as the outcome of the integration. The rest of the shapes are automatically taken as part of the temporal shape structure.

Furthermore, there is the possibility in which there's a constraint *C_iny* which didn't have a conflict with any of the *C_inx* constraints. In order to not lose these constraints, then *C_iny* is attempted to be added to the super shape and goes through the conflict resolution process. In addition, the number of conflicts that a constraint has could also change the order in which a constraint goes through the process. For example, if a constraint *C_inx*, during the current integration, previously had already a conflict with a constraint *C_iny* and currently has a conflict

with a second constraint *C_iny*, then instead of trying to add *C_inx* to the current super temporal shape of *TSti*, *C_iny* is the constraint that is attempted to be added.

Along the whole design of the integration process, there are some conditions which are added in order to ensure that the generated structure of shapes will respect the structure of an IRM defined in the previous chapter. In addition, some rules aim to delimit the number of shapes generated in order to obtain an IRM which has a manageable size. However, these rules could be easily changed in order to delete these constraints that delimit the outcome. The detail of the scenarios is summarized in the figure 5.6.

C

List of Abbreviations

- **ASP** Answer Set Programming
- **CL** Constraint Language
- **CRIP** Conflict Resolution and Integration Process
- **CSV** Comma-Separated Values
- **CWA** Closed World Assumption
- **EC** Exclusion Criteria
- **ESF** École du ski français
- **IC** Inclusion Criteria
- **IRM** Integrated Restrictions Model
- **IRI** International Resource Identifier
- **JSON** JavaScript Object Notation
- **KG** Knowledge Graph
- **NUNA** Non-Unique Name Assumption
- **OWL** Web Ontology Language
- **OWA** Open World Assumption
- **R2RML** RDB to RDF Mapping Language
- **RDB** Relational Database
- **RDD** RDF Data Description
- **RDF** Resource Description Framework
- **RDFS** Resource Description Framework Schema
- **RML** RDF Mapping Language

- **RQ** Research Question
- **SHACL** Shapes Constraint Language
- **SHEX** Shape Expressions
- **SLR** Systematic Literature Review
- **SPARQL** SPARQL Protocol and RDF Query Language
- **SPIN** SPARQL Inferencing Notation
- **SQL** Structured Query Language
- **SSG** Source Shape Graphs
- **SWRL** Semantic Web Rule Language
- **UNA** Unique Name Assumption
- **W3C** World Wide Web Consortium
- **XML** Extensible Markup Language
- **XSD** XML Schema Definition

Bibliography

[1] Abbas, A., Genevès, P., Roisin, C., Layaïda, N., 2017a. Optimising SPARQL Query Evaluation in the Presence of ShEx Constraints, in: BDA 2017 - 33ème conférence sur la " Gestion de Données - Principes, Technologies et Applications ", Nancy, France. pp. 1–12. URL: https://inria.hal.science/hal-01673067.

[2] Abbas, A., Genevès, P., Roisin, C., Layaïda, N., 2017b. Sparql query containment with shex constraints, in: Kirikova, M., Nørvåg, K., Papadopoulos, G.A. (Eds.), Advances in Databases and Information Systems, Springer International Publishing, Cham. pp. 343–356.

[3] Abbas, A., Genevès, P., Roisin, C., Layaïda, N., 2018. Selectivity estimation for sparql triple patterns with shape expressions, in: Mikkonen, T., Klamma, R., Hernández, J. (Eds.), Web Engineering, Springer International Publishing, Cham. pp. 195–209. doi:https://doi.org/10.1007/978-3-319-91662-0_15.

[4] Abián, D., Bernad, J., Trillo-Lado, R., 2019. Using contemporary constraints to ensure data consistency, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Association for Computing Machinery, New York, NY, USA. p. 2303–2310. URL: https://doi.org/10.1145/3297280.3297509, doi:10.1145/3297280.3297509.

[5] Aguilar, J., Valdiviezo-Díaz, P., Riofrio, G., 2017. A general framework for intelligent recommender systems. Applied Computing and Informatics 13, 147–160. URL: https://www.sciencedirect.com/science/article/pii/S2210832716300254, doi:https://doi.org/10.1016/j.aci.2016.08.002.

[6] Ahmetaj, S., David, R., Ortiz, M., Polleres, A., Shehu, B., Šimkus, M., 2021. Reasoning about Explanations for Non-validation in SHACL, in: Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, pp. 12–21. URL: https://doi.org/10.24963/kr.2021/2, doi:10.24963/kr.2021/2.

[7] Ahmetaj, S., David, R., Polleres, A., Šimkus, M., 2022a. Repairing shacl constraint violations using answer set programming, in: Sattler, U., Hogan, A., Keet, M., Presutti, V., Almeida, J.P.A., Takeda, H., Monnin, P., Pirrò, G., d'Amato, C. (Eds.), The Semantic Web – ISWC 2022, Springer International Publishing, Cham. pp. 375–391. doi:https://doi.org/10.1007/978-3-031-19433-7_22.

[8] Ahmetaj, S., Löhnert, B., Ortiz, M., Šimkus, M., 2022b. Magic shapes for shacl validation. Proc. VLDB Endow. 15, 2284–2296. URL: https://doi.org/10.14778/3547305.3547329, doi:10.14778/3547305.3547329.

[9] Ait-Ameur, Y., Méry, D., 2016. Making explicit domain knowledge in formal system development. Science of Computer Programming 121, 100–127. URL: https://www.sciencedirect.com/science/article/pii/S0167642315004153, doi:https://doi.org/10.1016/j.scico.2015.12.004. special Issue on Knowledge-based Software Engineering.

[10] Akhtar, W., Cortés-Calabuig, Á., Paredaens, J., 2011. Constraints in rdf, in: Schewe, K.D., Thalheim, B. (Eds.), Semantics in Data and Knowledge Bases, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 23–39. doi:https://doi.org/10.1007/978-3-642-23441-5_2.

[11] Akkermans, H., Harmelen, F., Schreiber, G., Wielinga, B., 1992. Formalization of knowledge-level models for knowledge acquisition. International Journal of Intelligent Systems - IJIS 8. doi:10.5555/171647.171659.

[12] Amara, F.Z., Djezzar, M., Hemam, M., Tiwari, S., Hafidi, M.M., 2023. Unlocking the power of semantic interoperability in industry 4.0: A comprehensive overview, in: Ortiz-Rodriguez, F., Villazón-Terrazas, B., Tiwari, S., Bobed, C. (Eds.), Knowledge Graphs and Semantic Web, Springer Nature Switzerland, Cham. pp. 82–96.

[13] Andresel, M., Corman, J., Ortiz, M., Reutter, J.L., Savkovic, O., Simkus, M., 2020. Stable model semantics for recursive shacl, in: Proceedings of The Web Conference 2020, Association for Computing Machinery, New York, NY, USA. p. 1570–1580. URL: https://doi.org/10.1145/3366423.3380229, doi:10.1145/3366423.3380229.

