UNIVERSIDAD DE LA REPÚBLICA URUGUAY

FACULTAD DE INGENIERÍA UDELAR

Design and foundations of ontologies with meta-modelling

Edelweis Rohrer Errecarte

Tesis de Doctorado presentada a la Facultad de Ingeniería de la

Universidad de la República

en cumplimiento parcial de los requerimientos para la obtención del título de Doctor en Informática.

Tutores

Regina Motz
Paula Severi

Tribunal

Claudio Gutiérrez
Alexandre Rademaker
Alvaro Tasistro
Anni-Yasmin Turhan
Oscar Corcho

Montevideo, Uruguay
Julio de 2022

Abstract

Ontologies are broadly used and proved modelling artifacts to conceptualize a domain. In particular the W3C standard ontology language OWL, based on description logics, allows the ontology engineer to formally represent a domain as a set of assertions about concepts, individuals and roles. Nowadays, complex applications leads to combine autonomously built ontologies into ontology networks by relating them through different kind of relations. Some relations, such as the mapping of two concepts from different ontologies, can be expressed by the standard ontology language OWL, i.e. by the description logics behind it. However, there are other kind of relations that are not soundly represented by OWL, such as the meta-modelling relation. The meta-modelling relation has to do with the modelling of the same real object with different abstraction levels, e.g. as a concept in one ontology and as an individual in another ontology.

Even though there are a set of approaches that extend description logics to deal with meta-modelling, they do not solve relevant requirements of some real scenarios. The present thesis work introduces an extension to the description logic $\mathcal{SHIQ}$ which provides a flexible syntax and a strong semantics, and moreover ensures the well-foundedness of the interpretation domain. This approach is different from existing meta-modelling approaches either in the syntax or in the semantics (or both), and moreover ensures the well-foundedness of the domain which is an original contribution from the theoretical point of view. The meta-modelling extension of $\mathcal{SHIQ}$ introduced in the present work is justified by a detailed description of a set of real case studies, with an analysis of the benefits of the new approach to solve some relevant requirements. Finally, the present work addresses the methodological issue by introducing a design pattern to help the ontology engineer in the use of the proposed meta-modelling approach.

Contents

Chapter 1

Introduction

Increasingly, knowledge bases are adopted as a way of integrating, organizing and giving meaning to sets of information, providing vocabularies which describe both general and specific domains of particular organizations.

Within the different kind of knowledge bases, ontologies are broadly used and proved modelling artifacts to conceptualize a domain. They describe vocabularies as concepts (or entity types), instances (or individuals) and roles (or relations), and also assertions about them. In particular the W3C standard ontology language OWL, based on description logics, allows the ontology engineer to formally represent a set of assertions for a given vocabulary. In this context, the motivation of the present work has two main complementaries dimensions:

- the "technical" motivation of exploring different ways of integrating either ontologies of different domains or ontologies that represent different perspectives of the same domain, and
- the "practical" motivation of contributing to the integration between academic and industry environments. In spite of the increasing popularity of ontologies, there are still areas of the software industry for which the benefit of ontologies are not exploited, in particular in Uruguay.

Regarding the technical dimension, the need of combining vocabularies in several case studies leads to the integration of a set of ontologies in an *ontology network*, with the identification of different kind of relations among them. Some relations can be expressed by the standard ontology language OWL. For instance, the mapping relation expresses that a concept A in a given ontology represents the same set of objects than a concept B of another ontology. This relation is able to be expressed in description logics by the statement $A \equiv B$. However, there are other kind of relations that are not fully enabled by OWL, such as the meta-modelling relation. The meta-modelling relation has to do with modelling different abstraction levels of the knowledge. For instance, given a real world object, it can be modelled as a concept or as an individual, depending on the perspective or granularity of the conceptualization. Figure 1.1 shows an example of an ontology network that combines three ontologies delimited by dotted lines, with concepts represented by ovals, individuals represented by bullets and roles (usually binary relations) represented by edges. There is an ontology about protected areas, with concepts *ProtectedArea* and *NonRenewableResource*, another one about aquifers, with concepts *River* and

Lake and the a third ontology about policies for natural resources, with concepts *Policies* and *NaturalResource*.

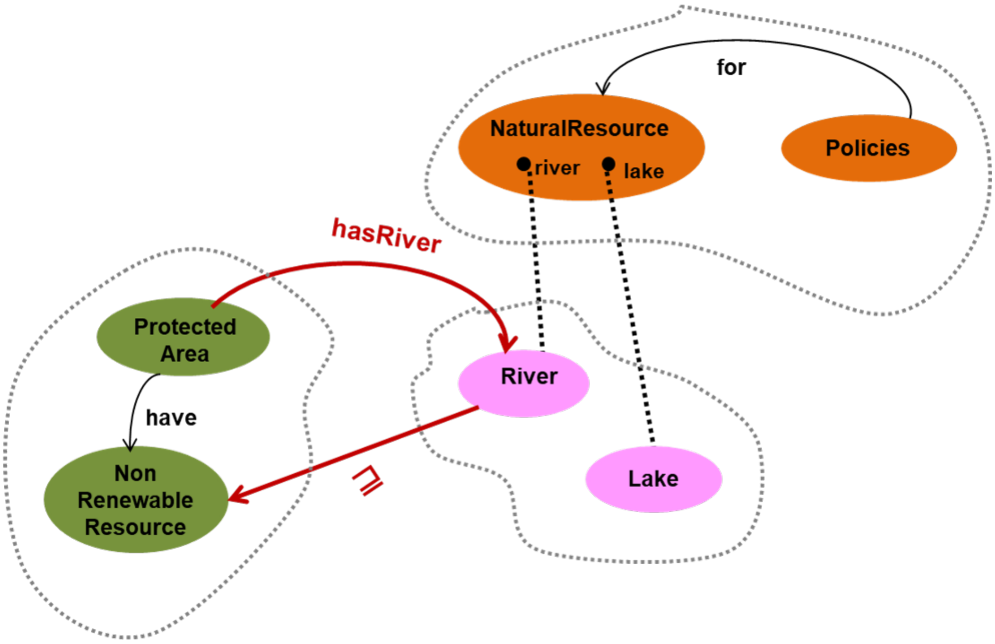

Figure 1.1: An example of ontology network

The relation between concepts *River* and *NonRenewable resource*, represented by an arrow labelled with $\sqsubseteq$ and declared by the statement $River \sqsubseteq NonRenewableResource$ in description logics, is a mapping relation which expresses that the set of rivers is a subset of the non-renewable resources. OWL is also able to declare that *hasRiver* is a relation (represented by an arrow) from instances of the concept *ProtectedArea* to instances of the concept *River*. However, there is no sound OWL constructor to express the relations given by dotted lines between instances *river* and *lake* of the concept *NaturalResource* and concepts *River* and *Lake*. The semantics of these relations is about that individuals *river* and *lake* are the same real objects than corresponding concepts *River* and *Lake*, i.e. *NaturalResource* has instances that are also treated as concepts[1]. This equality between an individual and a concept is called a *meta-modelling* relation. Actually, there is a weak OWL constructor that provides a form of meta-modelling where the same name is declared as an individual and as a concept but at the moment of checking consistency, the reasoner[2] treats them as if they were different objects. The present work deepens in the definition of a syntax, semantics and a reasoning algorithm to extend description logics with meta-modelling, by ensuring the absence of contradictions in the (big) ontology that results of the implementation of an ontology network from a set of ontologies and the relations that connect them. Basically, the reasons for addressing the *meta-modelling* relation are the ones described below.

- There is no sound support of OWL for this kind of relation, specifically, the consistency of a knowledge base with meta-modelling is not ensured.

[1]From a philosophical point of view an instance should not be instantiable, however meta-modelling is intended precisely to represent a concept as an instance of another concept

[2]A reasoner is a tool that can perform many tasks, amongst them checking consistency and entailing non explicit relations.

- Even though there is some related work about meta-modelling, existing approaches that extend description logics do not provide adequate solutions to a set of real scenarios analyzed in the present work. The application of existing meta-modelling approaches to these domains shows some drawbacks such as a complex syntax or a weak semantics which is not able to detect relevant inconsistencies that arise from meta-modelling.

A set of real case studies that were analyzed shows on the one hand, the need of expressing the meta-modelling correspondence between individuals and concepts, and on the other hand the requirement of ensuring that some relations between individuals with meta-modelling must be translated to relations between the corresponding concepts. With the aim of allowing an explicit representation of these requirements, the meta-modelling approach introduced in the present work extends the syntax of description logics with new statements to express meta-modelling. Moreover, for checking consistency, it also extends the tableau algorithm[3] by adding new rules to deal with the new statements. It is important to point out that this approach of adding new rules to the reasoning algorithm is a bit different from existing meta-modelling approaches (described in Chapter 2) that basically follow two main mechanisms: (i) the codification of meta-modelling behind existing constructs of description logics (e.g. by adding fictitious roles to link individuals and concepts) without modifying existing reasoners, or (ii) either extending or not the syntax[4], the reduction of the new meta-modelling description logics to the description logics being extended. The first alternative was discarded due to its low expressiveness since it does not explicit the meta-modelling relation. The second alternative has the advantage of reusing the tableau algorithm for OWL without changing it. Despite the approach of the present work is different since it extends the tableau algorithm, it goes in the same direction in the sense that existing rules are not changed at all. This approach adds new independent rules each time a more expressive logic is needed.

Regarding the practical motivation, many implemented applications do not benefit from the conceptual level of expressivity provided by ontologies, and instead, they have a lot of business rules hidden in the code. Besides being a powerful artifact for conceptual design, existing implementations of reasoners enable the automatic validation of the ontology constraints on the final application. Even though maybe semantic technologies are not mature enough to be used in the software industry, in many organizations there is not even a discussion about using or not using ontologies. Taking into account different factors such as the volume of information, and the priorization of quality attributes such us performance, soundness or scalability, different fragments of OWL could be more or less suitable for different applications. In this sense, a set of real scenarios from different domains such as geography, health, recommender systems, education and accounting (presented in Chapter 3) were identified. These scenarios have a set of common requirements for which ontologies and in particular meta-modelling provides a way of explicitly conceptualize them. In particular, some of these domains have requirements associated to different levels of users, such as experts and operators, who have different views of the

[3]Most popular reasoners implement tableau algorithms which define a set of rules to deal with the constructs of description logics [Hitzler09].

[4]When meta-modelling is represented within a single ontology, the same name can be used as individual and as a concept, so the syntax does not need to be extended

same business. Then, in the present work existing meta-modelling approaches are analyzed for this set of real case studies, leading to the conclusion that they do not solve the most relevant identified requirements. As a result, this work introduces a new meta-modelling approach to solve requirements at different knowledge levels and in particular for different views or perspectives of users. Finally, to provide the ontology engineer a useful mechanism to discern how and in what scenarios to apply the introduced meta-modelling approach, the present work presents a design pattern for conceptualizing ontologies with meta-modelling. From the analysis of the related work about ontology engineering presented in Chapter 2, it arises that it is the first design pattern that helps the ontology engineer in conceptualizing domains for which there are requirements at different knowledge levels which represent different perspectives of the same real objects.

The main contribution of the present work is in the following directions.

- This work extends the description logic $\mathcal{SHIQ}$[5] with new statements to deal with meta-modelling, specifically to represent different perspectives[6] of the same domain.
- This work shows the usefulness of the meta-modelling approach for different case studies.
- This work introduces a design pattern to help in the use of the proposed meta-modelling approach.

The underlying spirit of the work is the approaching of theoretical foundations on meta-modelling to practical scenarios, which cannot be achieved by limited examples not embedded in real contexts. The remainder of the document is organized in chapters, which are briefly sketched below together with a set of publications that are part of the thesis work. The author of this work has played an active role in all the publications mentioned below. In particular, she has come up with the idea of all design aspects related to the application of meta-modeling to case studies, as well as the identification of the extensions and restrictions to description logics that were required to represent such scenarios with sufficient expressiveness. All the publications are original works that were accepted in good quality lectures and journals, or in workshops where interesting discussions arise. For example, KR conferences have a rank A^* (source: CORE), I3E conferences have a rank $B3$ (source: Qualis) and AMW workshops have a rank $B4$ (source: Qualis).

Chapter 2. Background and related work. With the aim of making the work self-contained this chapter starts introducing the foundations on ontologies and description logics. Moreover, notions of ontology network and module are introduced together with different approaches about ontology modularization which solve different kind of interactions between (modules of) ontologies. Some of the presented approaches are nowadays rather obsoletes but anyway are mentioned in the present work due to two main reasons. On the one hand, they are a first step to the ontology networks semantics adopted later and on the other hand, they inspired some of

[5] $\mathcal{SHIQ}$ is an expressive fragment of $\mathcal{SROIQ}$, the description logic underlying OWL.

[6] The term “perspective” is defined in Chapter 3,in the context of the present work it intuitively means “view” or “perception”

the ontology relationships identified in case studies presented in Chapter 3. Next, existing description logics meta-modelling approaches are described, due to they are the work straight related to the main contribution of the present work. The most relevant differences among the semantics of the presented approaches are addressed and compared regarding two main aspects: (i) the structure of the interpretation domain and (ii) the properties "intensional regularity" and "extensionality", which make semantics stronger or weaker, affecting the detection of inconsistencies in ontologies with meta-modelling. Finally, the chapter gives an overview of the main metodological approaches about ontology engineering, both for ontologies without and with meta-modelling, and in particular about ontology design patterns.

Chapter 3. Motivating case studies. This chapter presents five different case studies about geographic, health, education, accounting and recommender systems domains, which have in common the need of representing different views or perceptions of a set of real objects. These views are called "perspectives" and are in general associated to a user role that perceives each object with a given granularity (as an individual or as a concept). Besides defining the notion of perspective, as the presented domains are mostly conceptualized with more than one ontology, different relations (including meta-modelling) between networked ontologies are identified and intuitively defined. Each case study is then conceptualized by using the meta-modelling relation as well as other relations if necessary. Moreover, the expressivity of the meta-modelling approaches presented in Chapter 2 is analized regarding the capability to represent each domain by ensuring its consistency. It is showed that to ensure the consistency of the addressed domains, two additional conditions[7] must hold: (i) the different perspectives of the domain must be consistent to each other, and (ii) the domain of interpretation must be well-founded[8]. Finally, to better delimit the scope of the meta-modelling addressed in the present work, a different scenario of use of meta-modelling (to represent "terms" that denote objects) is briefly analyzed.

Below the list of publications that support the content of this chapter is briefly described.

E. Rohrer, R. Motz, A. Díaz. *Ontology-Based Process for Recommending Health WebSites.* Software Services for e-World - 10th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society (I3E), 2010. Describes a recommender system of contents for the health domain.

E. Rohrer, R. Motz, A. Díaz. *Modeling a web site quality-based recommendation system.* The 12th International Conference on Information Integration and Web-based Applications and Services (iiWAS), 2010. Describes a generic recommender system based on the quality of contents.

E. Rohrer, R. Motz, A. Díaz. *Modeling and Use of an Ontology Network for Website Recommendation Systems.* On the Move to Meaningful Internet Systems (OTM), 2010. Describes a health recommender system showing the interaction of the popu-

[7]These conditions must be added to the consistency checks made by OWL reasoners

[8]Intuitively, in a well-founded domain an object cannot belong to itself

lated networked ontologies.

A. Díaz, R. Motz, E. Rohrer. *Making Ontology Relationships Explicit in a Ontology Network.* The 5th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW), 2011. Describes different kind of relations between networked ontologies.

E. Rohrer, R. Motz, A. Díaz. *Modelling a web site quality-based recommendation system.* Int. J. Web Inf. Syst., 2011. Extends the work [Rohrer10b].

A. Díaz, R. Motz, E. Rohrer, L. Tansini. *An Ontology Network for Educational Recommender Systems.* Educational Recommender Systems and Technologies: Practices and Challenges, 2012. Describes an educational recommender system based on the quality of contents and the user profile.

E. Rohrer, P. Severi, R. Motz, A. Díaz. *Metamodelling in a Ontology Network.* The 8th International Workshop on Modular Ontologies (WOMO), 2014. Provides a formal definition of different kind of relations between networked ontologies and introduces the geographic domain.

R. Motz, E. Rohrer, P. Severi. *Reasoning for* ALCQ *Extended with a Flexible Meta-Modelling Hierarchy.* Semantic Technology - 4th Joint International Conference (JIST), 2014. Introduces a conceptualization of the geographic domain with meta-modelling, as the motivating case study for the introduction of the description logics $\mathcal{ALCQM}$.

R. Motz, E. Rohrer, P. Severi. *Applying Description Logics Extended with Meta-modelling to SNOMED-CT.* The 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web (AMW), 2017. Introduces a conceptualization of the health domain with meta-modelling.

E. Rohrer, P. Severi, R. Motz. *Applying meta-modelling to an accounting application.* The XI Seminar on Ontology Research in Brazil and II Doctoral and Masters Consortium on Ontologies (ONTOBRAS), 2018. Introduces a conceptualization of the accounting domain with meta-modelling.

E. Rohrer, P. Severi, R. Motz. *Meta-Modelling Ontology Design Pattern.* Knowledge Graphs and Semantic Web - First Iberoamerican Conference (KGSWC), 2019. Introduces a conceptualization of the educational domain with meta-modelling, as the motivating case study for the introduction of a meta-modelling ontology design pattern.