[14] Arenas, M., Díaz, G., Fokoue, A., Kementsietsidis, A., Srinivas, K., 2014. A principled approach to bridging the gap between graph data and their schemas. Proc. VLDB Endow. 7, 601–612. URL: https://doi.org/10.14778/2732296.2732297, doi:10.14778/2732296.2732297.

[15] Balduccini, M., Baral, C., Lierler, Y., 2008. Chapter 20 knowledge representation and question answering, in: van Harmelen, F., Lifschitz, V., Porter, B. (Eds.), Handbook of Knowledge Representation. Elsevier. volume 3 of *Foundations of Artificial Intelligence*, pp. 779–819. URL: https://www.sciencedirect.com/science/article/pii/S1574652607030209, doi:https://doi.org/10.1016/S1574-6526(07)03020-9.

[16] Banerjee, A., Dalal, R., Mittal, S., Joshi, K.P., 2017. Generating digital twin models using knowledge graphs for industrial production lines, in: Proceedings of the 2017 ACM on Web Science Conference, Association for Computing Machinery, New York, NY, USA. p. 425–430. URL: https://doi.org/10.1145/3091478.3162383, doi:10.1145/3091478.3162383.

[17] Berges, I., Ramírez-Durán, V.J., Illarramendi, A., 2021. A semantic approach for big data exploration in industry 4.0. Big Data Research 25, 100222. URL: http://dx.doi.org/10.1016/j.bdr.2021.100222, doi:10.1016/j.bdr.2021.100222.

[18] Berners-Lee, T., Hendler, J., Lassila, O., 2001. The semantic web. Scientific American 284, 34–43. URL: http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21.

[19] Bizer, C., Heath, T., Berners-Lee, T., 2023. Linked Data - The Story So Far. 1 ed.. Association for Computing Machinery, New York, NY, USA. p. 115–143. URL: https://doi.org/10.1145/3591366.3591378, doi:10.4018/jswis.2009081901.

[20] Bogaerts, B., Jakubowski, M., 2021. Fixpoint semantics for recursive shacl. doi:10.4204/EPTCS.345.14.

[21] Bogaerts, B., Jakubowski, M., Van den Bussche, J., 2022. Shacl: A description logic in disguise, in: Gottlob, G., Inclezan, D., Maratea, M. (Eds.), Logic Programming and Nonmonotonic Reasoning, Springer International Publishing, Cham. pp. 75–88. doi:https://doi.org/10.1007/978-3-031-15707-3_7.

[22] Bogaerts, B., Jakubowski, M., den Bussche, J.V., 2023. Expressiveness of shacl features and extensions for full equality and disjointness tests. URL: https://arxiv.org/abs/2212.03553, doi:https://doi.org/10.46298/lmcs-20(1:16)2024, arXiv:2212.03553.

[23] Boneva, I., Labra Gayo, J.E., Prud'hommeaux, E.G., 2017. Semantics and validation of shapes schemas for rdf, in: d'Amato, C., Fernandez, M., Tamma, V., Lecue, F., Cudré-Mauroux, P., Sequeda, J., Lange, C., Heflin, J. (Eds.), The Semantic Web – ISWC 2017, Springer International Publishing, Cham. pp. 104–120. doi:https://doi.org/10.1007/978-3-319-68288-4_7.

[24] Boneva, I., Staworko, S., Lozano, J., 2020. Consistency and certain answers in relational to rdf data exchange with shape constraints, in: Darmont, J., Novikov, B., Wrembel, R. (Eds.), New Trends in Databases and Information Systems, Springer International Publishing, Cham. pp. 97–107. doi:https://doi.org/10.1007/978-3-030-54623-6_9.

[25] Bosch, T., Acar, E., Nolle, A., Eckert, K., 2015a. The role of reasoning for rdf validation, in: Proceedings of the 11th International Conference on Semantic Systems, Association for Computing Machinery, New York, NY, USA. p. 33–40. URL: https://doi.org/10.1145/2814864.2814867, doi:10.1145/2814864.2814867.

[26] Bosch, T., Eckert, K., 2014. Towards description set profiles for rdf using sparql as intermediate language, in: Proceedings of the 2014 International Conference on Dublin Core and Metadata Applications, Dublin Core Metadata Initiative. p. 129–137. doi:10.5555/2771234.2771248.

[27] Bosch, T., Nolle, A., Acar, E., Eckert, K., 2015b. Rdf validation requirements - evaluation and logical underpinning. URL: https://arxiv.org/abs/1501.03933, doi:https://doi.org/10.48550/arXiv.1501.03933, arXiv:1501.03933.

[28] Boufarès, F., Ben Salem, A., 2012. Heterogeneous data-integration and data quality: Overview of conflicts, pp. 867–874. doi:10.1109/SETIT.2012.6482029.

[29] Brachman, R.J., Levesque, H.J., 1985. Readings in Knowledge Representation. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. doi:10.5555/577033.

[30] Braun, C.H.J., Papanchev, V., Käfer, T., 2023. Sissi: An architecture for semantic interoperable self-sovereign identity-based access control on the web, in: Proceedings of the ACM Web Conference 2023, Association for Computing Machinery, New York, NY, USA. p. 3011–3021. URL: https://doi.org/10.1145/3543507.3583409, doi:10.1145/3543507.3583409.

[31] Brickley, D., Guha, R., 2000. Resource description framework schema (rdf/s) specification 1.1 URL: https://www.w3.org/TR/rdf-schema/.

[32] Chah, N., 2018. Ok google, what is your ontology? or: Exploring freebase classification to understand google's knowledge graph. URL: https://arxiv.org/abs/1805.03885, arXiv:1805.03885.

[33] Chmurovic, A., Simkus, M., 2022. Well-founded semantics for recursive SHACL, in: Alviano, M., Pieris, A. (Eds.), Proceedings of the 4th International Workshop on the Resurgence of Datalog in Academia and Industry (Datalog-2.0 2022) co-located with the 16th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2022), Genova-Nervi, Italy, September 5, 2022, CEUR-WS.org. pp. 2–13. URL: https://ceur-ws.org/Vol-3203/paper1.pdf.

[34] Cimmino, A., Fernández-Izquierdo, A., García-Castro, R., 2020. Astrea: Automatic generation of shacl shapes from ontologies, in: Harth, A., Kirrane, S., Ngonga Ngomo, A.C., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M. (Eds.), The Semantic Web, Springer International Publishing, Cham. pp. 497–513.

[35] Console, M., Lenzerini, M., 2020. Epistemic integrity constraints for ontology-based data management. Proceedings of the AAAI Conference on Artificial Intelligence 34, 2790–2797. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5667, doi:10.1609/aaai.v34i03.5667.

[36] Corcho, O., Poveda-Villalón, M., Gomez-Perez, A., 2015. Ontology engineering in the era of linked data: Ontology engineering in the era of linked data. Bulletin of the American Society for Information Science and Technology 41. doi:10.1002/bult.2015.1720410407.

[37] Corman, J., Florenzano, F., Reutter, J.L., Savković, O., 2019. Validating shacl constraints over a sparql endpoint, in: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham. pp. 145–163.

[38] Corman, J., Reutter, J.L., Savković, O., 2018. Semantics and validation of recursive shacl, in: Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.A., Simperl, E. (Eds.), The Semantic Web – ISWC 2018, Springer International Publishing, Cham. pp. 318–336.

[39] Cyganiak, R., Hyland-Wood, D., Lanthaler, M., 2014. Rdf 1.1 concepts and abstract syntax. W3C Proposed Recommendation .

[40] Dalponte Ayastuy, M., Torres, D., Fernández, A., 2021. Adaptive gamification in collaborative systems, a systematic mapping study. Computer Science Review 39, 100333. URL: https://www.sciencedirect.com/science/article/pii/S1574013720304330, doi:https://doi.org/10.1016/j.cosrev.2020.100333.

[41] Dam, J., Koehorst, J., Schaap, P., Santos, V., Suarez-Diez, M., 2015. Rdf2graph a tool to recover, understand and validate the ontology of an rdf resource. Journal of biomedical semantics 6, 39. doi:10.1186/s13326-015-0038-9.

[42] Davis, R., Shrobe, H., Szolovits, P., 1993. What is a knowledge representation? AI Magazine 14, 17. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1029, doi:10.1609/aimag.v14i1.1029.

[43] De Meester, B., Heyvaert, P., Arndt, D., Dimou, A., Verborgh, R., 2020. Rdf graph validation using rule-based reasoning. Semantic Web 12, 117–142. doi:10.3233/SW-200384.

[44] De Meester, B., Heyvaert, P., Dimou, A., Verborgh, R., 2018. Towards a uniform user interface for editing data shapes, in: Ivanova, V., Lambrix, P., Lohmann, S., Pesquita, C. (Eds.), Proceedings of the 4th International Workshop on Visualization and Interaction for Ontologies and Linked Data. URL: http://ceur-ws.org/Vol-2187/paper2.pdf.

[45] Delva, T., De Smedt, B., Min Oo, S., Van Assche, D., Lieber, S., Dimou, A., 2021. Rml2shacl: Rdf generation taking shape, in: Proceedings of the 11th Knowledge Capture Conference, Association for Computing Machinery, New York, NY, USA. p. 153–160. URL: https://doi.org/10.1145/3460210.3493562, doi:10.1145/3460210.3493562.

[46] Delva, Thomas and Dimou, Anastasia and Jakubowksi, Maxime and Van den Bussche, Jan, 2023. Data provenance for SHACL, in: Proceedings 26th International Conference on Extending Database Technology (EDBT 2023), pp. 285–297. URL: http://doi.org/10.48786/edbt.2023.23.

[47] Di Ciccio, C., Ekaputra, F.J., Cecconi, A., Ekelhart, A., Kiesling, E., 2019. Finding non-compliances with declarative process constraints through semantic technologies, in: Cappiello, C., Ruiz, M. (Eds.), Information Systems Engineering in Responsible Information Systems, Springer International Publishing, Cham. pp. 60–74. doi:https://doi.org/10.1007/978-3-030-21297-1_6.

[48] Dibowski, H., 2021. Property assertion constraints for an informed, error-preventing expansion of knowledge graphs, in: Villazón-Terrazas, B., Ortiz-Rodríguez, F., Tiwari, S., Goyal, A., Jabbar, M. (Eds.), Knowledge Graphs and Semantic Web, Springer International Publishing, Cham. pp. 234–248. doi:https://doi.org/10.1007/978-3-030-91305-2_18.

[49] Dibowski, H., 2022. Property assertion constraints for ontologies and knowledge graphs. Data Technologies and Applications , 1–20doi:10.1108/DTA-05-2022-0209.

[50] Duan, X., Chaves-Fraga, D., Derom, O., Dimou, A., 2024. Scoop all the constraints' flavours for your knowledge graph, in: Meroño Peñuela, A., Dimou, A., Troncy, R., Hartig, O., Acosta, M., Alam, M., Paulheim, H., Lisena, P. (Eds.), The Semantic Web, Springer Nature Switzerland, Cham. pp. 217–234. doi:10.1007/978-3-031-60635-9_13.

[51] Duan, X., Chaves-Fraga, D., Dimou, A., 2023. Xsd2shacl: Capturing rdf constraints from xml schema, in: Proceedings of the 12th Knowledge Capture Conference 2023, Association for Computing Machinery, New York, NY, USA. p. 214–222. URL: https://doi.org/10.1145/3587259.3627565, doi:10.1145/3587259.3627565.

[52] d'Amato, C., Melo, A., Paulheim, H., 2020. Automatic detection of relation assertion errors and induction of relation constraints. Semant. Web 11, 801–830. URL: https://doi.org/10.3233/SW-200369, doi:10.3233/SW-200369.

[53] Ekaputra, F.J., Lin, X., 2016. Shacl4p: Shacl constraints validation within protégé ontology editor, in: 2016 International Conference on Data and Software Engineering (ICoDSE), pp. 1–6. doi:10.1109/ICODSE.2016.7936162.

[54] Ekaputra, F.J., Llugiqi, M., Sabou, M., Ekelhart, A., Paulheim, H., Breit, A., Revenko, A., Waltersdorfer, L., Farfar, K.E., Auer, S., 2023. Describing and organizing semantic web and

machine learning systems in the swemls-kg. URL: https://arxiv.org/abs/2303.15113, arXiv:2303.15113.

[55] Faraj, G., Micsik, A., 2021. Representing and validating cultural heritage knowledge graphs in cidoc-crm ontology. Future Internet 13. URL: https://www.mdpi.com/1999-5903/13/11/277, doi:10.3390/fi13110277.

[56] Felin, R., Faron, C., Tettamanzi, A.G.B., 2023. A framework to include and exploit probabilistic information in shacl validation reports, in: Pesquita, C., Jimenez-Ruiz, E., McCusker, J., Faria, D., Dragoni, M., Dimou, A., Troncy, R., Hertling, S. (Eds.), The Semantic Web, Springer Nature Switzerland, Cham. pp. 91–104. doi:https://doi.org/10.1007/978-3-031-33455-9_6.

[57] Fenza, G., Gallo, M., Loia, V., Marino, D., Orciuoli, F., Volpe, A., 2020. Semantic CPPS in industry 4.0. CoRR abs/2011.11395. URL: https://arxiv.org/abs/2011.11395, arXiv:2011.11395.

[58] Fernandez-Álvarez, D., Labra-Gayo, J.E., Gayo-Avello, D., 2022. Automatic extraction of shapes using shexer. Knowledge-Based Systems 238, 107975. URL: https://www.sciencedirect.com/science/article/pii/S0950705121010972, doi:https://doi.org/10.1016/j.knosys.2021.107975.

[59] Ferranti, N., Polleres, A., De-Souza, J., Ahmetaj, S., 2022. Formalizing property constraints in wikidata URL: http://hdl.handle.net/20.500.12708/150282.

[60] Ferreirone, M., Lezoche, M., Torres, D., Panetto, H., 2023. Transformations on knowledge representation between OWL and RDF knowledge graphs: A study case. SADIO electronic journal of informatics and operations research 22, 80–93. URL: https://hal.science/hal-04312488.