Chapter 4. A Henkin meta-modelling approach. The development of Chapter 3 shows that description logics meta-modelling approaches described in Chapter 2 do not cover some relevant requirements of the scenarios presented in Chapter 3. Hence, Chapter 4 introduces a new meta-modelling approach which follows a Henkin style semantics that satisfies both "intensional regularity" and "extensionality" properties, and that defines a well-founded domain with flexible layers that interact with each other. The new approach allows the ontology engineer representing restrictions that solve the requirements of the case studies in Chapter 3. This approach is different from the ones presented in Chapter 2 in the sense that they either follow a Henkin semantics but with a domain of fixed layers, or follow a Hilog semantics that satisfy "intensional regularity" but not "extensionality". First of all, this chapter presents the description logic $\mathcal{SHIQM}$ that extends $\mathcal{SHIQ}$ by introducing a statement to represent that an individual a is the same real object than an atomic concept A. Next, to solve some requirements of educational and accounting domains, the description logic $\mathcal{SHIQM}^*$ is introduced. $\mathcal{SHIQM}^*$ extends $\mathcal{SHIQM}$ with a new statement that transfers relations between individuals at higher meta-modelling levels into restrictions on corresponding concepts in lower levels. Moreover, it is described how $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$ solve the requirements of scenarios presented in Chapter 3.

Publications mentioned below present description logics $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$.

R. Motz, E. Rohrer, P. Severi. *Reasoning for* ALCQ *Extended with a Flexible Meta-Modelling Hierarchy.* Semantic Technology - 4th Joint International Conference (JIST), 2014. Introduces the description logic $\mathcal{ALCQM}$ that extends $\mathcal{ALCQ}$ with a statement representing that an individual a corresponds to a concept A by meta-modelling. It is a previous step to the definition of the description logics $\mathcal{SHIQM}$ (since $\mathcal{ALCQ}$ is less expressive than $\mathcal{SHIQ}$).

R. Motz, E. Rohrer, P. Severi. *The description logic* SHIQ *with a flexible meta-modelling hierarchy.* J. Web Semant., 2015. Describes the syntax, semantics and reasoning algorithm of the description logics $\mathcal{SHIQM}$ and shows its usefulness for the geographic domain.

M. Martinez, E. Rohrer, P. Severi. *Complexity of the Description Logic* ALCM. Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR), 2016. Shows that complexity does not change when the description logic $\mathcal{ALC}$ is extended with statements that equate individuals to concepts by meta-modelling.

P. Severi, E. Rohrer, R. Motz. *A Description Logic for Unifying Different Points of View.* Knowledge Graphs and Semantic Web - First Iberoamerican Conference (KGSWC), 2019. Describes the syntax, semantics and reasoning algorithm of the description logic $\mathcal{SHIQM}^*$ that extends $\mathcal{SHIQM}$, by allowing the transference of relations between diferent layers that conceptualize different "perspectives" or "points of view".

Chapter 5. Meta-modelling ontology pattern. This chapter introduces an ontology design pattern that helps the ontology engineer to apply the description logics $\mathcal{SHIQM}^*$ to model a scenario with requirements at different knowledge levels, in particular different user perspectives. The design pattern that is named *meta-modelling ontology pattern* is mainly motivated by scenarios about education and accounting presented in Chapter 3. In educational and accounting domains, two kind of business rules are identified in each knowledge level: *static* and *dynamic* rules. The last ones are the main motivation of the pattern. They have the particularity that even though they are restrictions to a given level, these are defined in the immediate upper level.

The *meta-modelling ontology pattern* is presented in the publication mentioned below.

E. Rohrer, P. Severi, R. Motz. *Meta-Modelling Ontology Design Pattern.* Knowledge Graphs and Semantic Web - First Iberoamerican Conference (KGSWC), 2019. Describes the design pattern and its application to the educational domain and moreover how the pattern would enhance an implemented accounting system.

Chapter 6. Conclusions, work in progress and future work. This chapter presents a summary of the thesis work as a brief analysis of the theoretical and practical contributions, as well as a general evaluation of the work done. Afterthat the work in progress is mentioned, in particular the implementation of the main thesis results. Finally, the future work is depicted, by covering theoretical and methodological work.

Chapter 2

Background and related work

This chapter presents the background on description logics and some related work about ontology networks, meta-modelling and ontology engineering, in particular about design patterns.

The present chapter is organized as follows. In Section 2.1 some basics about ontologies and description logics are presented so that the chapter is self-contained. Section 2.2 introduces the notion of ontology network and summarizes different semantic approaches to deal with the integration of autonomously built ontologies. Section 2.3 addresses the related work about the syntax and semantics to represent meta-modelling relations, which is the main topic of the present work. Section 2.4 describes the main metodological approaches for ontology engineering, both to build single level ontologies and ontologies with meta-modelling. Finally, Section 2.5 presents some conclusions.

2.1 Ontologies and description logics

The most broadly accepted notion of *ontology* is that of Studer et. al, "an *ontology* is a formal, explicit specification of a shared conceptualization" [Studer98]. This definition adds to the known idea of *conceptualization* the conditions *formal*, *explicit* and *shared*. A *conceptualization* is an abstract view of the world which is represented for some purpose, and basically consists of a set called the *universe of discourse* and a set of *relations on that universe* [Staab09]. Ontologies have been adopted as a (broadly proved) modelling artifact to *explicitly* represent a *shared* understanding of a domain, with the definition of concepts or classes (subsets of the universe), individuals or instances (elements of the universe) and roles or properties (usually binary relations). The main benefit of ontologies is that they combine the *formal* conceptualization of a domain with the validation of its definitions and rules. In particular, the W3C standard ontology language OWL is based on *description logics*[1], which provides a *formal* semantics and enables the automatic validation of the ontology constraints [Grau08a, Baader03]. For this, there are implementations of reasoners that, given an ontology, check its consistency and draw logical inferences from the domain conceptualization [Sirin07, Motik09].

[1]Even though an ontology can be represented by other formalisms such as Knowledge Interchange Format (KIF), the present work directly addresses the description logics formalism since it underlies OWL which is an standard of W3C.

An ontology conceptualizes a domain in two ways: (i) *extensionally*, by assertions about particular elements of the universe of discourse called *individuals or instances*, and (ii) *intensionally*, by assertions about sets of elements that are *concepts and roles*. Both extensional and intensional assertions are called *axioms or statements* in description logics. The axiom $A \sqsubseteq B$ is an example of an intensional assertion which expresses that the set of elements represented by the concept A is a subset of the set represented by B. The axiom $A(a)$ is an extensional assertion and expresses that the individual a belongs to the set represented by the concept A [Staab09].

Description logics is a family of decidable fragments of the first order logic, which allow representing extensional and intensional knowledge with different expressivity. For instance, the axiom $R \sqsubseteq S$ expresses that the binary relation represented by the role R is a subset of that represented by the role S. It can be expressed in a very expressive fragment of first order logic, the description logic called $\mathcal{SHIQ}$, but not in less expressive fragments such as the description logic called $\mathcal{ALC}$. $\mathcal{ALC}$ allows assertions about concepts such as $A \sqsubseteq B$ but not about roles. Moreover, *general concepts and roles* can be defined with different expressivity. For instance, the description logic $\mathcal{ALC}$ allows declaring the general concept $A \sqcap B$ for some concept names A and B. $A \sqcap B$ is unabiguously interpreted as the intersection of two sets, the set of elements belonging to the concept A and the set of elements belonging to the concept B. However, general concepts such as $\leqslant 2R.C$, which denotes the set of objects that are related at most to two objects belonging to the (general) concept C by the role R, are allowed for description logics more expressive than $\mathcal{ALC}$. The most broadly adopted style of semantics for ontologies is the *direct model-theoretic semantics* that defines the meaning of OWL directly to description logics [2] [Hitzler09]. Description logics adheres to the *Open World Assumption*, which means that we cannot say a statement to be false because we cannot show to be true, i.e. it is assumed that a knowledge base may be incomplete [Antoniou09][Hitzler09, chapter 4].

To describe the syntax and semantics of description logics, next paragraphs introduce the description logics $\mathcal{ALC}$ (Attributive Language for Complement). A slight variation of $\mathcal{ALC}$ is described without the unique name assumption, i.e. with equalities and differences between individuals.

Syntax of description logics. Assume three pairwise disjoint sets: a set of individuals $a, b, \ldots$, a set of atomic concepts $A, B, \ldots$ (i.e. concept names) and a set of atomic roles $R, S, \ldots$ (i.e. role names). *General concepts* C, D in the description logic $\mathcal{ALC}$ are constructed as follows.

$C,\ D ::= A \mid \top \mid \bot \mid (\neg C) \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$ with A an atomic concept, C, D general concepts, R an atomic role and two predefined concepts $\top$ and $\bot$ which represent the universe of discourse and the empty set respectively.

An *ontology in the description logic* $\mathcal{ALC}$ has two groups of axioms: (i) *Tbox axioms* of the form $C \sqsubseteq D$ called *(general) concept inclusion axioms*, with C, D general concepts and (ii) *Abox axioms* of the form $C(a)$, $R(a, b)$, $a = b$, $a \neq b$, with a, b individuals, C a general concept and R en atomic role, which contain assertional konwledge about individuals. Tbox axioms contain intensional knowledge whereas

[2] There exists also the model-theoretic semantics for RDF(S) that is not addressed in the present work

Abox axioms contain extensional knowledge [Hitzler09].

Example 1. Supose an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ where the Tbox $\mathcal{T}$ and the Abox $\mathcal{A}$ are as follows [3]

$$\begin{aligned}\mathcal{T} = \ & \{Person \sqcap Country \sqsubseteq \bot, Person \sqsubseteq \exists lives.Country\}\\ \mathcal{A} = \ & \{Person(maria), Person(ana), Country(uruguay), lives(maria, uruguay),\\ & lives(ana, uruguay)\}\end{aligned}$$

The first axiom in the Tbox expresses that concepts $Person$ and $Country$ are disjoint, and the second axiom expresses that each person lives at least in one country. The Abox says that $Maria$ and Ana are persons, $Uruguay$ is a country and both $Maria$ and Ana live in $Uruguay$.

. . . .

Given a role R, the *domain* and *range* of R can be declared by Tbox axioms. Intuitively, the domain of R is the set of instances that can take values for R, and the range is the set to which the values of R belong. For the ontology of Example 1, to restrict that domain and range of the role *lives* are concepts *Person* and *Country*, axioms $\exists lives.\top \sqsubseteq Person$ and $\top \sqsubseteq \forall lives.Country$ are declared.

Semantics of description logics. The *direct model-theoretic semantics of description logics* is defined by the notion of *interpretation*. An *interpretation* has a *domain of interpretation* Δ and an *interpretation function* $\cdot^{\mathcal{I}}$ which maps every concept to a subset of Δ ($A^{\mathcal{I}} \subseteq \Delta$), every role to a subset of $\Delta \times \Delta$ ($R^{\mathcal{I}} \subseteq \Delta \times \Delta$) and every individual to an element of Δ ($a^{\mathcal{I}} \in \Delta$). Then, the semantics of general concepts and axioms for $\mathcal{ALC}$ is inductively defined as showed in Table 2.1.

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
complement	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
universal restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \text{for all } y \in \Delta^{\mathcal{I}} \text{ if } (x,y) \in R^{\mathcal{I}} \text{ then } y \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$R(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
same individuals	$a = b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
different individuals	$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

Table 2.1: Syntax and semantics of $\mathcal{ALC}$

[3] In examples, individual and role names start in lowercase whereas concept names start in uppercase.

Interpretations which capture the structure of an ontology in terms of sets are called *models*. Formally, and interpretation $\mathcal{I}$ of an $\mathcal{ALC}$ ontology $\mathcal{O}$ is a *model of* $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$, if the following holds.

- $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for each $C \sqsubseteq D$ in $\mathcal{O}$.
- $a^{\mathcal{I}} \in C^{\mathcal{I}}$ holds for each $C(a)$ in $\mathcal{O}$.
- $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ holds for each $R(a, b)$ in $\mathcal{O}$.
- $a^{\mathcal{I}} = b^{\mathcal{I}}$ holds for each $a = b$ in $\mathcal{O}$.
- $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ holds for each $a \neq b$ in $\mathcal{O}$.

An *ontology is consistent or satisfiable* if it has at least a model $\mathcal{I}$, otherwise it is inconsistent or unsatisfiable. If we add axioms $Country(x)$ and $Person(x)$ to the ontology of Example 1, the ontology would become inconsistent since there is no interpretation that satisfies axioms $Country(x)$, $Person(x)$ and $Person \sqcap Country \sqsubseteq \bot$, and then the ontology has a *contradiction*. However, an ontology can have more than one model as the following example for the ontology of Example 1 shows.

Example 2. For the ontology of Example 1 there are two models $\mathcal{I}_1$ and $\mathcal{I}_2$ as follows.

$$\begin{array}{l}
\Delta_1 = \{m, a, u\} \\
maria^{\mathcal{I}_1} = m, \quad ana^{\mathcal{I}_1} = a, \quad uruguay^{\mathcal{I}_1} = u \\
Person^{\mathcal{I}_1} = \{m, a\}, \quad Country^{\mathcal{I}_1} = \{u\} \\
lives^{\mathcal{I}_1} = \{(m, u), (a, u)\}
\end{array}$$

$$\begin{array}{l}
\Delta_2 = \{a, u\} \\
maria^{\mathcal{I}_2} = a, \quad ana^{\mathcal{I}_2} = a, \quad uruguay^{\mathcal{I}_2} = u \\
Person^{\mathcal{I}_2} = \{a\}, \quad Country^{\mathcal{I}_2} = \{u\} \\
lives^{\mathcal{I}_2} = \{(a, u)\}
\end{array}$$

The model $\mathcal{I}_1$ interprets individuals *ana* and *maria* as two different objects of the domain Δ_1 whereas $\mathcal{I}_2$ interprets them as the same object a in Δ_2. However, as there is no statement that restricts *ana* and *maria* to be different individuals, the two models capture the structure declared by axioms in Example 1.

....

Even though each model provides a possible realization of the ontology, there are some assertions that are not declared and that hold for all models of the ontology. Such assertions are the *logical consequences or inferences* of the ontology. An axiom α is a *logical consequence of an ontology* $\mathcal{O}$ if $\alpha^{\mathcal{I}}$ holds in every model $\mathcal{I}$ of $\mathcal{O}$. This notion provides a means to derive implicit knowledge as Example 3 shows.

Example 3.

$$\begin{array}{rl}
\mathcal{O} = & (\mathcal{T}, \mathcal{A}) \\
\mathcal{T} = & \{LatinCountry \sqsubseteq Country, AngloCountry \sqsubseteq Country\} \\
\mathcal{A} = & \{LatinCountry(uruguay), AngloCountry(unitedKingdom)\}
\end{array}$$

Logical consequences $Country(uruguay)$, $Country(unitedKingdom)$ are implicit knowledge not declared in $\mathcal{O}$ that hold for every model of $\mathcal{O}$.

. . . .

A set of algorithms have been proposed and implemented to check the consistency and to calculate the logical consequences of an ontology. Implementations of such algorithms are called *reasoners*. Most reasoner implementations are based on *tableau algorithms*. Basically, a tableau algorithm builds a canonical model for the ontology. This model is represented by a graph structure where nodes are domain elements that initially correspond to individuals, edges are role assertions between nodes and moreover nodes are labelled with concept names to which domain elements belong. The algorithm extends the initial graph by applying a set of rules which deal with the different statements that belong to the syntax of the language. If after applying the rules the obtained graph do not contain contradictions the ontology is consistent and the algorithm returns the logical consequences. Otherwise, the ontology is inconsistent[4] [Hitzler09].

More expressive description logics beyond $\mathcal{ALC}$ are named by different letters: $\mathcal{S}$ (role transitivity), $\mathcal{H}$ (role hierarchies, i.e. for role inclusion axioms), $\mathcal{O}$ (nominals, i.e. for closed concepts with one element), $\mathcal{I}$ (inverse roles), $\mathcal{N}$ (cardinality restrictions), $\mathcal{Q}$ (qualified cardinality restrictions), $\mathcal{R}$ (generalized role inclusion axioms) and $\mathcal{E}$ (existential role restrictions) [Hitzler09]. Syntax and semantics of some of the above description logics are addressed in Chapter 4 which describes an extension of the description logics $\mathcal{SHIQ}$. Besides the *Tbox* and the *Abox*, ontologies in some description logics more expressive than $\mathcal{ALC}$ (e.g. $\mathcal{SHIQ}$) have also a set of axioms called *Rbox* that represents intensional knowledge about roles. An *Rbox* has (generalized) role inclusion axioms (letters $\mathcal{H}$ for simple inclusion axioms and $\mathcal{R}$ for generalized) and role transitivity (letter $\mathcal{S}$).

Regarding the OWL language, there are two main versions: OWL1 and OWL2 (more expressive than OWL1), and there are different framents of OWL1 and OWL2 that include a restricted set of the description logics mentioned above. In particular, the more expressive fragment of OWL2 that keeps decidability is OWL2 DL which is based on the description logic $\mathcal{SROIQ}$. OWL2 EL, OWL2 RL and OWL2 QL are broadly used fragments of OWL2 DL which are less expressive but have less complexity and are useful in several scenarios, such as to deal with a great volume of instances or to formulate queries to the knowledge base [Hitzler09]. The present work uses the term OWL to denote (in general) all versions and fragments of the language, and when corresponds, a particular fragment is specifically mentioned.