[61] Figuera, M., Rohde, P.D., Vidal, M.E., 2021. Trav-shacl: Efficiently validating networks of shacl constraints, in: Proceedings of the Web Conference 2021, Association for Computing Machinery, New York, NY, USA. p. 3337–3348. URL: https://doi.org/10.1145/3442381.3449877, doi:10.1145/3442381.3449877.

[62] Fischer, P.M., Lausen, G., Schatzle, A., Schmidt, M., 2015. Rdf constraint checking, in: Proceedings of the Workshops of the EDBT/ICDT 2015 Joint Conference Brussels, Belgium, March 27th, 2015, pp. 205–212. URL: https://ceur-ws.org/Vol-1330/paper-33.pdf.

[63] Fleischhacker, D., Paulheim, H., Bryl, V., Völker, J., Bizer, C., 2014. Detecting errors in numerical linked data using cross-checked outlier detection, in: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (Eds.), The Semantic Web – ISWC 2014, Springer International Publishing, Cham. pp. 357–372. doi:https://doi.org/10.1007/978-3-319-11964-9_23.

[64] Franciosi, C., Voisin, A., Miranda, S., Riemma, S., Iung, B., 2020. Measuring maintenance impacts on sustainability of manufacturing industries: from a systematic literature review to a framework proposal. Journal of Cleaner Production 260, 121065. URL: https://www.sciencedirect.com/science/article/pii/S0959652620311124, doi:https://doi.org/10.1016/j.jclepro.2020.121065.

[65] Fürber, C., Hepp, M., 2010. Using sparql and spin for data quality management on the semantic web, in: Abramowicz, W., Tolksdorf, R. (Eds.), Business Information Systems, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 35–46. doi:https://doi.org/10.1007/978-3-642-12814-1_4.

[66] Gaignard, A., Rosnet, T., Lamotte, F., Lefort, V., Devignes, M.D., 2023. Fair-checker: supporting digital resource findability and reuse with knowledge graphs and semantic web standards. Journal of Biomedical Semantics 14. doi:10.1186/s13326-023-00289-5.

[67] Galkin, M., Auer, S., Scerri, S., 2016. Enterprise knowledge graphs: A backbone of linked enterprise data, pp. 497–502. doi:10.1109/WI.2016.0083.

[68] García-González, H., Labra Gayo, J., 2018. Xmlschema2shex: Converting xml validation to rdf validation. Semantic Web , 1–19doi:10.3233/SW-180329.

[69] Gayo, J., Prud'hommeaux, E., Boneva, I., Staworko, S., Solbrig, H., Hym, S., 2015. Towards an rdf validation language based on regular expression derivatives, in: Proceedings of the Workshops of the EDBT/ICDT 2015 Joint Conference (EDBT/ICDT), Brussels, Belgium, March 27th, 2015., pp. 197–204.

[70] Gayo, J.E.L., 2022. Wshex: A language to describe and validate wikibase entities. URL: https://arxiv.org/abs/2208.02697, doi:https://doi.org/10.48550/arXiv.2208.02697, arXiv:2208.02697.

[71] Giustozzi, F., Saunier, J., Zanni-Merk, C., 2024. A semantic framework for condition monitoring in industry 4.0 based on evolving knowledge bases 15. doi:10.3233/sw-233481.

[72] Gomez-Perez, A., 1999. Ontological engineering: A state of the art. Expert Update 2.

[73] Groz, B., Lemay, A., Staworko, S., Wieczorek, P., 2022. Inference of Shape Graphs for Graph Databases, in: International Conference on Database Theory, Edinbourgh, United Kingdom. URL: https://inria.hal.science/hal-03559309, doi:10.4230/LIPIcs.ICDT.2022.7.

[74] Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220. URL: https://www.sciencedirect.com/science/article/pii/S1042814383710083, doi:https://doi.org/10.1006/knac.1993.1008.

[75] Haav, H.m., Maigre, R., 2019. A semantic model for product configuration in timber industry. doi:10.3233/978-1-61499-941-6-143.

[76] Halfeld-Ferrari, M., Hara, C.S., Uber, F.R., 2017. Rdf updates with constraints, in: Różewski, P., Lange, C. (Eds.), Knowledge Engineering and Semantic Web, Springer International Publishing, Cham. pp. 229–245. doi:https://doi.org/10.1007/978-3-319-69548-8_16.

[77] Hamdan, A.H., Scherer, R., 2020. Integration of bim-related bridge information in an ontological knowledgebase. doi:10.1016/j.advengsoft.2019.02.001.

[78] Hansen, J., Beveridge, A., Farmer, R., Gehrmann, L., Gray, A., Khutan, S., Robertson, T., Val, J., 2015. Validata: An online tool for testing rdf data conformance. URL: https://www.researchgate.net/publication/283906921_Validata_An_online_tool_for_testing_RDF_data_conformance.

[79] Hartig, O., Hidders, J., 2019. Defining schemas for property graphs by using the graphql schema definition language, in: Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), Association for Computing Machinery, New York, NY, USA. URL: https://doi.org/10.1145/3327964.3328495, doi:10.1145/3327964.3328495.

[80] Hartmann, T., Eckert, K., 2014. Requirements on rdf constraint formulation and validation.

[81] Hartmann, T., Eckert, K., 2015. Guidance, please! towards a framework for rdf-based constraint languages. doi:10.5555/2907896.2907906.

[82] Hartmann, T., Zapilko, B., Wackerow, J., Eckert, K., 2015. Constraints to validate rdf data quality on common vocabularies in the social, behavioral, and economic sciences. URL: https://arxiv.org/abs/1504.04479, doi:https://doi.org/10.48550/arXiv.1504.04479, arXiv:1504.04479.

[83] Hartmann, T., Zapilko, B., Wackerow, J., Eckert, K., 2016a. Directing the development of constraint languages by checking constraints on rdf data. International Journal of Semantic Computing 10, 193–217. URL: https://doi.org/10.1142/S1793351X16400079, doi:10.1142/S1793351X16400079, arXiv:https://doi.org/10.1142/S1793351X16400079.

[84] Hartmann, T., Zapilko, B., Wackerow, J., Eckert, K., 2016b. Validating rdf data quality using constraints to direct the development of constraint languages, in: 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), pp. 116–123. doi:10.1109/ICSC.2016.43.

[85] Hogan, A., 2020. The Web of Data. Springer. URL: https://doi.org/10.1007/978-3-030-51580-5, doi:10.1007/978-3-030-51580-5.

[86] Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.C.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A., 2021a. Knowledge graphs. ACM Comput. Surv. 54. URL: https://doi.org/10.1145/3447772, doi:10.1145/3447772.

[87] Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.C.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A., 2021b. Knowledge graphs. ACM Comput. Surv. 54. URL: https://doi.org/10.1145/3447772, doi:10.1145/3447772.

[88] Hou, Z.W., Qin, C.Z., Zhu, A.X., Wang, Y.J., Liang, P., Wang, Y.J., Zhu, Y.Q., 2021. Formalizing parameter constraints to support intelligent geoprocessing: A shacl-based method. ISPRS International Journal of Geo-Information 10. URL: https://www.mdpi.com/2220-9964/10/9/605, doi:10.3390/ijgi10090605.

[89] Idehen, K., 2017. Semantic web layer cake tweak, explained. OpenLink Software, via Medium URL: https://medium.com/openlink-software-blog/semantic-web-layer-cake-tweak-explained-6ba5c6ac3fab.

[90] Jalali, S., Wohlin, C., 2012. Systematic literature studies: Database searches vs. backward snowballing, in: Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 29–38. doi:10.1145/2372251.2372257.

[91] K Soman, R., 2019. Modelling construction scheduling constraints using shapes constraint language (shacl), pp. 351–358. doi:10.35490/EC3.2019.170.

[92] Kellou-Menouer, K., Kedad, Z., 2015. Evaluating the gap between an rdf dataset and its schema, in: Jeusfeld, M.A., Karlapalem, K. (Eds.), Advances in Conceptual Modeling, Springer International Publishing, Cham. pp. 283–292. doi:https://doi.org/10.1007/978-3-319-25747-1_28.

[93] Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2009. Systematic literature reviews in software engineering – a systematic literature review. Information and Software Technology 51, 7–15. URL: https://www.sciencedirect.com/science/article/pii/S0950584908001390, doi:https://doi.org/10.1016/j.infsof.2008.09.009. special Section - Most Cited Articles in 2002 and Regular Research Papers.

[94] Kitchenham, B.A., Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001. Keele University and Durham University Joint Report. URL: https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf.

[95] Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A., 2014. Test-driven evaluation of linked data quality, in: Proceedings of the 23rd International Conference on World Wide Web, Association for Computing Machinery, New York, NY, USA. p. 747–758. URL: https://doi.org/10.1145/2566486.2568002, doi:10.1145/2566486.2568002.

[96] Köcher, A., da Silva, L.M.V., Fay, A., 2021. Constraint checking of skills using shacl, in: 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), pp. 1–6. doi:10.1109/INDIN45523.2021.9557549.

[97] Labra-Gayo, J.E., García-González, H., Fernández-Alvarez, D., Prud'hommeaux, E., 2019. Challenges in RDF Validation. Springer International Publishing, Cham. pp. 121–151. URL: https://doi.org/10.1007/978-3-030-06149-4_6, doi:10.1007/978-3-030-06149-4_6.

[98] Labra Gayo, J.E., Prud'hommeaux, E., Boneva, I., Kontokostas, D., 2017. Validating RDF Data. volume 7 of *Synthesis Lectures on the Semantic Web: Theory and Technology*. Morgan & Claypool Publishers LLC. URL: https://doi.org/10.2200/s00786ed1v01y201707wbe016, doi:10.2200/s00786ed1v01y201707wbe016.

[99] Labra-Gayo, J.E., Prud'hommeaux, E., Solbrig, H., Boneva, I., 2014. Validating and describing linked data portals using shapes. URL: https://arxiv.org/abs/1701.08924, doi:https://doi.org/10.48550/arXiv.1701.08924, arXiv:1701.08924.

[100] Larhrib, M., Escribano, M., Cerrada, C., Escribano, J.J., 2020. Converting ocl and cgmes rules to shacl in smart grids. IEEE Access 8, 177255–177266. doi:10.1109/ACCESS.2020.3026941.

[101] Lausen, G., Meier, M., Schmidt, M., 2008. Sparqling constraints for rdf, in: Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology, Association for Computing Machinery, New York, NY, USA.

p. 499–509. URL: https://doi.org/10.1145/1353343.1353404, doi:10.1145/1353343.1353404.

[102] Lei, X., Wu, P., Zhu, J., Wang, J., 2022. Ontology-based information integration: A state-of-the-art review in road asset management. Archives of Computational Methods in Engineering 29. doi:https://doi.org/10.1007/s11831-021-09668-6.

[103] Leinberger, M., Seifer, P., Rienstra, T., Lämmel, R., Staab, S., 2020. Deciding shacl shape containment through description logics reasoning, in: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham. pp. 366–383. doi:https://doi.org/10.1007/978-3-030-62419-4_21.

[104] Leinberger, M., Seifer, P., Schon, C., Lämmel, R., Staab, S., 2019. Type checking program code using shacl, in: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham. pp. 399–417. doi:https://doi.org/10.1007/978-3-030-30793-6_23.

[105] Lieber, S., 2022. Assessing, creating and using knowledge graph restrictions. Ph.D. thesis.

[106] Lieber, S., De Meester, B., Heyvaert, P., Brückmann, F., Wambacq, R., Mannens, E., Verborgh, R., Dimou, A., 2021. Visual notations for viewing rdf constraints with unshacled. Semantic Web 13, 1–36. doi:10.3233/SW-210450.

[107] Lihui, C., Wai Lian, C., 2005. Using web structure and summarisation techniques for web content mining. Information Processing & Management 41, 1225–1242. URL: https://www.sciencedirect.com/science/article/pii/S0306457304001049, doi:https://doi.org/10.1016/j.ipm.2004.08.003.

[108] Lodemann, M., Marnau, R., Luttenberger, N., 2012. Using semantic constraints for data verification in an open world, in: Lukose, D., Ahmad, A.R., Suliman, A. (Eds.), Knowledge Technology, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 206–215.

[109] Malburg, L., Klein, P., Bergmann, R., 2020. Using semantic web services for ai-based research in industry 4.0. CoRR abs/2007.03580. URL: https://arxiv.org/abs/2007.03580, arXiv:2007.03580.

[110] Mao, X., Sun, H., Zhu, X., Li, J., 2022. Financial fraud detection using the related-party transaction knowledge graph. Procedia Computer Science 199, 733–740. URL: https://www.sciencedirect.com/science/article/pii/S1877050922000928, doi:https://doi.org/10.1016/j.procs.2022.01.091. the 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19.

[111] Marchetti, A., Ronzano, F., Tesconi, M., Minutoli, S., 2008. Formalizing knowledge by ontologies: Owl and kif .

[112] Martin, L., Henrich, A., 2022. Specification and validation of quality criteria for git repositories using rdf and shacl, in: Reuss, P., Eisenstadt, V., Schönborn, J., Schäfer, J. (Eds.), Proceedings of the LWDA 2022 Workshops: FGWM, FGKD, and FGDB, Oktober 5-7th, 2022, RWTH Aachen, Hildesheim (Germany). pp. 124–135. URL: https://ceur-ws.org/

Vol-3341/WM-LWDA_2022_CRC_1149.pdf. https://fis.uni-bamberg.de/handle/uniba/89863.

[113] Martin, P.A., 2018. Evaluating ontology completeness via sparql and relations-between-classes based constraints, in: 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC), pp. 255–263. doi:10.1109/QUATIC.2018.00045.

[114] Martínez-Costa, C., Schulz, S., 2017. Validating ehr clinical models using ontology patterns. Journal of Biomedical Informatics 76, 124–137. URL: https://www.sciencedirect.com/science/article/pii/S1532046417302381, doi:https://doi.org/10.1016/j.jbi.2017.11.001.