2.2 Ontology networks and modular approaches

Nowadays, there are ontologies of different domains that naturally emerge, for instance SNOMED-CT for the health domain and OntoREA for accounting [SNOMED, Fischer-Pauzenberger17]. In general, independently built ontologies are used together in complex applications, without explicitly expressing the way how they are combined for a specific purpose. This situation leads to think on *ontology networks* as an ontology engineering concept instead of the custom-building of new ontologies

[4] Chapter 4 addresses details about tableau algorithms for the present work.

from scratch. An *ontology network* is defined as a collection of ontologies related together through a variety of different relationships such as mapping, modularization, and versioning, among others [Alloca09]. An ontology network differs from a set of interconnected ontologies in that the relationships among the different ontologies are explicitly expressed [Suarez-Figueroa10]. This aspect is important to visualize the conceptualization of a complex application at an upper level of abstraction as well as to facilitate the impact analysis when one of the ontologies evolves. Figure 1.1 of Chapter 1 shows an ontology network that combines three ontologies about protected areas, aquifers and policies for natural resources. Different relations connect the networked ontologies, e.g. the mapping of concepts from different ontologies or the meta-modelling relation between an instance of one ontology and a concept from another ontology.

When an ontology network is built from a set of ontologies, it is likely that ontologies were built separately by different engineers or domain experts. Then, each ontology is a conceptualization of some domain for a given context, and so, each ontology is interpreted in that context which is its *interpretation domain*. Hence, it is not a trivial issue how the contexts or domains of the networked ontologies are integrated in the domain of the ontology network, e.g. domains of ontologies could be disjoint or overlapped. Suppose that to conceptualize a given application it is needed to integrate the three ontologies of Figure 1.1, denoted $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$, in an ontology network. Suppose that $\mathcal{O}_1$ describes aquifers and has an axiom $\top \equiv River \sqcup Lake \sqcup Stream$. This statement should be interpreted as "everything is a river or a lake or a stream", which can be correct within the context of aquifers. However, when $\mathcal{O}_1$ is combined with $\mathcal{O}_2$ and $\mathcal{O}_3$ (that describe protected areas and policies of natural resources) in a single knowledge base, the statement $\top \equiv River \sqcup Lake \sqcup Stream$ means that "a protected area is a river or a lake or a stream", which is not longer valid. The cause of these undesired effects lies on the fact that some axioms do not make sense out of the context or domain of the ontology they belong to. $\top \equiv River \sqcup Lake \sqcup Stream$ makes sense in the domain of $\mathcal{O}_1$ but not in the domain of the ontology network that includes $\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$ and the relations among them. Actually, it is the ontology engineer who has to face this kind of situations when building an ontology network from a set of independently built ontologies. The problem of facilitating the work of engineers (often not experts in description logics) for scenarios as the one illustrated above breaks down in two problems: (i) what are the requirements for an autonomously developed ontology to be a reusable ontology, and (ii) how to relate the networked ontologies and deal with the contextuality of the knowledge and the reasoning on the ontology network [Homola10a]. There are different approaches that propose solutions to the kind of problems described above. In general, existing approaches address the reuse and integration of *modules* instead of ontologies. Inspired in software engineering, intuitively a *module* is a self-contained subset of an ontology (it can be the ontology itself) that encapsulates knowledge to be reused. The topic about building (big) ontologies (or ontology networks) composed by modules is commonly named *ontology modularization* or *modularity* [Grau07, Homola10a]. Some authors identify a set of properties which a module must hold such as escalability for evolution and maintenance, complexity management, reuse, collaborative development, understandabiliy and controlled integration, among others [Parent09, Grau07]. For the sake of simplicity and with the aim that the terminology keeps coherence with the notion of ontology network, in this work we refer to ontologies (as the parts of ontology networks) instead of modules (parts of ontologies built from multiple domains). Basically, there are two main research directions about ontology modularization which differ regarding syntax and semantics: *modular ontology languages* (MOL) and *modular reuse*.

Modular ontology languages (MOL) are a set of *formalisms that provide control over the interaction between ontologies*. Each MOL formalism solves a different kind of interaction and provides a special syntax to represent such interaction as well as a local syntax to represent each ontology of the ontology network. MOL formalisms define "distributed knowledge bases" which indeed are ontology networks with a distributed semantics approach where each ontology is locally interpreted in a *local domain*. Moreover, entailment rules are provided [Grau07, Homola10a]. Some of the most relevant MOL approaches are *ε-connections*, *distributed description logics (DDL)*, *integrated distributed description logics (IDDL)* and *package-based description logics (P-DL)* that are briefly described below.

- *ε-connections* allows relating two ontologies by and external role represented by a new construct called *link property*, and assumes that domains of related ontologies are disjoint [Grau06, Grau09a].

- *Distributed description logics (DDL)* defines distributed Tboxes and Aboxes, and solves the mapping between concepts, between roles and between individuals through the introduction of *bridge rules*. In this approach domains of combined ontologies can overlap [Borgida03, Serafini05a, Homola07, Serafini09, Homola10b].

- *Integrated distributed description logics (IDDL)* also allows the mapping between concepts and between roles by defining a construct called *correspondence*. Besides local (and possibly overlapped) domains, this approach defines a global domain which integrates local ones (local and global semantics) [Zimmermann07, ZimmermannD08].

- *Package-based description logics (P-DL)* supports that one ontology reuses knowledge from multiple ontologies, with local (and possibly overlapped) domains. This approach ensures that the meaning of reused ontologies is kept in any context [Bao09].

The *modular reuse* approach defines *a set of specialized non-standard reasoning services* that ensure a "safe" combination of ontologies in an ontology network. This approach defines a semantics which interprets the union of axioms of the ontologies into a single *global domain* [Grau08b, Grau09b, Sattler09, Konev09, Konev13]. Notions of *conservative extension* and *locality* are defined as the foundations of the reasoning services proposed by the modular reuse approach. Basically, these services are capable of formulating the same problems about interaction between ontologies as the MOL formalisms, but without introducing a special syntax and semantics. Moreover, the new reasoning services are able to be implemented on top of existing standard reasoning tasks, e.g. the tableau algorithm for OWL. Intuitively, an ontology network $\mathcal{ON}$ is a *conservative extension* of some of its networked ontologies $\mathcal{O}_1$ if it does not causes unexpected semantic consequences on the vocabulary of $\mathcal{O}_1$, i.e. not entailed by $\mathcal{O}_1$ itself. For example, suppose in the ontology network of Figure 1.1 $\mathcal{O}_1$ is the ontology that describes aquifers and $\mathcal{O}_2$ the ontology about protected areas, and instead of the relation $River \sqsubseteq NonRenewableResource$ between $\mathcal{O}_1$ and $\mathcal{O}_2$, the mappings $River \equiv NonRenewableResource$ and $Lake \equiv NonRenewableResource$ are declared. Hence, the ontology network infers the axiom $River \equiv Lake$ that involves only vocabulary of $\mathcal{O}_1$. However, $\mathcal{O}_1$ itself does not infer this entailment. In this case, the ontology network of Figure 1.1 is not a conservative extension of $\mathcal{O}_1$ (see [Antunes19] for a method to detect violations to the conservative extension property). A networked ontology is *local* if it does not fix the meaning of the universal concept $\top$ [Grau07]. For example, the ontology $\mathcal{O}_1$ of Figure 1.1, with the axiom $\top \equiv River \sqcup Lake \sqcup Stream$ is not a local ontology .

Figure 2.1 depicts the three different semantics approaches for ontology networks followed by the formalisms described above.

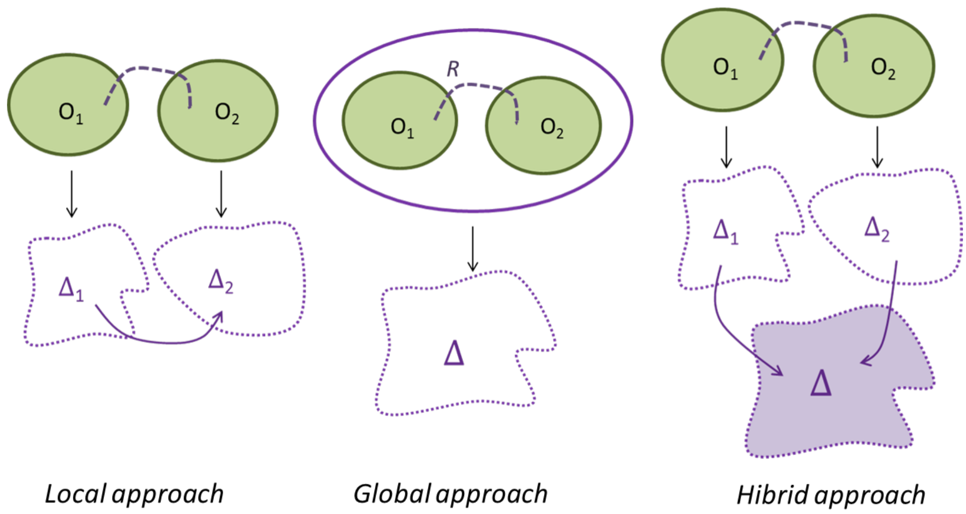

Figure 2.1: Semantics approaches for ontology networks

The *local approach* is followed by MOL formalisms ε-connections, DDL and P-DL. This approach was conceived to capture a *distributed* view of a domain, focused to *reuse*, i.e. the ontology engineer locally designs an ontology $\mathcal{O}_1$ by *reusing* vocabulary of another ontology $\mathcal{O}_2$ in such a way the meaning of symbols from the domain of $\mathcal{O}_2$ is brought into the local domain of $\mathcal{O}_1$. For the three formalisms ε-connections, DDL and P-DL, extensions of standard reasoning algorithms exist to deal with the different kind of interactions between ontologies, but according the review done in the present work, only extensions for ε-connections and DDL have been implemented [SWOOP, Serafini05b].

The *global approach* is followed by the modular reuse formalism. The idea is that an ontology network is built from an ontology $\mathcal{O}_1$ by *reusing* and *integrating* the vocabulary of an autonomously built ontology $\mathcal{O}_2$ and then vocabularies of $\mathcal{O}_1$ and $\mathcal{O}_2$ are interpreted into a single domain. Reasoning services are formally defined and implemented [LOCMOD].

The *hibrid approach* is followed by the MOL formalism IDDL. It captures a *distributed and integrated* view of a domain by transforming local domains into a global domain. A formally defined and implemented reasoning algorithm supports this approach [DRAON].

Table 2.2 summarizes the syntax and semantics of the main formalisms about

Research direction	Formalism	Semantics approach	Design perspective	Reasoning
	ε-connections	local	distributed, reuse	defined/implemented
MOL	DDL	local	distributed, reuse	defined/implemented
	IDDL	hibrid	distributed, integrated	defined/implemented
	P-DL	local	distributed, reuse	defined/not implem.
Modular reuse	Description logics	global	reuse, integrated	defined/implemented

Table 2.2: Comparison of ontology modularization approaches

ontology modularization.

Some works have compared the above approaches in deep. Wang et al. compare ε-connections, DDL and P-DL regarding the ability of each approach to support networked, dynamic and distributed ontologies to conceptualize a case study about fishery [Wang07]. Serafini et al. compares MOL approaches identifying their differences from the point of view of different scenarios of use and the transitivity of interactions between ontologies [Serafini09]. Homola et al. analyze reductions from DDL to P-DL and from DDL to ε-connections, and map each formalism to an intuitive name that describes the kind of interaction it solves: *linking* for ε-connections, *mapping* for DDL and IDDL and *importing* for P-DL [Homola10a]. Grau et al. present a complete summary of the MOL approaches ε-connections, DDL and P-DL, as well as a comparison with the modular reuse approach [Grau07]. Regarding MOL approaches, the authors emphasize in their common aspects such as a similar distributed and local semantics and a special syntax to model the interaction between ontologies. By contrast, the modular reuse formalism proposes a unified global semantics keeping the standard description logics syntax and providing a set of reasoning services to solve problems in the interactions between ontologies. As advantages of the modular reuse approach Grau et al. emphasize that the specification of ontology languages such as OWL does not need to be modified, and that new reasoning services can be implemented on top of the standard ones. Regarding MOL approaches, the authors identify both drawbaks and advantages. On the one hand, the distributed semantics makes MOL approaches a clean way of controlling the interaction between ontologies and moreover, the special syntax can be convenient for the ontology engineer to distinguish which axioms play the role of combining different ontologies and which are used within each ontology. On the other hand, MOL approaches can be confused for the ontology engineer since each MOL formalism provides a different syntax to express the kind of relation it solves, and moreover, the treatment of entailment propagation across ontologies is not the same for all MOL formalisms.

Some works intend to counteract the problem of the syntactic and semantic differences among MOL approaches. Brockmans et al. define “a mapping metamodel that allows us to encode the formal differences on the conceptual level and facilitate the selection of an appropriate formalism on the basis of a formalism-independent specification of semantic relations between different ontologies by means of a graphical modelling language” [Brockmans09]. Ghidini et al. propose a more general formalism called *distributed first order logic* (DFOL) that allows encoding the kind of relations between ontologies provided by ε-connections, DDL and P-DL [Ghidini17].

As a brief conclusion, two main aspects are in favor of the widely used modular reuse approach: (i) syntax and semantics are simpler and (ii) it covers the different ways of interaction among ontologies (each one solved by a different MOL approach).

2.3 Meta-modelling

Chapter 1 introduces the notion of meta-modelling in the context of ontology networks as a correspondence between individuals and concepts from different ontologies, such that the individual and the concept have the same meaning for the ontology engineer. This approach is coherent with ontology engineering best practices of modularization and reuse, keeping different visions of the same real object in different ontologies. For example, in the ontology network of Figure 1.1, for the domain of policies about natural resources *river* is visualized as a particular natural resource which is necessary to preserve. However, for the domain of aquifers *River* is visualized with a greater granularity as the set of rivers of a region or country. A more general notion of meta-modelling also considers the correspondence between individuals and roles, e.g. the case study of recommender systems in Chapter 3 represents *hasAuthor* as an individual and as a role. Moreover, meta-modelling can be considered within the same ontology, dropping the restriction that the individual and the concept belong to different ontologies. For instance, if instead of designing the ontology network of Figure 1.1 from autonomously built ontologies an engineer models the whole domain from scratch, the ontology would look like the one illustrated in Figure 2.2. In this ontology there is a Tbox axiom $River \sqsubseteq NonRenewableResource$ with *River* playing the role of concept, and an Abox axiom $NaturalResource(River)$ with *River* playing the role of individual. In spite of *River* and $NaturalResource$ are both concepts they belong to different *meta-modelling levels*. *River* has level 1 since it has instances that are atomic objects (level 0), as the Uruguay river, whereas $NaturalResource$ has level 2 since it has instances that are concepts of level 1. Moreover, to represent real objects a desirable property for ontologies with meta-modelling is the *well-foundedness of the interpretation domain* [Barwise96]. To understand the intuition behind this notion suppose that someone introduces the Tbox axiom $NaturalResource \sqsubseteq River$ in the example of Figure 2.2. By the Abox axiom $NaturalResource(River)$, the interpretation of *River* belongs to the interpretation of $NaturalResource$, which by $NaturalResource \sqsubseteq River$ is also a subset of *River*, i.e. $River^{\mathcal{I}} \in NaturalResource^{\mathcal{I}} \sqsubseteq River^{\mathcal{I}}$. Then, *River* belongs to itself, what means that the interpretation domain is not *well-founded*.

Two main research directions exist for meta-modelling: (i) conceptual oriented meta-modelling approaches and (ii) description logics meta-modelling approaches. Before describing them, a couple of clarifications are introduced below to better understand the different approaches to meta-modeling and their application to real cases studies (addressed in Chapter 3).

- There are some approaches which address issues about metalevel information such as provenance or access rights to domain ontologies [Tran08]. However, it is not the kind of meta-modelling addressed in the present thesis work, so it is intensionally excluded from the related work discussion.

- Considering a more philosophical approach to ontologies, the notion of *instances* or *individuals* strictly denotes “not instantiable objects” [Keet18,

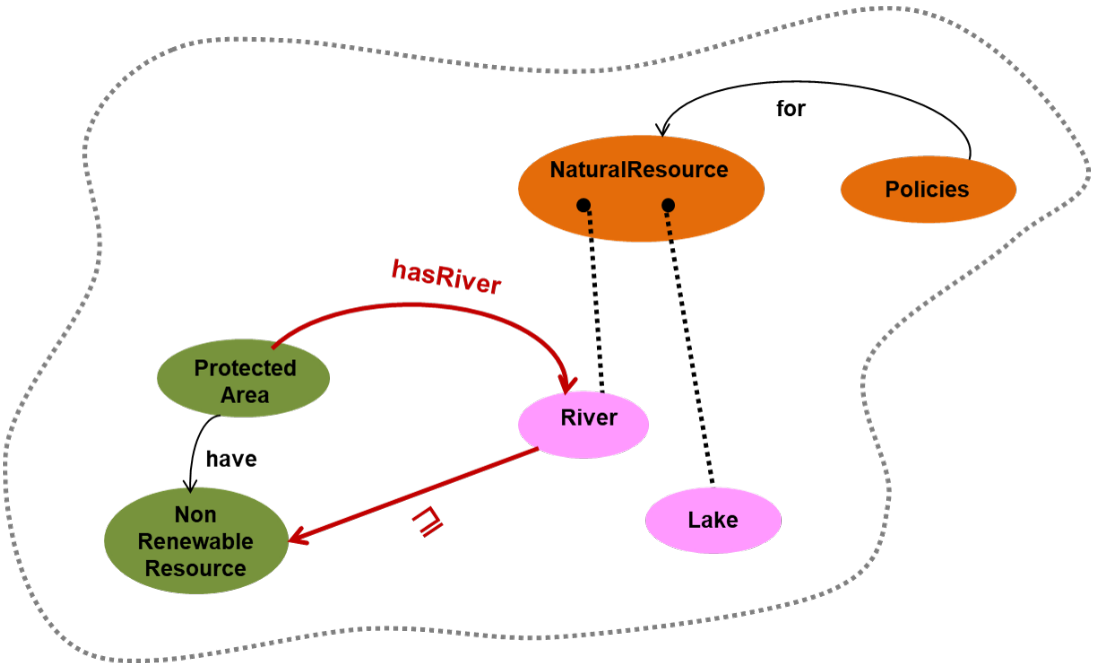

Figure 2.2: An ontology about protected areas and politics of natural resources

Chapter 6]. Even though it is a sound principle for a single knowledge level[5], the main subject of the present thesis work is about representing different "perspectives" or views of a domain, which is a set of real objects. Hence, whereas from a given perspective a real object can be viewed or "managed" as an atomic object (e.g. the instance *river*), from another point of view the same object can be viewed as a set of objects (e.g. rivers Santa Lucia and Queguay). Moreover, from a logical approach, the domain of interpretation does not necessarily contains only atomic objects, as is described later in this chapter for the description logics meta-modelling approaches.

2.3.1 Conceptual oriented meta-modelling approaches

Regarding conceptual modelling and design, the meta-object facility standard (MOF) is broadly adopted for many areas such as model driven engineering [MOF, Schmidt06]. It describes modelling languages such as UML, which specify object oriented models and are implemented by frameworks such us Eclipse Modeling Framework [UML, ECLIPSE]. Then, MOF allows the definition of an structure of layers of models which in the end describes an application domain as a set of classes and instances of classes. However, the meta-modelling approach of the present thesis work addresses the need of a lot of applications of representing several layers or knowledge levels (as classes with instances that also are classes) within the domain itself.

The conceptual oriented meta-modelling emerges from some approaches that have identified the benefits of ontologies for conceptual modelling. These approaches use MOF languages to represent ontologies and have proposed ONTOUML, a language for ontology-driven conceptual modelling based on the Unified Foundational Ontology (UFO) [Guizzardi12, OntoUML]. Some approaches go further and extend

[5]Each object is represented with a single level of granularity (either as an individual or as a concept)

MOF languages to conceptualize a domain with more than one knowledge level. Among them, Atkinson et al. coin the term "clabject" to emphasize that there are classes that are also objects [Atkinson15a, Atkinson15b]. De Lara et al. present design patterns for meta-modelling using an extended MOF language and Carvalho et al. introduce a first order logic theory, called MLT theory [DeLara14, Carvalho17, Carvalho18]. All these approaches focuse on ontology design, by making interesting contributions for ontology engineers with the identification of design patterns and antipatterns for ontologies with meta-modelling, which are addressed in Section 2.4 [Brasileiro16, Kuhne18, Almeida18]. However, from the review done in the present work, none of these approaches define the semantics of such extensions based on the notion of interpretation of description logics, which underlies standard ontology languages such as OWL. Note that the MLT theory formalizes meta-modelling by representing classes (and meta-classes) as variables, and also constants $1stOT$, $2ndOT$ and $3rdOT$ that represent sets of all classes, meta-classes and meta-meta-classes. However, according to the standard translation of description logics into first order logic, classes are translated into first order predicates [Hitzler09]. Moreover, these meta-modelling approaches do not propose mechanisms to generate (automatically) constraints on final applications, e.g. by the implementation of algorithms that check consistency and draw inferences from the ontology conceptualization.

2.3.2 Description logics meta-modelling approaches.

Even though the ultimate goal of description logics is the conceptualization of real scenarios, the main difference between conceptual oriented meta-modelling approaches and description logics meta-modelling approaches is that the main target of the latter is the formalization of sound extensions of description logics (and reasoning algorithms based on them) to deal with meta-modelling. In the following the main description logics meta-modelling approaches are described and compared.

Among the MOL approaches for ontology networks addressed in Section 2.2, the *DDL* approach defines unidirectional *bridge rules* to represent relations between ontologies (or modules). Some bridge rules allow the mapping of an individual from an ontology $\mathcal{O}_1$ to a set of individuals in another ontology $\mathcal{O}_2$ (possibly instances of a concept) [Borgida03]. Even though this approach can be considered as a mechanism to express meta-modelling, the proposed syntax does not explicitly declare the correspondence between an individual and a concept, so it is not considered in the discussion below.

To better describe existing description logics meta-modelling approaches, in this work they are classified by two different criteria regarding their semantics:

- according to the definition of the *meta-modellig levels in the interpretation domain*, into *fixed layered* and *global layered* approaches, and
- according to the *relation between intensions and extensions*, into *Henkin* and *Hilog* approaches

Regarding definition of the *meta-modelling levels in the interpretation domain*, *fixed layered* approaches define the interpretation domain Δ of an ontology with meta-modelling as the disjoint union of domains Δ_i which correspond to meta-modelling levels i, with $1 \leq i \leq n$, $n \in \mathbb{N}$. These approaches follow a style like the local and hibrid semantic approaches for ontology networks adopted by MOL approaches (see Figure 2.1 parts a and c in Section 2.2), but with meta-modelling layers instead of ontologies. Like them, *fixed layered* approaches introduce a special syntax that makes explicit the meta-modelling level of concepts and roles, and also provide control over the interaction between layers. For example, the concept $River$ of Figure 2.2 is denoted as $River^1$ and the concept $NaturalResource$ as $NaturalResource^2$. Moreover, *fixed layered* approaches introduce restrictions on the levels of concepts and roles that can be combined to define general concepts, roles and axioms, as a mechanism for controlling the interaction of layers. In particular, all instances of a (general) concept must belong to the immediate lower level. By contrast, *global layered* approaches define a unique domain Δ and interpret all layers into Δ. The way layers are mixed into the unique domain differs for each particular approach, and also depends on how each approach relates intensions and extensions (second criteria). Semantics of *global layered* approaches is similar to the global semantics followed by the modular reuse approach for ontolology networks (illustrated in Figure 2.1 part b).

Regarding the *relation between intensions and extensions*, an intuitive notion of intensional and extensional knowledge is given in Section 2.1. Given a symbol X of an ontology $\mathcal{O}$, the interpretation of X as an individual is called *intension* whereas the interpretation of X as a concept or role is called *extension*. Two properties called *intensional regularity* and *extensionality* determine how the intensions and extensions of symbols are related [Homola13]. Given two symbols (or names) A, B that are treated both as individuals and concepts in an ontology $\mathcal{O}$ [6], properties of *intensional regularity* and *extensionality* are expressed as follows.

1. If $\mathcal{O} \models A = B$ then $\mathcal{O} \models A \equiv B$ (*intensional regularity*).
2. If $\mathcal{O} \models A \equiv B$ then $\mathcal{O} \models A = B$ (*extensionality*).

Recall from Section 2.1 that $A = B$ represent the equality of individuals A and B whereas $A \equiv B$ is the equivalence of concepts A and B. It is clear that when the interpretation of A in $A = B$ (intension) is equal to the interpretation of A in $A \equiv B$ (extension), then the semantics must satisfy both *intensional regularity* and *extensionality*. *Henkin*-style semantics gives a unique interpretation to the individual (intension) and the corresponding concept (extension), so it satisfies properties of *intensional regularity* and *extensionality*. Hence, all syntactic higher-order objects (as concepts with instances that are also concepts) have a direct set-theoretical interpretation via a hierarchy of power sets. For example, if A has two instances a and b, a possible interpretation of A as individual and as concept is the set $\{a, b\}$. Then, for $B(A)$, the interpretation of B will be a set of sets, i.e. $\{\{a, b\}\}$. However, for *Hilog*-style semantics each name has as *intension* a domain element which acts as an *identifier* of the name, and when the name is used as a concept then the *intension* has associated an *extension* that is the set of intensions of the concept's instances [Kubincova16a, Kubincova16b]. A given name X is interpreted by its intension when it plays the role of individual whereas is interpreted by its extension (different from the intension) when it plays the role of concept. The Hilog semantics satisfies *intensional regularity* but does not satisfy *extensionality*. [Motik07, Giacomo11, Kubincova16b].

[6]Note that in $C(A)$ the name A is an individual whereas in $A \sqsubseteq B$, A is a concept

Approach	Base DL	Domain levels	Henkin /Hilog	Unlim. levels	Inter-layer roles	Well-foundness	Meta-mod. for roles
Punning	$\mathcal{SROIQ}$	Global	-	Y	Y	N	Y
Glimm et al.	$\mathcal{SROIQ}$	Fixed	Hilog	N	Y(1-2)	Y	N
Pan et al.	$\mathcal{SHOIN}$	Fixed	Henkin	Y	N	Y	Y
Motik (υ-sem.)	$\mathcal{ALCHIQ}$	Global	Hilog	Y	Y	N	Y
De Giacomo et al.	$\mathcal{SHIQ}$	Global	Hilog	Y	Y	N	Y
Lenzerini et al.	OWL2 $\mathcal{QL}$	Global	Hilog	Y	Y	N	Y
Homola et al.($\mathcal{TH}$)	$\mathcal{SROIQ}$	Fixed	Henkin	Y	Y	Y	N
Homola et al.($\mathcal{TH}$)	$\mathcal{SROIQ}$	Fixed	Hilog	Y	Y	Y	N
Homola et al.($\mathcal{HIRS}_*$)	$\mathcal{SROIQ}$	Global	Hilog	Y	Y	N	Y

Table 2.3: Comparison of meta-modelling approaches

Table 2.3 shows the main description logics meta-modelling approaches. The second column (Base DL) specifies the description logic that is extended whereas third and fourth columns identify the essential differences among meta-modelling approaches according to the classification criteria mentioned above. The column "Domain Levels" with values "Fixed" and "Global" denotes the semantics of the *meta-modelling levels in the interpretation domain* and the column "Henkin/Hilog" corresponds to the semantics of intensional and extensional knowledge. The remaining columns show if the approach allows for an unlimited number of levels, if roles can be defined across different levels, if the approach ensures the well-foundedness of the interpretation domain and allows meta-modelling for roles (concepts with instances that are roles).

The meta-modelling approach called *Punning* provided by OWL 2 allows using the same name as an individual, a concept and a role, but the reasoner treats them as if they were actually different [Hitzler09]. It satisfies neither intensional regularity nor extensionality properties so it does not even follow a Hilog-style semantics. As *Punning* does not impose restrictions about the combination of names in the ontology, then it allows for unlimited levels, inter-layer roles and meta-modelling for roles. It does not ensure the well-foundedness of the interpretation domain. Moreover, existing reasoners do not consider the meta-modelling correspondence between interpretations of the same name as individual and concept to check consistency.

Glimm et al. codify meta-modelling within OWL for only two levels of knowledge without defining a set-theoretical semantics for meta-modelling [Glimm10]. Given an ontology $\mathcal{O}$, the authors define an ontology $\mathcal{O}'$ by extending $\mathcal{O}$ with two roles *type* and $subClassOf$, the disjoint concepts $Class$ and $Inst$, and a set of axioms that separate $\mathcal{O}'$ in two fixed layers, the upper layer represented by the concept $Class$ and the lower layer represented by the concept $Inst$. For each Abox axiom $A(a)$, the axiom $type(a, O_A)$ is added, which relates the individual a in the lower level with the new individual O_A, instance of $Class$, that represents the concept A in the upper level. For each axiom $A \sqsubseteq B$, the axiom $subClassOf(O_A, O_B)$ relates individuals O_A, O_B in the upper level to represent that A is a subset of B in the lower level. Hence, even though there is no special semantics, for any model of $\mathcal{O}'$ the domain Δ is divided in two fixed layers, the upper layer given by the interpretation of the concept $Class$ and the lower layer given by the interpretation of the concept $Inst$. Indeed, for each concept A the interpretation of the individual O_A play the role of the intension of A whereas the interpretation of A itself is the extension. The set of added axioms ensures the property of *intensional regularity* but it does not ensure *extensionality.* In fact, it is an encoding of meta-modelling which is also used to formalize the rules from the OntoClean methodology in OWL [Guarino09].

Pan et al. introduce the *OWL FA* extension to OWL DL [Pan05, Jekjantuk09, Jekjantuk10, Groner12]. The semantics of *OWL FA* is tipically *fixed layered*, i.e. given a concept of a certain level all its instances belong to the immediate lower level. It also provides meta-modelling for roles, hence instances of a concept of the level $i+1$ can be concepts or roles of the level i. However, inter-layer roles are not allowed. As well as for all fixed layered approaches, the syntax of *OWL FA* forces to represent the meta-modelling level in the concept or role name. They give the same interpretation to the individual and the corresponding concept by meta-modelling, so they follow a *Henkin*-style semantics. Starting from a domain Δ_0 of atomic objects, the domain of every level $i > 0$ is defined as $\Delta_{i+1} = \mathcal{P}(\Delta_i) \cup \mathcal{P}(\Delta_i \times \Delta_i)$. Both the fixed layered and the Henkin semantics ensure the well-foundedness of the interpretation domain.

Motik proposes two alternative semantics for meta-modelling: the context approach (π-semantics), similar to punning of OWL 2, and the Hilog approach (υ-semantics) [Motik07]. The *υ-semantics* is a *global layered* and *Hilog* extension of $\mathcal{ALCHIQ}$ that allows using the same name as an individual, a concept and a role. Each name has associated an intension that belongs to the (unique) domain Δ, and when the name is used as concept (role) the intension is associated to an extension that belongs to $\mathcal{P}(\Delta)$ ($\mathcal{P}(\Delta \times \Delta)$). As it satisfies intensional regularity, it detects some inconsistencies such as the coexistence of axioms $A = B$ and $A \sqcap B \sqsubseteq \bot$. However, it does not ensure the well-foundedness of the interpretation domain. It also allows meta-modelling for roles since for some names A, R, a, b, axioms $A(R)$, $R(a, b)$ are allowed; R in $A(R)$ is interpreted by its intension in Δ whereas in $R(a, b)$ is interpreted by its extension in $\mathcal{P}(\Delta \times \Delta)$. This approach has been extended to more expressive logics and query languages [Gu16, Gu18].

The approach followed by Giacomo et al., called Higher Order Description Logics, is a *global layered* and *Hilog*-style semantics extension of $\mathcal{SHIQ}$, very similar to the υ-semantics of Motik [Giacomo11]. However, the Hilog semantics of Giacomo et al. assigns intensions also to general concepts and roles, e.g. the concept $A \sqcap B$, whereas Motik assigns intensions only to atomic concept and role names. Lenzerini et al. follow the same approach for lighweight description logics such as OWL2 QL, a profile targeted to scenarios of large amount of data to be accessed through conjunctive queries, addressing more practical scenarios of ontology-based data access [Lenzerini14, Lenzerini16a, Lenzerini16b, Cima17a, Cima17b, Lenzerini18].

Rows seventh and eighth of Table 2.3 shows two semantics approaches of the *typed higher order description logic* $\mathcal{TH}(\mathcal{L})$ of Homola et al. with $\mathcal{L}$ a description logic up to $\mathcal{SROIQ}$ [Homola13, Homola14]. Both alternatives have a *fixed layered* semantics with a syntax that restricts the sets of concept and role names to be pairwise disjoint, and moreover “typed” with the level of meta-modelling. The *Henkin* approach is similar to the *OWL FA* semantics of Pan et al. although it does not allow meta-modelling for roles. Given a domain Δ_0 of atomic objects, each domain level $i > 0$ is defined as $\Delta_i \subseteq \mathcal{P}(\Delta_{i-1})$, then typed concepts A^i are interpreted as a subset of the immediate lower level Δ_{i-1} but no other level [Homola13]. However, unlike *OWL FA*, $\mathcal{TH}(\mathcal{L})$ allows inter-layer roles that have domain and range in different levels, i.e. typed roles R^{ij} that are interpreted as a subset of $\Delta_{i-1} \times \Delta_{j-1}$. The $\mathcal{TH}(\mathcal{L})$

Hilog alternative defines a domain Δ as the disjoint union of domains Δ_i, $i \geq 0$, and Δ_{ij}, $i, j > 0$ [Homola13, Homola14]. A typed concept A^i has an intension $a \in \Delta_i$, which has an extension $a^\epsilon \subseteq \Delta_{i-1}$, and a typed role R^{ij} has an intension $r \in \Delta_{ij}$, which has an extension $r^\epsilon \subseteq \Delta_{i-1} \times \Delta_{j-1}$. Both alternatives (Henkin and Hilog) ensure the well-foundedness of the domain since they are fixed layered. The authors are more aligned with the Hilog approach, arguing that the Henkin semantics is too strong for certain case studies of particular domains.