[115] Melo, A., Paulheim, H., 2017. Detection of relation assertion errors in knowledge graphs, in: Proceedings of the 9th Knowledge Capture Conference, Association for Computing Machinery, New York, NY, USA. URL: https://doi.org/10.1145/3148011.3148033, doi:10.1145/3148011.3148033.

[116] Mihindukulasooriya, N., Rashid, M.R.A., Rizzo, G., García-Castro, R., Corcho, O., Torchiano, M., 2018. Rdf shape induction using knowledge base profiling, in: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, Association for Computing Machinery, New York, NY, USA. p. 1952–1959. URL: https://doi.org/10.1145/3167132.3167341, doi:10.1145/3167132.3167341.

[117] Muñoz, E., 2016. On learnability of constraints from rdf data, in: Sack, H., Blomqvist, E., d'Aquin, M., Ghidini, C., Ponzetto, S.P., Lange, C. (Eds.), The Semantic Web. Latest Advances and New Domains, Springer International Publishing, Cham. pp. 834–844. doi:https://doi.org/10.1007/978-3-319-34129-3_52.

[118] Muñoz, E., Nickles, M., 2017. Mining cardinalities from knowledge bases, in: Benslimane, D., Damiani, E., Grosky, W.I., Hameurlain, A., Sheth, A., Wagner, R.R. (Eds.), Database and Expert Systems Applications, Springer International Publishing, Cham. pp. 447–462. doi:https://doi.org/10.1007/978-3-319-64468-4_34.

[119] Muñoz, E., Nickles, M., 2018. Statistical Relation Cardinality Bounds in Knowledge Bases. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 67–97. URL: https://doi.org/10.1007/978-3-662-58415-6_3, doi:10.1007/978-3-662-58415-6_3.

[120] Nguyen, T.H.H., Le-Thanh, N., 2014. Ensuring the semantic correctness of workflow processes: an ontological approach, in: Proceedings of the 10th International Conference on Knowledge Engineering and Software Engineering - Volume 1289, CEUR-WS.org, Aachen, DEU. p. 30–41. doi:10.5555/3053689.3053695.

[121] Nicolas Prat, I.C.W., Akoka, J., 2015. A taxonomy of evaluation methods for information systems artifacts. Journal of Management Information Systems 32, 229–267. doi:https://doi.org/10.1080/07421222.2015.1099390.

[122] Nikolaou, C., Cuenca Grau, B., Kostylev, E.V., Kaminski, M., Horrocks, I., 2019. Satisfaction and implication of integrity constraints in ontology-based data access, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization. pp. 1829–1835. URL: https://doi.org/10.24963/ijcai.2019/253, doi:10.24963/ijcai.2019/253.

[123] Nosek, B., Alter, G., Banks, G., Borsboom, D., Bowman, S., Breckler, S., Buck, S., Chambers, C., Chin, G., Christensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D., Hesse, B., Humphreys, M., Yarkoni, T., 2015. Promoting an open research culture. Science (New York, N.Y.) 348, 1422–5. doi:10.1126/science.aab2374.

[124] Omran, P., Taylor, K., Rodríguez Méndez, S., Haller, A., 2023. Learning shacl shapes from knowledge graphs. Semantic Web 14, 1–21. doi:10.3233/SW-223063.

[125] Pandit, H.J., O'Sullivan, D., Lewis, D., 2019. Test-driven approach towards gdpr compliance, in: Acosta, M., Cudré-Mauroux, P., Maleshkova, M., Pellegrini, T., Sack, H., Sure-Vetter, Y. (Eds.), Semantic Systems. The Power of AI and Knowledge Graphs, Springer International Publishing, Cham. pp. 19–33. doi:https://doi.org/10.1007/978-3-030-33220-4_2.

[126] Pareti, P., Konstantinidis, G., 2022. A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs. Springer International Publishing, Cham. pp. 115–144. URL: https://doi.org/10.1007/978-3-030-95481-9_6, doi:10.1007/978-3-030-95481-9_6.

[127] Pareti, P., Konstantinidis, G., Mogavero, F., 2022. Satisfiability and containment of recursive shacl. Journal of Web Semantics 74, 100721. URL: https://www.sciencedirect.com/science/article/pii/S1570826822000130, doi:https://doi.org/10.1016/j.websem.2022.100721.

[128] Pareti, P., Konstantinidis, G., Mogavero, F., Norman, T.J., 2020. Shacl satisfiability and containment, in: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham. pp. 474–493. doi:https://doi.org/10.48550/arXiv.2009.09806.

[129] Pareti, P., Konstantinidis, G., Norman, T.J., Şensoy, M., 2019. Shacl constraints with inference rules, in: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham. pp. 539–557. doi:https://doi.org/10.1007/978-3-030-30793-6_31.

[130] Patel, A., Jain, S., 2018. Formalisms of representing knowledge. Procedia Computer Science 125, 542–549. URL: https://www.sciencedirect.com/science/article/pii/S187705091732834X, doi:https://doi.org/10.1016/j.procs.2017.12.070. the 6th International Conference on Smart Computing and Communications.

[131] Patel-Schneider, P.F., 2015. Using description logics for rdf constraint checking and closed-world recognition, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press. p. 247–253. doi:10.5555/2887007.2887042.

[132] Patel-Schneider, P.F., Franconi, E., 2012. Ontology constraints in incomplete and complete data, in: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (Eds.), The Semantic Web – ISWC 2012, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 444–459. doi:https://doi.org/10.1007/978-3-642-35176-1_28.

[133] Pavlic, M., Meštrović, A., Jakupovic, A., 2013. Graph-based formalisms for knowledge representation.

[134] Petersen, K., Gencel, C., 2013. Worldviews, research methods, and their relationship to validity in empirical software engineering research. doi:10.1109/IWSM-Mensura.2013.22.

[135] Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology 64, 1–18. URL: https://www.sciencedirect.com/science/article/pii/S0950584915000646, doi:https://doi.org/10.1016/j.infsof.2015.03.007.

[136] Potoniec, J., Jakubowski, P., Ławrynowicz, A., 2017. Swift linked data miner: Mining owl 2 el class expressions directly from online rdf datasets. Journal of Web Semantics 46-47, 31–50. URL: https://www.sciencedirect.com/science/article/pii/S157082681730032X, doi:https://doi.org/10.1016/j.websem.2017.08.001.

[137] Prud'hommeaux, E., Labra Gayo, J.E., Solbrig, H., 2014. Shape expressions: an rdf validation and transformation language, in: Proceedings of the 10th International Conference on Semantic Systems, Association for Computing Machinery, New York, NY, USA. p. 32–40. URL: https://doi.org/10.1145/2660517.2660523, doi:10.1145/2660517.2660523.