The *Hilog* approach of Homola et al. evolves to a *global layered* semantics $\mathcal{HIRS}_*(\mathcal{L})$ (ninth row of Table 2.3) for $\mathcal{L}$ up to $\mathcal{SROIQ}$ [Kubincova15, Kubincova16a, Kubincova16b]. Besides allowing meta-modelling for roles, inspired in the work of Glimm et al. it also provides *instantiation* by adding the role $instanceOf$, similar to the role *type* of Glimm, and *subsumption* by adding the role $subClassOf$ as Glimm et al. do. $\mathcal{HIRS}_*(\mathcal{L})$ is similar to the approach of Motik et al. but keeping the disjointness of the sets of concept, role, and individual names. For this, it defines the domain Δ as the disjoint union of Δ_I for individual intensions, Δ_C for concept intensions and Δ_R for role intensions. Despite the domain is divided into three domains, the approach is global layered since Δ_C (Δ_R) is the set of intensions of concepts (roles) belonging to any meta-modelling level. Moreover, given an intension $c \in \Delta_C$ ($r \in \Delta_R$) the associated extension $c^\epsilon \subseteq \Delta$ ($r^\epsilon \subseteq \Delta \times \Delta$) is a subset of the unique domain Δ since a concept can have instances that are roles. This approach does not ensure the well-foundedness of the interpretation domain.

Regarding consistency checking, approaches of Pan et al., Motik, Giacomo et al. and Homola et al. define algorithms that reuse the tableau algorithm for OWL by reducing ontologies with meta-modelling to ontologies in OWL.

The main conclusions about meta-modelling aproaches that extend description logics can be summarized as follows.

- Fixed layered approaches have the drawback of having a rigid syntax that forces to mark concepts and roles with the meta-modelling level, which results very restrictive from the point of view of modelling choices. However, it ensures the well-foundedness of the interpretation domain. Global layered approaches are flexible both regarding the syntax and modelling alternatives, but do not ensure the well-foundedness of the domain.

- Henkin semantics is stronger than Hilog semantics since it satisfies both intensional regularity and extensionality. Hence, Henkin approaches detect inconsistencies which Hilog approaches do not, e.g. the coexistence of axioms $A \equiv B$ and $A \neq B$[7]. However, even though there are scenarios (of rather closed domains) that require a Henkin style of semantics for a strong checking of inconsistencies, there are other scenarios (of more open domains) for which the Hilog semantics results more appropiate. Chapter 3 describes a set of case studies that are better modelled by Henkin approaches and moreover a scenario for which Hilog is a more suitable modelling alternative.

- As Table 2.3 shows, there are meta-modelling approaches for the combinations Henkin-fixed layered, Hilog-fixed layered and Hilog-global layered, but from the review of the literature done, we conclude that there is no approach that follows a Henkin semantics with a global layered domain. The present thesis

[7] A and B can be different as individuals and equivalent as concepts

work presents arguments and motivations to fill this gap, which are described in chapters 3 and 4.

2.4 Ontology engineering and design patterns

Ontology engineering is the discipline of developing ontologies. On the one hand, ontology engineers need to have a thorough understanding of the domains under study which is mainly obtained from domain experts, but on the other hand, they must have a solid understanding of how these domains are best represented by the logical axioms that make up ontologies [Hammar17]. Given that ontologies are often reused or integrated as components of larger systems, ontology design mistakes can result expensive to fix. To minimize such mistakes, different methodologies and tools have been proposed, some of them covering different engineering phases from requirements elicitation to the ontology implementation. Some of them include the identification of different classes of problems that can be solved through a common solution called a *design pattern*, such as the design patterns proposed in software engineering [Gamma95, Fowler97, Buschmann96]. An *ontology design pattern* (ODP) is a modelling solution to a recurrent ontology design problem [Gangemi09].

Sections 2.2 and 2.3 in the present chapter describe different approaches to integrate ontologies that conceptualize both a single level of abstraction and moreover two or more knowledge levels. On the one hand, Section 2.2 describes different semantics to control the interaction between (modules of) ontologies that conceptualize a single level of knowledge. Among them, the semantics of the modular reuse appropach is the most common adopted semantics [Grau07]. This approach interprets networked ontologies and its relations into a unique domain, adhering to the direct model-theoretic semantics of OWL2 DL. Hence, regarding methodologies for single level ontologies, the present work assumes the modular reuse approach whenever the adopted semantics is not specified. On the other hand, Section 2.3 presents different description logics approaches to deal with more than one level of abstraction, i.e. meta-modelling. In this case it is not so evident the semantics adopted to interpret an ontology (nework) with meta-modelling. As is showed in Section 2.3, different styles of semantics change the meaning of the conceptualization, and at the moment there is no standard semantics for meta-modelling. Hence, whenever a methodology of ontology design with meta-modelling does not explicitly adopt a semantics approach, no particular semantics is assumed.

The two following sections describe some related work about ontology (network) design methodologies and patterns, for conceptualizing a single knowledge level and for conceptualizing several abstraction levels.

2.4.1 Methodologies for single level approaches

The most commonly adopted approach to conceptualize a domain is the definition of concepts, roles and individuals such that individuals are interpreted as atomic objects (not instantiable), which represents a single level of knowledge. This section describes some works about methodologies and design patterns for single-level knowledge modelling.

The NeOn methodology [SuarezFigueroa12]. NeOn is a methodology which proposes guidelines for ontology network development and covers all engineering activities from ontology requirements, conceptualization, implementation and evolution, and also proposes design patterns, and addresses ontology modularization and evaluation. This methodology takes elements from single ontology methodologies such as Methontology, On-To-Knowledge and DILIGENT, which address the conceptualization of a single ontology [Gomez-Perez10, Staab01, Vrandecic05]. Unlike them, NeOn focuses in the reuse and integration of existing resources. The authors identify nine scenarios that include developing an ontology network from scratch, reusing and re-engineering ontological and non ontological resources, and consider the alignment and merge of ontologies and the reuse of ODPs. They also define a set of life cycle models for organizing ontology development activities, and relate them to the most suitable of the nine identified scenarios. The NeOn methodology classifies design patterns into *structural*, *correspondence*, *reasoning*, *presentation*, *lexico-syntactic*, and *content* (and subcategories of them), and deepens in the content (or domain) ODPs. They propose the *eXtreme Design* (XD) methodology that emphasizes in the ontology design by reusing ODPs, which corresponds to one of the nine proposed scenarios. Additionally, a recommendation for the scheduling in ontology engineering projects is provided. Regarding the semantics of ontology networks, the authors describe how to locate axioms that cause inconsistencies in the ontology network.They also propose the NeOn toolkit to facilitate the development process and presents the application of the methodology to a set of case studies.

Patterns and antipatterns [Falbo13, PrinceSales15, PrinceSales16]. The group of Guizzardi et al. presents a set of design patterns and antipatterns which are mainly oriented to detect common errors made by ontology engineers in the conceptualization of a domain.

Falbo et al. reorganizes the classification of ontology patterns presented by the NeOn methodology and Gangemi et al., by making an analogy between patterns in software engineering and ontology engineering [Gangemi09, Falbo13]. They classify ontology patterns according to the phases in the ontology development into *conceptual patterns*, *architectural patterns*, *design patterns* and *idioms*. *Conceptual patterns*, that are independent of the ontology language, correspond to the ontology conceptual modelling development phase. They are also classified into *foundational ontology patterns* (reusable fragments of foundational ontologies) and *domain-related ontology patterns* (reusable fragments of reference domain ontologies). *Architectural patterns* describe how to arrange ontologies and modules in ontology networks; they are defined to be used both in the ontology conceptual modelling and the ontology design development phases. *Design patterns* correspond to the ontology design phase. They are also classified into *reasoning ontology patterns*, which address aspects such as computational tractability, decidability and performance, and *logical ontology patterns*, that aim to solve problems about the expressivity of the formalism used. *Idioms* are related to the implementation phase, they describe how to solve certain scenarios by using a particular ontology language, e.g. OWL. Two ways of reusing ontology patterns are proposed: by *analogy*, that corresponds to reproduce the structure of the pattern in the domain being developed, and by *extension*, that consists in reusing and extending the structure of the pattern with new concepts and roles specific to the problem being solved. The work also presents different examples of reuse of patterns by analogy, extension and the combination of both mechanisms.

Prince Sales and Guizzardi identify a set of ontological anti-patterns by exploring 54 models in OntoUML [PrinceSales15, PrinceSales16]. The authors define anti-patterns as “a recurrent error-prone modeling decision” and as “model structures that, albeit producing syntactically valid conceptual models, are prone to result in unintended domain representations”. OntoUML is a conceptual modelling language designed with the expressivity to conceptualize the Unified Foundational Ontology (UFO) [UFO]. The expressivity of OntoUML is mainly given by the definition of stereotypes which are enhanced by OCL meta-properties [OntoUML]. The identification of anti-patterns aims to support ontology engineers in “building the (ontologically) correct model for the domain”, assisting in the model validation and in “building models correctly” (model verification). Sales et al. address problems in the definition of relations (roles), e.g. detection of cycles in relations and (non) reflexive relations [PrinceSales15]. They also address the identification of problems in the modelling of types (concepts) with the “role” stereotype which corresponds to an “anti-rigid” type according to the Ontoclean methodology (that evaluates the conceptual consistency of the models) [Guarino09, PrinceSales16]. An example of the latter is the detection of mandatory relations from “rigid” types (essential properties, instances belong to them while existing) to other types called “relator” that cannot be correct since rigid types are independent and should not be obligatorily related to a relator (as anti-rigid types do since they represent accidental properties of instances). Both works propose a set of refactoring plans to fix the unexpected consequences caused by the anti-patterns, and moreover provide a tool to help in the detection of anti-patterns and in the execution of the refactoring.

The eXtreme Design (XD) methodology [Blomqvist16]. XD was initially proposed as one of the scenarios of the NeOn methodology. However, it is sufficiently self-contained to be an ontology engineering methodology by itself. XD proposes an agile and iterative life cycle with three different phases for the ontology (network) development: (i) a project initiation and scoping step which is generally performed once at the start of the project, (ii) a development loop that iteratively produces new modules, and (iii) an integration loop that adds ontology increments to the overall solution. The first phase consists in collecting requirements by developing use stories (mainly for functional requirements). In the second phase, priorized stories for each iteration are developed as ontologies or modules, and if some stories are overlapping their solutions are merged. In this phase the more suitable ODPs are reused and specialized and/or composed if necessary. Finally, ontologies are integrated by identifying the suitable points of alignment.

Simplified Agile Methodology for Ontology Development (SAMOD) [Peroni16]. To validate if a developed ontology is consistent with requirements, this methodology promotes the use of three types of tests: model, data, and query tests. For this, SAMOD proposes a process of three main steps that iterate once for every requirements scenario. Basically, the steps consist on the following activities: (i) collect and formalise requirements, specify their tests, and construct an ontology that covers the requirements and fulfils the tests, (ii) merge the built ontology (which is an increment) and its tests with the branch ontology under development, and (iii) as a precondition to proceed to the next iteration, refactor the main branch as needed by executing all the tests defined for the increment and the main branch. The methodology adheres to some design principles such as develop small ontologies, reuse knowledge in the form of ODPs, employ a middle-out conceptualization strategy [8] and use self-explanatory and human readable entity naming.

Content Ontology Design Patterns: Qualities, Methods, and Tools [Hammar17]. This thesis work presents the *ODP Methodology Development* as an improvement of the XD methodology. It takes elements from single ontology methodologies such as Methontology, On-To-Knowledge and DILIGENT, and from agile methodologies such as SAMOD. It also analyzes ontology design pattern classifications from methodologies to develop ontology networks, such as the classification of the NeOn methodology and the one proposed by Blomqvist [Blomqvist09]. The latter is based on the granularity of the reusable solution, it distinguishes *application*, *Architecture*, *Design*, and *Syntactic* patterns. Multiple quality frameworks are inputs of the *ODP Methodology Development*. Among them, *MAPPER* addresses the evaluation of processes and conceptual models, *Conceptual Model Quality* identifies six relevant quality factors for models (changeability, reusability, formalness, mobility, correctness and usability), *Entity Relationship Model Quality* provides quality metrics that can be reused for ODP quality, *Information System Quality*, since ontologies are used as components within information systems, and *Pattern Quality* provides object oriented design pattern quality indicators that are applicable to ontology design patterns [Sandkuhl08, Thorn10, Genero02, ISO, Chen09]. Moreover, some approaches that present ODPs as simple, small, and reusable modular ontologies are considered, among them: *O2* and *oQual* that are two meta-ontologies for understanding, classifying, and selecting ontologies, *Ontometric* that presents a criteria as a hierarchy of dimensions (different aspects that the ontology engineers has to solve when building an ontology), and *Ontoclean*, a methodology for evaluating the conceptual consistency of ontologies [Gangemi06, Tello04, Guarino09]. Then, this thesis work presents an *ODP Quality Model* where the most relevant result is a hierarchy of quality characteristics that guides the ontology engineer for selecting the correct design pattern and provides criteria for specializing patterns that solve concrete modelling problems. The final result of the work of Hammar is the *ODP Methodology Development*. It addresses relevant aspects such as the definition of a set of roles and responsibilities in the development of ontologies, provides guidelines for ontology reuse and proposes a context-based methodology adaptation based on questionnaires about the customer involvement, the team distribution and the team proficiency.

An Introduction to Ontology Engineering [Keet18]. This book presents a complete and illustrative reference guide about best practices for novice ontology engineers. It is not just a compilation of different methodologies, but it presents concrete examples and exercises to build quality ontologies. It addresses ontology engineering methodologies by distinguishing *macro-level development* from *micro-level development*. Macro-level development methodologies are mainly oriented to the development life cycle and process, e.g. *Methontology* for non collaborative scenarios of

[8] This strategy consists on defining concepts at an intermediate granularity level, i.e. neither very general nor very specific [Uschold96].

a single ontology, and the *NeOn* methodology that "replaces" Methontology including scenarios of ontology integration and a collaborative process model. Micro-level development methodologies are about how to go from an informal ontology representation to a logic-based one. In this sense, this work references the methodologies OntoSpec, OD101, and DiDOn which address ontology engineering aspects such as requirements analysis, ontology architecture (e.g. modular ontologies, distributed ontologies), representation languages, how to choose a foundational ontology, modelling decisions, and the reuse of domain ontologies, design patterns and matching techniques for ontology alignment among others [Kassel05, Keet12, Noy01].

To guide ontology engineers in developing quality ontologies, this book introduces different methods, (i) *logic-based* methods, with focus on the results given by reasoners about inconsistencies and non desirable inferences, (ii) *based on philosophy* methods, e.g. OntoClean, (iii) *a combination of the two*, e.g. the RBox compatibility service for coherent hierarchies and role chains of object properties [Keet08, Keet12], and (iv) practical guidelines, antipatterns and TIPS [Keet13]. Some TIPS are about how to express in description logics requirements written in natural language, and also about the identification and naming of classes in ontologies. For the latter, a recommendation is "avoid synonymy and polysemy: distinguish the concept itself from the different names that a concept can have (the synonyms) and create just one class for the concept and provide, if needed, different names for such a class using rdfs:label annotations". Chapter 4 of the present work turns out on this topic when addressing semantics and case studies of meta-modelling.

This book classifies ontology development approaches into *top-down* and *bottom-up* approaches. Top-down development addresses the use of foundational ontologies such as DOLCE and BFO [DOLCE, BFO]. Foundational ontologies were built based on philosophical principles, such as the endurantism, to represent objects that persist in time and the perdurantism, for objects that unfold in time. Another issue is the distinction between class and individual from a philosophical point of view: "ontologically, instances/individuals/particulars are, roughly, those things that cannot be instantiated, whereas classes (or universals or concepts) can..". This topic is also revisited in chapters 3 and 4 of the present thesis work, in particular regarding the use of individuals for the representation of a domain point of view or "perspective" in the context of meta-modelling. Moreover, this book addresses how to represent general or domain-independent categories of things (e.g. physical objects), hierarchies of categories, relations (e.g. part-whole) and attributes. Among the benefits of using foundational ontologies, the author mentions the reuse of principled design decisions that facilitates interoperability between ontologies and prevents novice ontology developers from making mistakes. Regarding bottom-up development, the book provides some guidelines about how to construct an ontology (in particular the Tbox) from some more or less structured information, both manually or (semi) automatically. Finally, *ontology design patterns* are defined as "a middle out way for developing ontologies", because they are in the middle of top-down development (viewed as foundational ontology fragments that serve for good modelling practices) and bottom-up development (different design solutions can be combined for solving some modelling aspect).

2.4.2 Methodologies for meta-modelling approaches

The review of the literature done in the present work shows that there is little work about design patterns for meta-modelling approaches such as the ones described in Section 2.3. The present section describes some related wok about meta-modelling design patterns and antipatterns that correspond to conceptual oriented meta-modelling approaches such as the ones described in Section 2.4.1, for single level ontologies. Like them, the design patterns for meta-modelling described below do not specify which is the adopted semantics for a given meta-modelling extension to OWL (see meta-modelling approaches presented in Section 2.3).

De Lara et al. present a set of meta-modelling design patterns, and describe them with UML extended with special constructors to model more than one knowledge level [DeLara14]. Basically, they address the problem of dinamically create object types that both have instances and are instances of other types. The authors introduce the meta-modelling level as the notion of *potency*. A *potency* can be attached to models, or elements of models. In a multi-level model, there are three kind of elements: (i) elements with the greatest potency n are types (or classes) than can be instantiated, (ii) elements in the intermediate potencies i, $0 < i < n$, are called *Clabjects*; they can be instantiated and can also be instances of types in the immediate upper level, and (iii) elements of the potency 0 can only be instances of types in the immediate upper level. Moreover, the authors distinguish two orthogonal typings for elements of the model: *ontological* and *linguistic*. The ontological classification of an element expresses instantiation within a domain whereas the linguistic typing of an element refers to it as an instance of an element of the language (as type, Clabject, inheritance or the instantiation itself, among others). Regarding the ontological typing, they address the problem of modelling a particular multi-level example with three knowledge levels: product types, products, and product instances. They present a set of meta-modelling design patterns. Some of them are really two-knowledge level models (class and instances), since they simulate the product type and product levels with inheritance or through a property. However, they also present a design pattern that indeed models three different knowledge levels by making use of potencies 0 to 2. Moreover, they analyze how to rearchitect a two-level solution into the more expressive multi-level solution and present some case studies that can be modelled with the multi-level approach.

Brasileiro et al. and Almeida et al. presents a set of *antipatterns* for meta-modelling ontology design [Brasileiro16, Almeida18]. They consider the notion of meta-modelling given by the MLT theory. Even though this notion is formally defined based on first order logic, it does not follow a description logics semantics (individuals interpreted as domain elements and concepts interpreted as domain subsets, as described in Section 2.3.1) [Carvalho18]. Antipatterns are illustrated by showing their occurrence in existing multi-level hierarchies of Wikidata and are assessed by making queries to Wikidata based on the MLT rules. Occurrences of several antipatterns are detected in Wikidata: (i) a class A is both a subclass of a class B and an instance of B which means that A and B are simultaneously at the same and at different levels, (ii) a class C is a subclass of two classes A and B, and A is an instance of B that means that C is simultaneously a subset of classes at different levels, and (iii) an element C (class or individual) is an instance of both classes A and B, and A is an instance of B that also means that C belongs simultaneously to classes at different levels.

2.5 Conclusions

This chapter introduces the foundations of ontologies and description logics, and related work on ontology networks, meta-modelling, and design patterns.

The analysis of related work on ontology networks is mainly focused on semantics, in particular how each approach defines and controls the relation between the interpretation of each ontology (or module) and the interpretation of the ontology network. Three main semantic approaches are identified: a *local* approach, with each networked ontology interpreted in a local domain, a *global* approach, where the ontology network is interpreted in a single domain, and an *hibrid* approach, with each ontology interpreted in a local domain and the ontology network in a single domain.

This chapter describes some meta-modelling approaches from the semantics point of view, within the perspective of combining independently built ontologies into ontology networks through different kind of relations (in particular, the meta-modelling relationship). Two main research directions on meta-modelling are identified: *conceptual modelling* oriented approaches that use ontologies to enhance the design of domains but does not extend description logics, and *description logics meta-modelling* approaches that extend description logics underlying OWL and also the reasoning algorithms. The latter approaches differ from each other basically in two complementary aspects of the semantics. On the one hand, some approaches follow a stronger Henkin style semantics whereas others follow a weaker Hilog semantics. On the other hand, some approaches define a more rigid fixed-layered domain of interpretation whereas others define a more flexible global-layered domain that enable different layers to interact with each other.

Finally, some methodologies and design patterns for ontology (network) engineering are presented. Some of them address the engineering process, and others, as the design patterns, help the ontology engineer design a given scenario and identify common design errors (antipatterns).

Two main conclusions emerge from the reviewed related work:

- There is no description logics meta-modelling approach with a Henkin and global-layered semantics. The intensional regularity and extensionality of the Henkin approach would have the benefit of ensuring that individuals and corresponding concepts keep equality relations, and the global domain would be simpler for the ontology engineer and more flexible regarding modelling choices.

- There is little work about design patterns for meta-modelling and in particular there is no design pattern that guides the ontology engineer on what is the most suitable description logics meta-modelling approach to model a real scenario. Some of the reasons why this happens could be: (i) nowadays meta-modelling approaches are not broadly used, and (ii) even though meta-modelling approaches provide high expressivity to model some scenarios, they also introduce some complexity for the ontology engineer (who often do not have enough expertise with these approaches). Note that anyway it is possible to model such scenarios without applying meta-modelling approaches, e.g. given a real object that is represented by an individual a and a concept A, the meta-modelling correspondence between them can be simulated by a fictitious role that connect instances of A to the individual a (see the approach of Glimm et al. in Section 2.3.2)

Chapter 3

Motivating case studies

As mentioned in chapters 1 and 2 the meta-modelling treated in the present work addresses the representation of several knowledge levels within a given domain or a particular application. The main focus of the present chapter is to discuss how to model a set of case studies which require to represent at least two abstraction levels, and evaluate the application of the main meta-modelling description logics approaches presented in Chapter 2. The conceptualized scenarios are about domains of geographic objects, health, education, accounting and recommendation of web contents [Motz14, Motz17, Rohrer19, Rohrer18, Rohrer10a, Rohrer10b, Rohrer10c, Rohrer11, Diaz12]. For these case studies, some other relations apart from meta-modelling are identified and mapped to existing OWL constructs [Diaz11, Rohrer14]. In the process of conceptualizing these domains the present work introduces the idea of *perspective* which results useful to communicate a notion that is not covered neither by the existing notion of ontology nor by the notion of meta-modelling level.

The chapter is structured as follows. Section 3.1 presents an intuitive definition of the main relations among ontologies that are identified in the described domains, and moreover introduces the notion of perspective. Sections 3.2, 3.3, 3.4 and 3.5 introduce real case studies about geographic, health, education and accounting domains, which are conceptualized by using meta-modelling. These sections also analyze the capability of the three main semantic approaches for meta-modelling *Henkin-fixed layered*, *Hilog-fixed layered* and *Hilog-global layered* to model the case studies. Section 3.6 describes a generic solution for a scenario of recommendation of contents about a given topic such as health or education. In Section 3.7 a different scenario of use of meta-modelling is briefly described with the aim of precisely delimiting the scope of the present work. Finally, Section 3.8 presents some conclusions.

3.1 Ontology relations and the perspectives

Given that case studies depicted below are taken from real scenarios, they are conceptualized as ontology networks that integrate some (views of) domains in different ways, i.e., besides meta-modelling, networked ontologies are also related through other kind of relations. Altogether, four different relations among ontologies are identified for the analyzed case studies: *mapping*, *link*, *extension* and *meta-modelling*, which are intuitively described as follows [1] [Diaz11, Rohrer14].

[1] For a formal definition of mapping, link and extension see [Rohrer14].

- A *mapping* relation between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ expresses a correspondence between concepts or individuals from $\mathcal{O}_1$ and $\mathcal{O}_2$ [Serafini09, Homola10a]. A mapping between a concept C of $\mathcal{O}_1$ and a concept D of $\mathcal{O}_2$ allows expressing that C and D are equivalent ($C \equiv D$), C is subsumed by D ($C \sqsubseteq D$), or that C and D are disjoint ($C \sqcap D \sqsubseteq \bot$). A mapping between individuals expresses that an individual a from $\mathcal{O}_1$ is the same as an individual b from $\mathcal{O}_2$ ($a = b$). Moreover, it is possible to map an individual a from $\mathcal{O}_1$ to a concept C from $\mathcal{O}_2$ by expressing that a is an instance of C ($C(a)$).

- A *link* relation between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ is given by a role R not belonging neither to $\mathcal{O}_1$ nor to $\mathcal{O}_2$, which connects individuals from $\mathcal{O}_1$ and $\mathcal{O}_2$ by axioms such as $R(a, b)$ with a from $\mathcal{O}_1$ and b from $\mathcal{O}_2$ or $C \sqsubseteq \exists R.D$ with C from $\mathcal{O}_1$ and D from $\mathcal{O}_2$ [Serafini09, Homola10a].

- An *extension* relation between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ expresses that the signature[2] and axioms of $\mathcal{O}_1$ are included in the ones of $\mathcal{O}_2$ [Alloca09]. For example, $\mathcal{O}_1 = \{C \sqsubseteq D, \exists R.A \sqsubseteq C\}$ and $\mathcal{O}_2 = \mathcal{O}_1 \cup \{C_1 \sqsubseteq C, C_2 \sqsubseteq C, \geq 2R.\top \sqsubseteq C_1\}$

- A *meta-modelling* relation between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ expresses a correspondence between an individual from $\mathcal{O}_1$ and a concept (a role) from $\mathcal{O}_2$, which means that the individual and the concept (role) are the same real object. The example of the Figure 1.1 in Chapter 1 shows that the individual *river* (instance of $NaturalResource$) from one ontology and the concept *River* from another ontology are the same real object represented with different granularity.

MOL formalisms and the modular reuse approach described in Chapter 2 address the first three relations. To ensure a "safe" interaction between ontologies, MOL approaches define a special syntax and semantics whereas the modular reuse approach defines some reasoning services [Grau07]. In particular, the mapping relation is addressed by the MOL formalisms DDL and IDDL, the link relation by ε-connections, and the extension relation by P-DL which allows reusing an ontology by extending it. However, it is important to observe that the expressivity of OWL is enough to solve the relations *mapping*, *link* and *extension*. In this sense, the modular reuse approach does not introduce a special syntax and adopts a semantics that interprets the union of axioms of the networked ontologies into a single domain. To ensure a safe interaction of networked ontologies, this approach defines new reasoning services on top of the existing standard reasoning.

Nevertheless, *meta-modelling* relations cannot be soundly expressed by OWL nor by MOL and the modular reuse approaches. Chapter 2 classifies the main description logics approaches that solve meta-modelling according to their style of semantics and compares them identifying a set of pros and cons. Then, next sections introduce different real scenarios and represent them by using the meta-modelling approaches described in Chapter 2. This allows visualizing the benefits and drawbacks of each approach for concrete domains.

[2]The signature of an ontology is the set of names to express its vocabulary.

Before describing the motivating case studies, the notion of *perspective* is introduced [3]. This notion helps in the understanding of the scenarios described below as well as in the use of meta-modeling for conceptualizing them.

In the definition below, the notion of *real object* infomally introduced before is assimilated to the general definition of *entity* that denotes a real-world object or concept [Elmasri15].

Definition 1 (Perspective) *Let* $\mathbf{I}$, $\mathbf{C}$ *and* $\mathbf{R}$ *be pairwise disjoint sets of individuals, atomic concepts and atomic roles respectively,* $\mathbb{O} = \{O_1, \ldots O_k\}$ *a set of ontologies,* $Q \subseteq \mathbf{I} \cup \mathbf{C} \cup \mathbf{R}$ *the set of individuals, concepts and roles belonging to* $\mathbb{O}$*, and* $E = \{e_1, \ldots e_m\}$ *a set of real objects.*
A perspective of E in $\mathbb{O}$ *is defined as a function* $p : E \rightarrow Q$ *from the set of real objects* E *to the set of individuals, concepts and roles in* $\mathbb{O}$*.*
For two different perspectives p_1*,* p_2 *of a set of real objects* E*, it holds that there exists at least one real object* $e \in E$ *such that* $p_1(e) \in \mathbf{I}$ *and* $p_2(e) \in \mathbf{C}$ *or viceversa.*

Note that for each real object a perspective determines its representation as an individual, a concept or a rol in a set of ontologies; and two different perspectives represent at least one real object with different granularity, as an individual and as a concept. Intuitively, this notion captures the granularity with which an actor perceives a set of real objects, it is why in the above definition each perspective is formalized as a function that maps real objects to individuals, concepts and roles. Recalling the example of Figure 1.1 in Chapter 1, rivers and lakes as instances (of the concept $NaturalResource$) represent the *perspective* of government managers whereas rivers and lakes as concepts represent the *perspective* of experts who control aquifer pollution. As is pointed out in Chapter 2, even though from a philosophical point of view an individual denotes "not instantiable objects", within a particular perspective (e.g. the one of the government) some object can be visualized and "managed" as an instance although it is instantiable from another perspective (e.g. for the acuifer expert). It is important to observe that a perspective of a set of real objects does not necessarily corresponds to a single ontology, i.e. two different perspectives can be represented within the same ontology or two different ontologies can represent the same perspective.

3.2 Geographic objects in Uruguay

There is a great diversity of geographic data as well as different standards to represent them. In most countries and in particular in Uruguay, geographic data are sparsed in different sources since their owners are organizations such as the Ministry of Transportation and Public Works, the Ministry of Livestock, Agriculture and Fisheries, the Military Geographical Service and the Municipal Administration, among others. In Uruguay, all that information is analyzed by working groups of the Uruguayan Spatial Data Infrastructure (IDE.uy). The Agency for the Development of the Government of Electronic Management and the Information and Knowledge Society (AGESIC) manages these groups with the aim of integrating the information provided by the geographic data producing institutions [IDEUY, AGESIC]. Main

[3] Even though "perspective" is a very broad term, in the present work it is defined in the context of meta-modelling.

contributions on geographic data has been provided by the Military Geographical Institute, which is a reference institution of the Uruguayan government in charge of managing and mantaining quality geographic information [IGM].

The work of Comesaña is a starting point for the analysis of the geographic domain in the present work [Comesania15]. Taking into account all the sources mentioned above and specially the information provided by the Military Geographical Institute, Comesaña has developed a complete conceptual model for different sub-domains on geography such as hydrography, physiography and flora, among others. A deep analysis of this work led to identify three different knowledge levels:

- the level that corresponds to end-users such as industries which make use of particular natural resources such as rivers (e.g. the Arapey river),
- the level associated to users who are experts in the geographic sub-domains, e.g. data producers about hydrography or flora, or academic users devoted to the scientific study of each sub-domain,
- the level that corresponds to government users (e.g. the Ministry of Livestock, Agriculture and Fisheries) who have a general view of different kind of geographic data and how they are related, and must take decisions for their preservation.

Regarding the last level of government institutions, even though they must be able to retrieve data about particular resources (e.g. the Arapey river) they mainly need aggregated information to define policies oriented to classes of resources such as all the hydrographic resources. For the level of expert users, experts on hydrography study mechanisms for preserving rivers which differ from the ones for preserving lakes, and also study the relations with other sub-domains such as flora (e.g. a natural forest on the banks of a river acts as a barrier against pollution). Hence, according to Definition 1 three different perspectives are identified: *government user perspective*, *expert user perspective* and *end-user perspective*. To satisfy the requirements of each user from his/her particular perspective, real objects *river* and *lake* seem to be better represented as individuals for the expert user perspective whereas they are better represented as concepts (with the sets of all Uruguayan rivers and lakes as instances) for the end-user perspective. The following subsections present a model of the domain that captures geographic objects for each user perspective, compares the capability of description logics meta-modelling approaches (presented in Chapter 2) for representing this model, and finally analyzes the use of meta-modelling for this case study.

3.2.1 Meta-modelling for geographic objects

Figure 3.1 shows an ontology network that conceptualizes a small fragment of the Uruguayan geographic domain [Motz14]. Objects conceptualized by Comesaña are modelled by distinguishing the three user perspectives which are delimited by horizontal lines in the figure, and ontologies in each perspective are delimited by vertical lines. *Link* relations between ontologies (e.g. *over*) are represented by thick arrows (thinest arrows represent roles within an ontology), and *meta-modelling* relations between ontologies are represented by dashed edges.

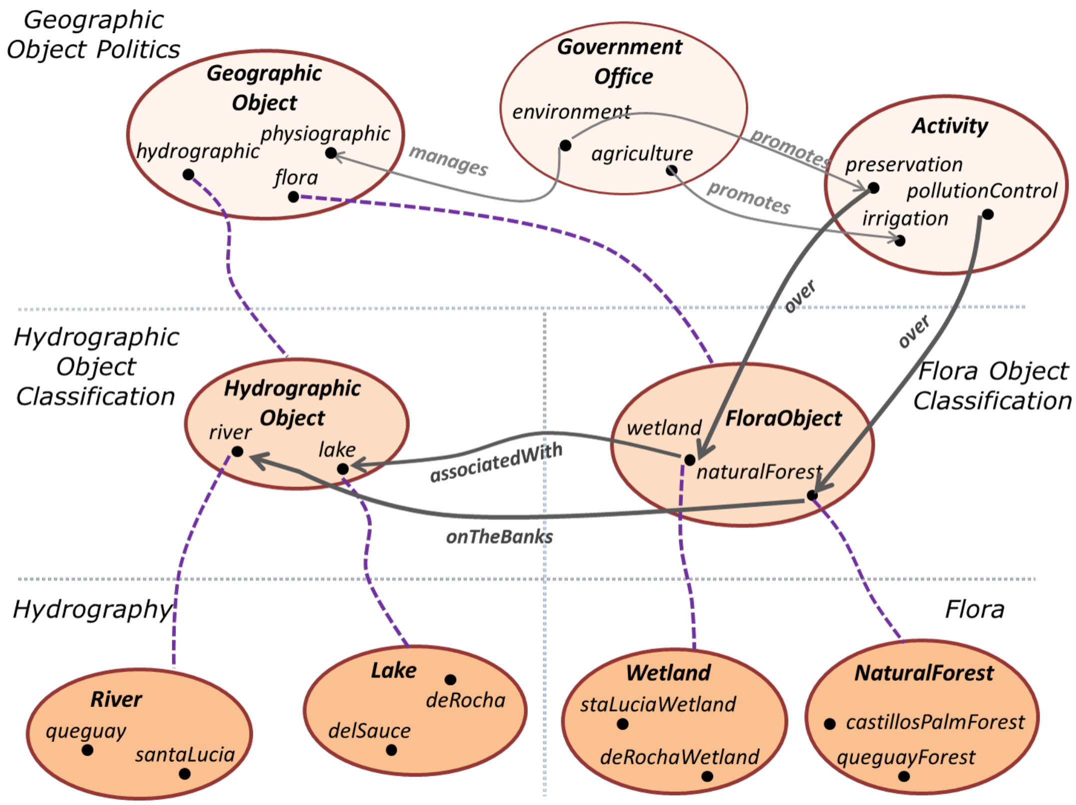

Figure 3.1: An example of the geographic domain

The *end-user perspective* is modelled by two ontologies at the bottom which conceptualize the concrete natural resources at a lower level of granularity with concepts *River*, *Lake*, *Wetland* and *NaturalForest*. The *expert user perspective* is modelled by two ontologies in the middle; the ontology in the left side describes hydrographic objects through the meta-concept *HydrographicObject* which contains individuals that are also concepts, and the ontology in the right side describes flora objects through the meta-concept *FloraObject*. The *government user perspective* is modelled by the ontology in the uppermost row which conceptualizes the policies about geographic objects. It defines the meta-meta concept *GeographicObject* that has instances which are meta-concepts, and also has the concepts *Activity* and *GovernmentOffice* with instances that are atomic objects.
Note that even though horizontal lines in Figure 3.1 delimit different user perspectives they do not represent the different meta-modelling levels for conceptualizing geographic objects. For instance, the upper ontology "Geographic Object Politics", that models the government perspective, has the meta meta-concept *GeographicObject* that corresponds to the higher abstraction level, but also has the concepts *Activity* and *GovernmentOffice* which are in the lower abstraction level. Moreover, the meta meta-concept *GeographicObject* has instances at different levels: the meta-concepts *hydrographic* and *flora* and the atomic object *physiographic*. Then, the perspective of a higher-level user will not necessarily have concepts that are all in the higher meta-modelling level. The meta-modelling hierarchy for the concepts of Figure 3.1 is depicted in Figure 3.2[4]. It is important to observe that the OWL

[4]It is missing a lower layer of atomic objects, it was intentionally excluded to simplify the picture.