[138] Publio, G.C., 2018. Shark: A test-driven framework for design and evolution of ontologies, in: Gangemi, A., Gentile, A.L., Nuzzolese, A.G., Rudolph, S., Maleshkova, M., Paulheim, H., Pan, J.Z., Alam, M. (Eds.), The Semantic Web: ESWC 2018 Satellite Events, Springer International Publishing, Cham. pp. 314–324. doi:https://doi.org/10.1007/978-3-319-98192-5_50.

[139] Rabbani, K., Lissandrini, M., Hose, K., 2023. Extraction of validating shapes from very large knowledge graphs. Proc. VLDB Endow. 16, 1023–1032. URL: https://doi.org/10.14778/3579075.3579078, doi:10.14778/3579075.3579078.

[140] Rashid, M.R.A., Rizzo, G., Torchiano, M., Mihindukulasooriya, N., Corcho, O., García-Castro, R., 2019. Completeness and consistency analysis for evolving knowledge bases. Journal of Web Semantics 54, 48–71. URL: https://www.sciencedirect.com/science/article/pii/S1570826818300623, doi:https://doi.org/10.1016/j.websem.2018.11.004. managing the Evolution and Preservation of the Data Web.

[141] Rettinger, A., Nickles, M., Tresp, V., 2009. Statistical relational learning with formal ontologies, in: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 286–301. doi:https://doi.org/10.1007/978-3-642-04174-7_19.

[142] Robaldo, L., 2021. Towards compliance checking in reified i/o logic via shacl, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, Association for Computing Machinery, New York, NY, USA. p. 215–219. URL: https://doi.org/10.1145/3462757.3466065, doi:10.1145/3462757.3466065.

[143] Robinson, I., Webber, J., Eifrem, E., 2015. Graph Databases: New Opportunities for Connected Data. 2nd ed., O'Reilly Media, Inc. doi:10.5555/2846367.

[144] Sagi, G., 2014. Formality in logic: From logical terms to semantic constraints. Logique et Analyse 57, 259–276. doi:10.2143/LEA.227.0.3053506.

[145] Saleh, D.R., Kartika, Y.A., Akbar, Z., Krisnadhi, A.A., Fatriasari, W., 2023. On generating shacl shapes from collective collection of plant trait data, in: Proceedings of

the 2022 International Conference on Computer, Control, Informatics and Its Applications, Association for Computing Machinery, New York, NY, USA. p. 326–330. URL: https://doi.org/10.1145/3575882.3575945, doi:10.1145/3575882.3575945.

[146] Samuel, J., 2021. Shexstatements: Simplifying shape expressions for wikidata, in: Companion Proceedings of the Web Conference 2021, Association for Computing Machinery, New York, NY, USA. p. 610–615. URL: https://doi.org/10.1145/3442442.3452349, doi:10.1145/3442442.3452349.

[147] Savković, O., Kharlamov, E., Lamparter, S., 2019. Validation of shacl constraints over kgs with owl 2 ql ontologies via rewriting, in: Hitzler, P., Fernández, M., Janowicz, K., Zaveri, A., Gray, A.J., Lopez, V., Haller, A., Hammar, K. (Eds.), The Semantic Web, Springer International Publishing, Cham. pp. 314–329. doi:https://doi.org/10.1007/978-3-030-21348-0_21.

[148] Schaffenrath, R., Proksch, D., Kopp, M., Albasini, I., Panasiuk, O., Fensel, A., 2020. Benchmark for performance evaluation of shacl implementations in graph databases, in: Gutiérrez-Basulto, V., Kliegr, T., Soylu, A., Giese, M., Roman, D. (Eds.), Rules and Reasoning, Springer International Publishing, Cham. pp. 82–96. doi:https://doi.org/10.1007/978-3-030-57977-7_6.

[149] Schenner, G., Bischof, S., Comploi-Taupe, R., Steyskal, S., 2018. Integrating semantic web technologies and asp for product configuration. URL: https://api.semanticscholar.org/CorpusID:52883313.

[150] Schmidt, M., Lausen, G., 2013. Pleasantly consuming linked data with rdf data descriptions. URL: https://arxiv.org/abs/1307.3419, doi:https://doi.org/10.48550/arXiv.1307.341, arXiv:1307.3419.

[151] Seifer, P., Lämmel, R., Staab, S., 2021. Progs: Property graph shapes language, in: Hotho, A., Blomqvist, E., Dietze, S., Fokoue, A., Ding, Y., Barnaghi, P., Haller, A., Dragoni, M., Alani, H. (Eds.), The Semantic Web – ISWC 2021, Springer International Publishing, Cham. pp. 392–409. doi:https://doi.org/10.1007/978-3-030-88361-4_23.

[152] Senthilvel, M., Beetz, J., 2020. A visual programming approach for validating linked building data. doi:10.14279/DEPOSITONCE-9977.

[153] Shu, Y., 2018. A practical approach to modelling and validating integrity constraints in the semantic web. Knowledge-Based Systems 153, 29–39. URL: https://www.sciencedirect.com/science/article/pii/S0950705118301862, doi:https://doi.org/10.1016/j.knosys.2018.04.021.

[154] Sirin, E., Tao, J., 2009. Towards integrity constraints in owl, in: Proceedings of the 6th International Conference on OWL: Experiences and Directions - Volume 529, CEUR-WS.org, Aachen, DEU. p. 79–88. doi:10.5555/2890046.2890055.

[155] Solbrig, H.R., Prud'hommeaux, E., Grieve, G., McKenzie, L., Mandel, J.C., Sharma, D.K., Jiang, G., 2017. Modeling and validating hl7 fhir profiles using semantic web shape expressions (shex). Journal of Biomedical Informatics 67, 90–100. URL: https://www.sciencedirect.com/science/article/pii/S1532046417300345, doi:https://doi.org/10.1016/j.jbi.2017.02.009.

[156] Spahiu, B., Maurino, A., Palmonari, M., 2018. Towards improving the quality of knowledge graphs with data-driven ontology patterns and shacl, in: WOP@ISWC. doi:10.3233/978-1-61499-894-5-103.

[157] Stavropoulos, T.G., Meditskos, G., Lazarou, I., Mpaltadoros, L., Papagiannopoulos, S., Tsolaki, M., Kompatsiaris, I., 2021. Detection of health-related events and behaviours from wearable sensor lifestyle data using symbolic intelligence: A proof-of-concept application in the care of multiple sclerosis. Sensors 21. URL: https://www.mdpi.com/1424-8220/21/18/6230, doi:10.3390/s21186230.

[158] Staworko, S., Boneva, I., Labra Gayo, J., Hym, S., Prud'hommeaux, E., Solbrig, H., 2015. Complexity and expressiveness of shex for rdf doi:10.4230/LIPIcs.ICDT.2015.195.

[159] Staworko, S., Wieczorek, P., 2019. Containment of shape expression schemas for rdf, in: Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Association for Computing Machinery, New York, NY, USA. p. 303–319. URL: https://doi.org/10.1145/3294052.3319687, doi:10.1145/3294052.3319687.

[160] Stolk, S., McGlinn, K., 2020. Validation of ifcowl datasets using shacl, in: LDAC. URL: https://api.semanticscholar.org/CorpusID:221461290.