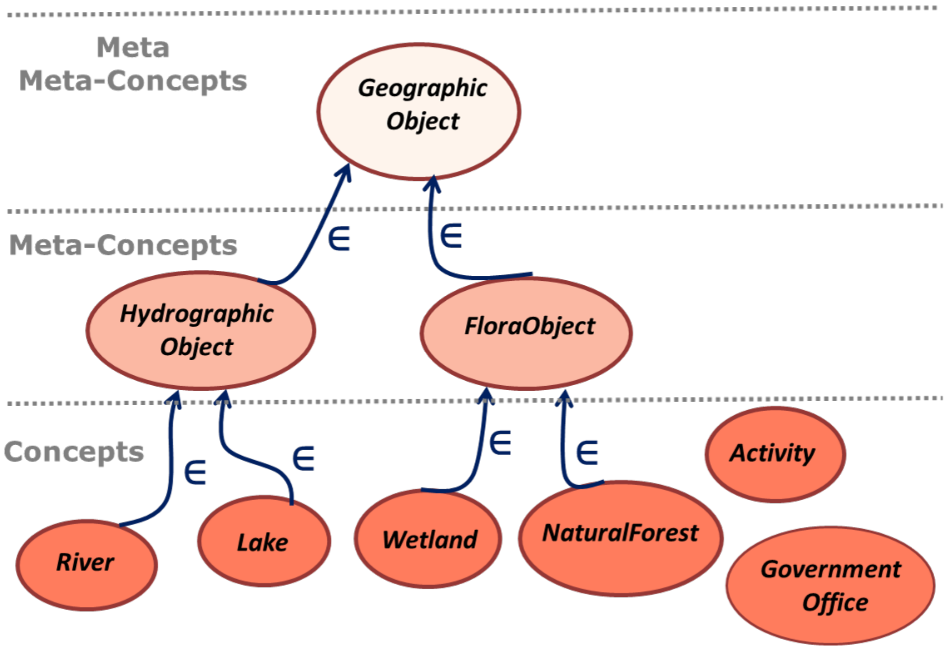

Figure 3.2: Meta-modelling hierarchy for the Ontology of Figure 3.1

language has only one notion of hierarchy which classifies concepts with respect to the inclusion $\sqsubseteq$. However, meta-modelling approaches also introduce the notion of *meta-modelling hierarchy*.

Figure 3.3 shows the Tbox, the Rbox and the Abox of the ontologies of Figure 3.1. However, Tbox, Rbox and Abox axioms are not enough to represent the meta-modelling relations given by dashed edges, so some description logics meta-modelling approach from the ones described in Chapter 2 is needed.

3.2.2 Comparison of meta-modelling approaches

In this section the three main semantic approaches for meta-modelling described in Section 2.3.2 are applied to represent the meta-modelling relations of de model of Figure 3.1. Besides analyzing the expressivity of *Henkin-fixed layered*, *Hilog-fixed layered* and *Hilog-global layered* approaches for representing the model, this section analizes the capability of each approach to prevent some inconsistencies due to meta-modelling and to ensure the well-foundedness of the interpretation domain (see Section 2.3 of Chapter 2).

Henkin-fixed layered approach. Since the interpretation domain is fixed-layered, concepts and roles must be labelled or "typed" with the meta-modelling level. Then, concepts of the level "Concepts" in Figure 3.2 are typed with 1 (level 0 corresponds to atomic objects), for instance $River^1$, and concepts in the level "Meta-Concepts" are typed with 2, for instance $HydrographicObject^2$. However, the concept *GeographicObject* in the "Meta Meta-Concepts" layer has two instances (*geographic* and *flora*) that belong to the level "Meta-Concepts" and one instance (*physiographic*) that is an atomic object (level 0). This causes that the concept *GeographicObject* cannot be represented due to the fixed-layer semantics restricts that concepts of each level have only instances of the immediate lower level, i.e. $\Delta_i \subseteq \mathcal{P}(\Delta_{i-1})$.

Tbox

$GovernmentOffice \sqsubseteq \exists manages.GeographicObject$
$Activity \sqsubseteq \forall over.(HydrographicObject \sqcup FloraObject)$
$FloraObject \sqsubseteq \forall associatedWith.HydrographicObject$
$River \sqcap Lake \sqsubseteq \bot$

Rbox

$onTheBanks \sqsubseteq associatedWith$

Abox

$GeographicObject(hydrographic)$	$GeographicObject(physiographic)$
$GeographicObject(flora)$	
$GovernmentOffice(environment)$	$GovernmentOffice(agriculture)$
$Activity(preservation)$	$Activity(irrigation)$
$manages(environment, physiographic)$	
$promotes(environment, preservation)$	$promotes(agriculture, irrigation)$
$HydrographicObject(river)$	$HydrographicObject(lake)$
$FloraObject(wetland)$	$FloraObject(grassland)$
$FloraObject(naturalForest)$	
$over(preservation, wetland)$	$over(irrigation, grassland)$
$associatedWith(wetland, lake)$	
$associatedWith(naturalForest, river)$	
$River(queguay)$	$River(santaLucia)$
$Lake(deRocha)$	$Lake(delSauce)$
$Wetland(staLuciaWetland)$	$Wetland(deRochaWetland)$
$NaturalForest(castillosPalmForest)$	$NaturalForest(queguayForest)$

Figure 3.3: Tbox, Rbox and Abox of the ontologies of Figure 3.1

Moreover, it is not allowed introducing a new concept by combining concepts of different layers, e.g. *Activity* $\sqcup$ *FloraObject*. Regarding inter-layer roles such as *over*, even though the approach of Pan does not allow expressing them, the approach of Homola does allow representing the role *over*, with $over^{\mathcal{I}} \subseteq \Delta_0 \times \Delta_1$. Roles within a meta-modelling level, such as *promotes* or *associatedWith*, can be represented, except for the role *manages* since its range is the concept *GeographicObject* which cannot be interpreted in a fixed-layered domain. The Henkin-fixed layered approach ensures the well-foundedness of the interpretation domain. For instance, the axiom $FloraObject^2 \sqsubseteq NaturalForest^1$ that combines different meta-modelling levels is not allowed. It would make the domain non well-founded since the concept *NaturalForest* is related by meta-modelling to an instance (*naturalForest*) of *FloraObject*, i.e. *NaturalForest* belongs to itself. Moreover, the fact that the Henkin semantics satisfies properties of intensional regularity and extensionality (see Section 2.3.2 of Chapter 2) prevents from a lot of inconsistencies such that the one illustrated in the example below.

Example 1. Suppose the ontology network of Figure 3.1 with the Tbox and Abox of Figure 3.3 extended with the two axioms:

$$Wetland \equiv NaturalForest$$
$$FloraObject \sqsubseteq \leq 1 associatedWith.HydrographicObject$$

Note that the second axiom expresses that the role *associatedWith* is functional. If $\mathcal{I}$ is an interpretation of the ontology, then $Wetland^{\mathcal{I}} = NaturalForest^{\mathcal{I}}$ must hold, and by the property of extensionality $wetland^{\mathcal{I}} = naturalForest^{\mathcal{I}}$. It follows from the fact that *onTheBanks* is a subrole of *associatedWith* and the functionality of the role *associatedWith* that $river^{\mathcal{I}} = lake^{\mathcal{I}}$. Then, by intensional regularity the interpretations of their corresponding concepts by meta-modelling must also be equal, i.e., $River^{\mathcal{I}} = Lake^{\mathcal{I}}$. But in this point the ontology becomes inconsistent because the sets $River^{\mathcal{I}}$ and $Lake^{\mathcal{I}}$ are disjoint as well as non-empty (see Figure 3.3).

. . . .

Hilog-fixed layered approach. Again, as the interpretation domain is fixed-layered, concepts and roles must be "typed" with the meta-modelling level, and *GeographicObject* cannot be interpreted following a fixed layered semantics. The same as for Henkin-fixed layered approaches occurs about that it is not possible to define a concept by combining concepts of different layers and with respect to the definition of roles (both inter-layer roles and within a level). The well-foundedness of the interpretation domain is also ensured since it is a fixed-layered approach. However, since Hilog semantics does not satisfy extensionality, there are inconsistencies from meta-modelling such as the one of Example 1 which are not detected. In the example, $wetland^{\mathcal{I}} = naturalForest^{\mathcal{I}}$ does not hold due to the lack of extensionality, then the inconsistency is not detected. Note that it is important to detect this kind of inconsistencies in the context of use of meta-modelling of the present work. The key point is that the same real objects are viewed with different granularity, depending on the particular perspective, i.e. if sets *Wetland* and *NaturalForest* are equivalent for the end-user, from the expert user perspective *wetland* and *naturalForest* correspond to the same individual.

Hilog-global layered approach. Due to the interpretation domain is global-layered, concepts and roles must not be "typed" with the meta-modelling level, and all the concepts can be represented since it is not an approach of fixed layers. In particular, *GeographicObject* can be interpreted even though it has instances of different levels. Moreover, it is possible to define a concept by combining concepts of different meta-modelling levels and the definition of roles with domain and range at different levels is allowed. However, the semantics of the (global) domain does not ensure well-foundedness, i.e. axioms such as $FloraObject \sqsubseteq NaturalForest$ can be introduced without detecting any inconsistency. Again, since Hilog does not satisfy extensionality, a lot of inconsistencies such as the one of Example 1 are not detected.

3.2.3 Conclusions about meta-modelling for geography

A first consideration about using meta-modelling to represent geographic data is the capability that the model provides to aggregate the information taking into account the requirements and points of view of different levels of users, named "perspectives" in the present work. Moreover, the representation of sets of data as instances, besides capturing the vision of higher level users, is a flexible mechanism for changing the structural elements of the ontology (e.g. concepts) by treating them as “data”.

Another characteristic that is desirable to conceptualize is about that instances of roles that connects individuals with meta-modelling at a higher level should be translated as relations between corresponding concepts at the lower level. In the model elaborated by Comesaña instances of the role *associatedWith* connect individuals at the lower level such as *deRochaWetland* and *deRocha* (since it is a single-level model). However, the model of Figure 3.1 conceptualizes this role at the expert user perspective since experts on geographic domains are who identify these kind of relations. Suppose the expert user identifies that wetlands are associated with lakes and rivers but with no other hidrographic object (e.g. cutwaters). Then, this restriction should be transferred to a role defined at the lower level by ensuring that individuals in the set of wetlands (e.g. *staLuciaWetland*) are related by that role to rivers (e.g. *santaLucia*) or lakes but not to other kind of hidrographic objects. This correspondence between relations of different levels is addressed in deph for case studies about education and accounting in the present chapter.

3.3 Health

There have been many attempts to build taxonomies that capture the agreed terminology related to the health domain. Among them, UMLS is an effort of integrating different vocabularies about health subdomains [UMLS]. It centralizes and makes available taxonomies such as RXNorm for clinical drugs, NCI for the cancer terminology and SNOMED-CT [RxNorm, NCI, SNOMED]. SNOMED-CT covers a wide range of concepts in the health domain, such as diseases, diagnostics and treatments [SNOMEDGuide]. In particular, this section addresses the scenario of integrating the SNOMED-CT vocabulary with electronic health records of patients.

Given that SNOMED-CT is a taxonomy with a great amount of concepts, it has been formalized with the OWL 2 profile $\mathcal{OWL}$-$\mathcal{EL}$, which is a lightweight description logic with some modelling restrictions that improve reasoning efficiency and scalability [Krotzsch12, Suntisrivaraporn07]. The $\mathcal{OWL}$-$\mathcal{EL}$ SNOMED-CT ontology has concepts, such as $Disease$ that subsumes $HeartDisease$ and $Endocarditis$, and roles, such as $findingSite$ that represents where diseases are located. However, it has not individuals. As Figure 3.4 illustrates, SNOMED-CT is mostly used as a standard vocabulary to be referenced in electronic health records of patients [El-Sappagh14, Koopman12, Rector09, Schulz11].

Different authors have identified structural problems in SNOMED-CT. Schulz et al. find that from an ontological point of view, the SNOMED-CT concept hierarchy is overloaded since some SNOMED-CT concepts are in fact meta-concepts, individuals or roles [Schulz09]. Rector et. al. analyze a mechanism for using generic information models such us HL7 RIM or OpenEHR to represent electronic healthcare records which reference different vocabularies such as SNOMED-CT or ICD [ICD, Rector09, HL7SNOMED]. In particular, the authors mention the ”coding system of the ontology” that “should be a meta model of the model of meaning” where each SNOMED or ICD code is associated to a concept.

Regarding the practical use of the SNOMED-CT ontology, in most scenarios, a data repository of electronic health records of patients referencing SNOMED-CT terms is modelled with individuals that represent such references and instantiate the

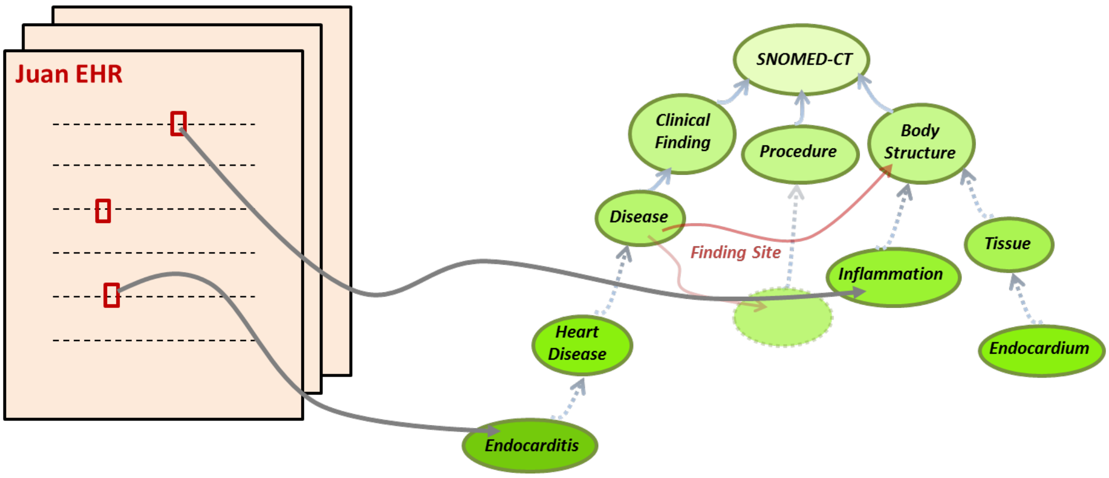

Figure 3.4: Electronic health records referencing SNOMED-CT

SNOMED-CT concepts [HL7SNOMED, Rector09]. Figure 3.5 illustrates the electronic health records of two patients *Juan* and *Pedro*, who suffer endocarditis. Individuals *juanEHRendocarditis*, *juanEHRendocardium* and *juanEHRinflammation*, instances of concepts *Endocarditis*, *Endocardium* and *Inflammation*, are just the references to these concepts in the electronic health records of Juan, and are linked through SNOMED-CT roles such as *findingSite* and *associatedMorphology*.