[161] Studer, R., Grimm, S., Abecker, A., 2007. Semantic Web Services: Concepts, Technologies, and Applications. doi:10.1007/3-540-70894-4.

[162] Sy, M.F., Roman, B., Kerrien, S., Montero Mendez, D., Genet, H., Wajerowicz, W., Dupont, M., Lavriushev, I., Machon, J.A., Pirman, K., Neela Mana, D., Stafeeva, N., Kaufmann, A.K., Lu, H., Lurie, J., Fonta, P.A., Garcia Rojas Martinez, A., Ulbrich, A., Lindqvist, C., Jimenez, s., Rotenberg, D., Markram, H., Hill, S., 2023. Blue brain nexus: An open, secure, scalable system for knowledge graph management and data-driven science. URL: https://infoscience.epfl.ch/handle/20.500.14299/197266, doi:10.3233/SW-222974.

[163] Taghzouti, Y., Vachtsevanou, D., Mayer, S., Ciortea, A., 2022. A step toward semantic content negotiation.

[164] Tao, J., Sirin, E., Bao, J., McGuinness, D.L., 2010. Integrity constraints in owl, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI Press. p. 1443–1448. doi:10.5555/2898607.2898837.

[165] Thapa, R.B., Giese, M., 2021. A source-to-target constraint rewriting for direct mapping, in: Hotho, A., Blomqvist, E., Dietze, S., Fokoue, A., Ding, Y., Barnaghi, P., Haller, A., Dragoni, M., Alani, H. (Eds.), The Semantic Web – ISWC 2021, Springer International Publishing, Cham. pp. 21–38. doi:https://doi.org/10.1007/978-3-030-88361-4_2.

[166] Thapa, R.B., Giese, M., 2022. Mapping relational database constraints to shacl, in: Sattler, U., Hogan, A., Keet, M., Presutti, V., Almeida, J.P.A., Takeda, H., Monnin, P., Pirrò, G., d'Amato, C. (Eds.), The Semantic Web – ISWC 2022, Springer International Publishing, Cham. pp. 214–230. doi:https://doi.org/10.1007/978-3-031-19433-7_13.

[167] Thornton, K., Solbrig, H., Stupp, G.S., Labra Gayo, J.E., Mietchen, D., Prud'hommeaux, E., Waagmeester, A., 2019. Using shape expressions (shex) to share rdf data models and to guide curation with rigorous validation, in: Hitzler, P., Fernández, M., Janowicz, K.,

Zaveri, A., Gray, A.J., Lopez, V., Haller, A., Hammar, K. (Eds.), The Semantic Web, Springer International Publishing, Cham. pp. 606–620. doi:https://doi.org/10.1007/978-3-030-21348-0_39.

[168] Thuluva, A.S., Anicic, D., Rudolph, S., 2018. Iot semantic interoperability with device description shapes, in: Gangemi, A., Gentile, A.L., Nuzzolese, A.G., Rudolph, S., Maleshkova, M., Paulheim, H., Pan, J.Z., Alam, M. (Eds.), The Semantic Web: ESWC 2018 Satellite Events, Springer International Publishing, Cham. pp. 409–422.

[169] Tomaszuk, D., 2017. Rdf validation: A brief survey, pp. 344–355. doi:10.1007/978-3-319-58274-0_28.

[170] Turki, H., Jemielniak, D., Taieb, M.A.H., Gayo, J.E.L., Aouicha, M.B., Banat, M., Shafee, T.M.A., Prud'hommeaux, E., Lubiana, T., Das, D., Mietchen, D., 2022. Using logical constraints to validate statistical information about disease outbreaks in collaborative knowledge graphs: the case of covid-19 epidemiology in wikidata. PeerJ Computer Science 8. URL: https://api.semanticscholar.org/CorpusID:252636723, doi:10.7717/peerj-cs.1085.

[171] Uschold, M., Grüninger, M., 1996. Ontologies: Principles, methods and applications. The Knowledge Engineering Review 11. doi:10.1017/S0269888900007797.

[172] Vrandečić, D., Pintscher, L., Krötzsch, M., 2023. Wikidata: The making of, in: Companion Proceedings of the ACM Web Conference 2023, Association for Computing Machinery, New York, NY, USA. p. 615–624. URL: https://doi.org/10.1145/3543873.3585579, doi:10.1145/3543873.3585579.

[173] Wattiau, I., Bouzeghoub, M., 2006. Constraint confrontation: An important step in view integration. pp. 507–523. doi:10.1007/BFb0035151.

[174] Werbrouck, J., Taelman, R., Verborgh, R., Pauwels, P., Beetz, J., Mannens, E., 2020. Pattern-based access control in a decentralised collaboration environment, in: Proceedings of LDAC2020 - 8th Linked Data in Architecture and Construction Workshop. URL: https://www.researchgate.net/profile/Jeroen_Werbrouck/publication/342354922_Pattern-based_access_control_in_a_decentralised_collaboration_environment/links/5ef085dda6fdcc73be944024/Pattern-based-access-control-in-a-decentralised-collaboration-environment.pdf.

[175] Wilson, G., Aruliah, D.A., Brown, C.T., Chue Hong, N.P., Davis, M., Guy, R.T., Haddock, S.H.D., Huff, K.D., Mitchell, I.M., Plumbley, M.D., Waugh, B., White, E.P., Wilson, P., 2014. Best practices for scientific computing. PLoS Biology 12, e1001745. URL: http://dx.doi.org/10.1371/journal.pbio.1001745, doi:10.1371/journal.pbio.1001745.

[176] Wright, J., Rodríguez Méndez, S.J., Haller, A., Taylor, K., Omran, P.G., 2020. Schímatos: A shacl-based web-form generator for knowledge graph editing, in: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham. pp. 65–80. doi:https://doi.org/10.1007/978-3-030-62466-8_5.

[177] Xu, L., Yuan, Q., Wu, J., Liu, C., 2009. Ontology-based web service robustness test generation, in: 2009 11th IEEE International Symposium on Web Systems Evolution, pp. 59–68. doi:10.1109/WSE.2009.5632118.

[178] Zemmouchi-Ghomari, L., 2021. How industry 4.0 can benefit from semantic web technologies and artefacts. International Journal of Software Science and Computational Intelligence 13, 64–74. doi:10.4018/IJSSCI.2021100105.

[179] Zhao, X., Li, F., Yang, H., 2023. Rdfs(c) schema inconsistency checking based on a key instance and its query rewriting. IEEE Access 11, 6122–6132. doi:10.1109/ACCESS.2023.3234084.

[180] Zhu, R., Shimizu, C., Stephen, S., Zhou, L., Cai, L., Mai, G., Janowicz, K., Schildhauer, M., Hitzler, P., 2022. Sosa-shacl: Shapes constraint for the sensor, observation, sample, and actuator ontology, in: Proceedings of the 10th International Joint Conference on Knowledge Graphs, Association for Computing Machinery, New York, NY, USA. p. 99–107. URL: https://doi.org/10.1145/3502223.3502235, doi:10.1145/3502223.3502235.