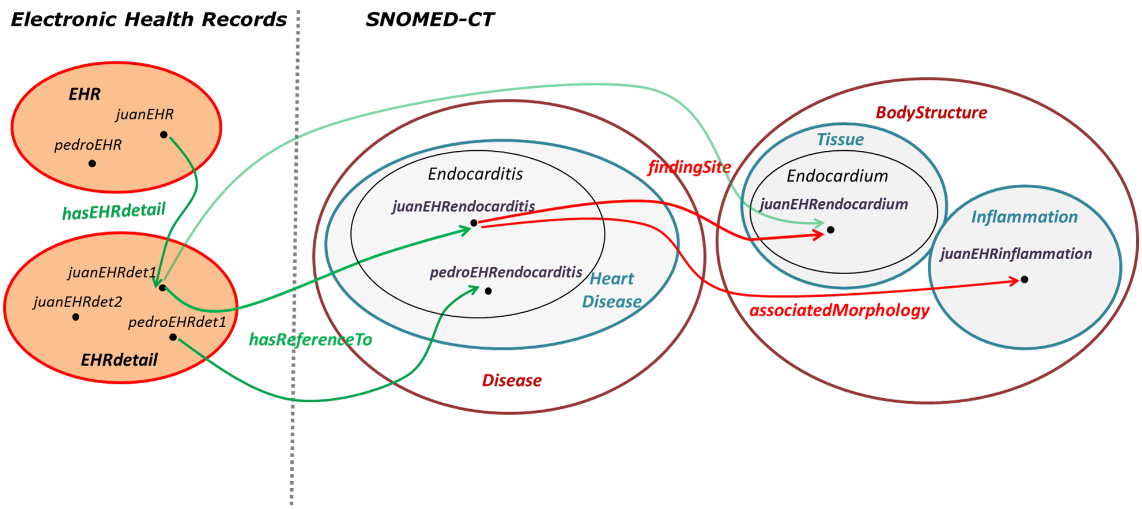

Figure 3.5: Ontology that integrates electronic health records and SNOMED-CT

In this scenario, Schulz et al. analyze SNOMED-CT complex concepts such as the following [Schulz08]:

$$\begin{aligned} ExtrOfForeignBodyFromStomachByExcision \equiv \\ \exists hasPart(\exists procedureSite.StomachStructure \sqcap \exists method.IncisionAction) \sqcap \\ \exists hasPart(\exists procedureSite.StomachStructure \sqcap \\ \exists directMorphology.ForeignBody \sqcap \exists method.RemovalAction) \end{aligned} \tag{3.1}$$

They observe that instances of *StomachStructure* in the first existential can be different from instances of *StomachStructure* in the second existential, and that these "stomachs" can belong to different patients. However, they do not give solution to this kind of misinterpretations within $\mathcal{OWL}$-$\mathcal{EL}$, i.e. without increasing the language expressivity. Recently, El-Sappagh et al. propose an upper-level ontology based on the Ontology for General Medical Science (OGMS) as the basis for defining the terms of SNOMED-CT and also identify some redundancies in the vocabulary [El-Sappagh18, OGMS].

The problems mentioned above with respect to the model illustrated in Figure 3.5 can be summarized in two main drawbacks: (i) the general knowledge of the health domain is represented at the level of each patient and (ii) definitions of SNOMED-CT does not detect some inconsistencies in electronic health records. The simplified description of the concept *Endocarditis* presented below illustrates the first problem.

$$\begin{aligned} Endocarditis \sqsubseteq & \exists findingSite.Endocardium \sqcap \\ & \exists associatedMorphology.Inflammation \end{aligned} \tag{3.2}$$

For the scenario of Figure 3.5, patients Juan and Pedro have references ($juanEHRendocarditis$ and $pedroEHRendocarditis$) to the concept $Endocarditis$, then the TBox axiom (3.2) is consistent with the following ABox axioms:

$$\begin{aligned} findingSite(juanEHRendocarditis, juanEHRendocardium) \\ findingSite(pedroEHRendocarditis, pedroEHRendocardium) \end{aligned} \tag{3.3}$$

Since the disease endocarditis will always be located in the endocardium, having these assertions at the level of each patient does not add any value. It is general knowledge of the health domain which does not differ for each patient.

With respect to the second problem mentioned above, it is interesting to see that the TBox axiom (3.2) also admits some extensions as the assertions given below, that are illustrated in Figure 3.6.

$$\begin{aligned} findingSite(juanEHRendocarditis, juanEHRendocardium) \\ findingSite(pedroEHRendocarditis, pedroEHRendocardium) \\ findingSite(pedroEHRendocarditis, juanEHRendocardium) \end{aligned} \tag{3.4}$$

Even though the knowledge base is consistent, it does not represent a real situation, because Pedro suffers endocarditis located in the endocardium of Juan!! Moreover, with such inconsistencies some frequent queries about records of patients can return invalid results. For instance, to obtain a chronological report about all clinical situations that affected the Pedro's endocardium, the query below returns the instances $juanEHRendocardium$ and $pedroEHRendocardium$.

$$\begin{aligned} q(z) = & \exists x, y.hasEHRdetail(pedroEHR, x) \wedge hasReferenceTo(x, y) \\ & \wedge findingSite(y, z) \wedge Endocardium(z) \end{aligned} \tag{3.5}$$

To restrict that references to SNOMED-CT concepts are linked for the same patients, a more expressive description logic with inverse roles and cardinality restrictions is needed. However, for a large knowledge base such as SNOMED-CT it would mean an important loss of efficiency.

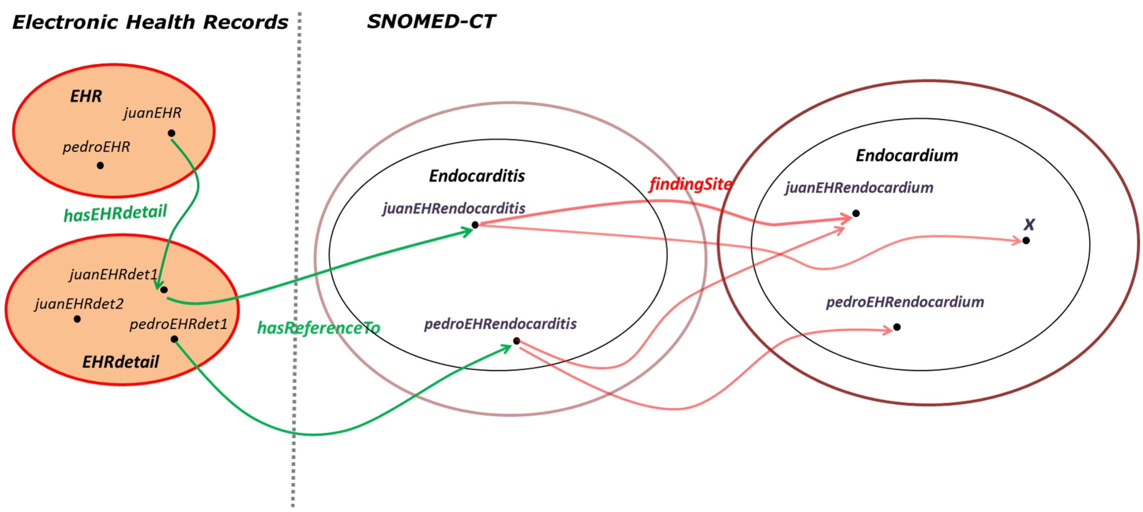

Figure 3.6: An extension for the definition of the concept *Endocarditis*

3.3.1 Meta-modelling in the health domain

Some solutions for solving the above problems have been presented such as the one of Schulz et al. who propose to modify the structure of SNOMED-CT, or to represent SNOMED-CT in a logic more expressive than $\mathcal{OWL}$-$\mathcal{EL}$. However, as SNOMED-CT is nowadays broadly used, to change its structure does not seem a realistic approach. Instead, the present work proposes adding an upper layer to the SNOMED-CT ontology. This layer represents *SNOMED-CT concepts as individuals* such that concepts and corresponding individuals *represent the same real objects*, i.e. they are related by meta-modelling [Motz17]. Moreover, instead of having references to diseases as instances of SNOMED-CT concepts, this proposal consist on *linking instances of electronic health records directly to the SNOMED-CT terms treated as individuals in the upper layer.* This new layer is a set of ABox axioms that represents general relations in the health domain, and is independent from the records of patients. For each TBox axiom containing an existential restriction, such as the description (3.2), an ABox axiom is added that connects SNOMED-CT concept names treated as individuals through the SNOMED-CT roles, and then records of patients are linked to the SNOMED-CT individuals. The following Abox axioms give solution to the scenario of Figure 3.5, illustrated in Figure 3.7:

$$\begin{aligned} hasEHRdetail(juanEHR, juanEHRdet1) \\ hasReferenceTo(juanEHRdet1, Endocarditis) \\ findingSite(Endocarditis, Endocardium) \\ associatedMorphology(Endocarditis, Inflammation) \end{aligned} \tag{3.6}$$

With this proposal, the solution of having instances of SNOMED-CT concepts such as $juanEHRendocarditis$ and $juanEHRendocardium$ connected by SNOMED-CT roles becomes unnecessary because now this kind of information is represented at the level of the meta-model layer. The application of meta-modelling to SNOMED-CT as described above has some advantages which are summarized below.

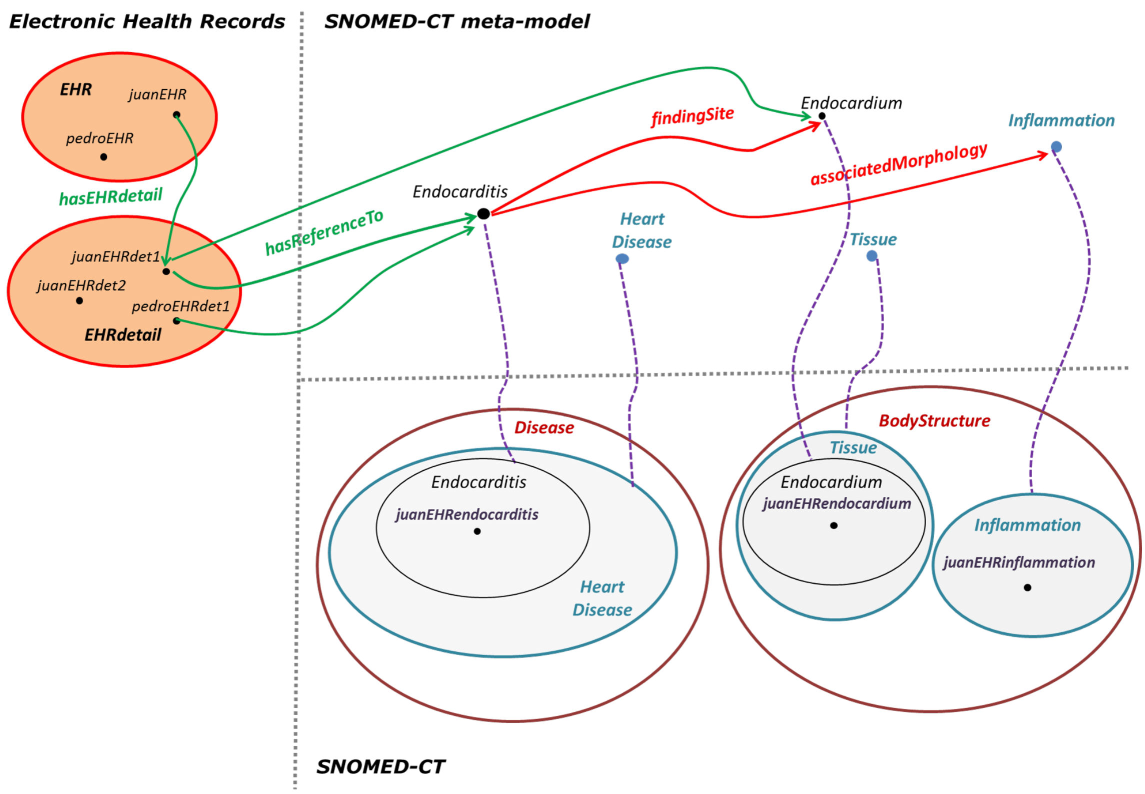

Figure 3.7: SNOMED CT with the meta-model layer

The proposed approach represents *the health domain through two different layers with different purposes*. In the lower level the SNOMED-CT ontology represents the hierarchy of medical terms whereas the upper level is defined to represent the relations between such medical terms and link electronic health records of patients to them. As illustrated in Figure 3.7, for the patient Juan there is a record represented by the individual $juanEHRdet1$ that is linked to the name *Endocarditis* treated as individual. Hence, *patients are related to medical terms at the proper level*. The upper layer is connected to the lower layer by meta-modelling, i.e. a given a term of the health domain is represented as an individual and as a concept. In Figure 3.7, the individual *Endocarditis* and the concept *Endocarditis* represent the same real object.

By representing medical terms as individuals, *redundancy of SNOMED-CT role instances is avoided*, e.g. Abox axioms (3.3). By contrast, ABox axioms such that $findingSite(Endocarditis, Endocardium)$ connects the medical terms *Endocarditis* and *Endocardium* in the meta-model, but not at the level of patients.

Invalid extensions of SNOMED-CT concepts are prevented. The proposed solution avoids extensions of the TBox axiom (3.2) such as (3.4) because records of patients are directly connected to SNOMED-CT terms in the meta-model whereas the lower layer describes the hierarchy of the vocabulary. Hence, with this approach the query 3.5 can be solved at the upper level as is showed below.

$$\begin{aligned} q(x) = & \exists y.hasEHRdetail(pedroEHR, x) \wedge hasReferenceTo(x, y) \wedge \\ & findingSite(y, Endocardium) \end{aligned} \tag{3.7}$$

The fact that SNOMED-CT terms can be treated either as individuals or concepts allows to infer useful information. For instance, it is useful to get inferences like "if Juan has endocarditis then Juan has a heart disease". The following query returns all patients that suffer a heart disease:

$$q(x) = \exists y, z.hasEHRdetail(x, y) \land hasReferenceTo(y, z) \land z \sqsubseteq HeartDisease \quad (3.8)$$

The set of solutions of the above query is the set of electronic health records of patients that reference SNOMED-CT individuals which treated as concepts are subsumed by the concept $HeartDisease$.

3.3.2 Comparison of meta-modelling approaches

Taking into account the proposed approach, an analysis of the differences among *Henkin-fixed layered*, *Hilog-fixed layered* and *Hilog-global layered* styles of semantics shows that main problems appear in approaches with either global layered or hilog semantics. Hence, unlike the geographic domain, below it is contrasted global to fixed layered and Henkin to Hilog semantics to better visualize the drawbacks.

Fixed vs global layered approaches. Even though fixed layered semantics forces to "type" concepts and roles with the meta-modelling level, in this particular scenario the SNOMED-CT hierarchy of concepts is only visualized as one level of knowledge and from a single perspective. Hence, all concepts would be typed with 1, which has no difference with not typing concepts, as in a global semantics. Then, the three approaches allow to represent meta-modelling in this scenario. However, as in the geographic domain global approaches do not ensure the well-foundedness of the interpretation domain as the example below shows.

Example 1. Suppose the axioms below are added to the meta-model layer of Figure 3.7.

$$DiseaseObject(Endocarditis)$$
$$DiseaseObject \sqsubseteq Endocarditis$$

Note that the second axiom is allowed in the global approach. The interpretation domain becomes non well-founded since *Endocarditis* belongs to *DiseaseObject* which is a subset of *Endocarditis*.

. . . .

Henkin vs Hilog approaches. Regarding the Henkin or Hilog styles of semantics, again the Hilog approach does not detect some inconsistencies such as the one illustrated in the example below.

Example 2. Suppose in the scenario of Figure 3.7 the SNOMED-CT term *Arrythmia* which is a concept in the lower level and is an individual in the upper level. Suppose that there exist the following axioms in the lower level and in the upper level respectively.

$$Arrythmia \sqsubseteq HeartDisease$$
$$Arrythmia \neq Endocarditis$$

If someone introduces *Arrythmia* $\equiv$ *Endocarditis* in the lower level, since the Hilog semantics does not satisfy extensionality *Arrythmia* and *Endocarditis* keep different in the upper level without detecting any inconsistency. So, in the electronic health records *Juan* has endocarditis but not arrythmia which is not coherent with the hierarchy of SNOMED-CT terms.

. . . .

The comparison of meta-modelling approaches for the health case study leads to conclude that: (i) the *Henkin-fixed layered* semantics allows to represent the scenario without inconsistencies, (ii) the *Hilog-fixed layered* semantics does not detect inconsistencies such as the previous example about *Arrythmia* and *Endocarditis*, and (iii) regarding the *Hilog-global layered* semantics, neither detects this kind of inconsistencies nor ensures the well-foundedness of the interpretation domain.

3.4 Education

The case study introduced in this section refers to an educational real-world escenario: the project DIIA at the public university of Uruguay [5]. The project DIIA is about the management of learning activities such as modules, workshops, and conferences, and the associated services such as work environments (learning platforms and classrooms) and equipment. Moreover, an important aspect is to capture the interaction of students with different work environments in all degree modules. Below the main requirements of the scenario are described, associating them to three different levels of users: *institution*, *teachers* and *students*.

The *institution* defines all possible learning activities that can be developed in the university and assigns the services that can be used in each activity. For instance, for learning modules the institution enables the use of services such as equipment and work environments. If the activity is a conference, it is also allowed to hire a catering service, which is not allowed for a module. According to different factors (as the economical or social policy, or change of authorities) services assigned to activities can vary over the time, although at least one service must be enabled so that the activity can work.

For each module of the degree structure, every year *teachers* are in charge of defining what particular services (within those enabled by the institution) they will use to develop the module, as well as for other activities such as workshops or conferences. For example, if modules are enabled to use work environments, the teacher of the module of Basic Programming can decide to use two work environments: the classroom (for a face-to-face teaching of de module) and the web platform, and the teacher of Data Base Foundations can define to use the web platform and the computer laboratory. These decisions can vary in each edition of the same module depending on factors such as the number of enrolled students or the physical space in classrooms. However, they only can take such decisions for the services enabled by the institution, which also can change over the time. However, it is a policy of the university that at least two different work environments must be available for all modules.

[5] ANII research project FSED_2_2016_1_130712 - Descubrimiento de Interacciones que Impactan en el Aprendizaje - Creación de un ambiente de software para descubrir patrones semánticos de interacción.

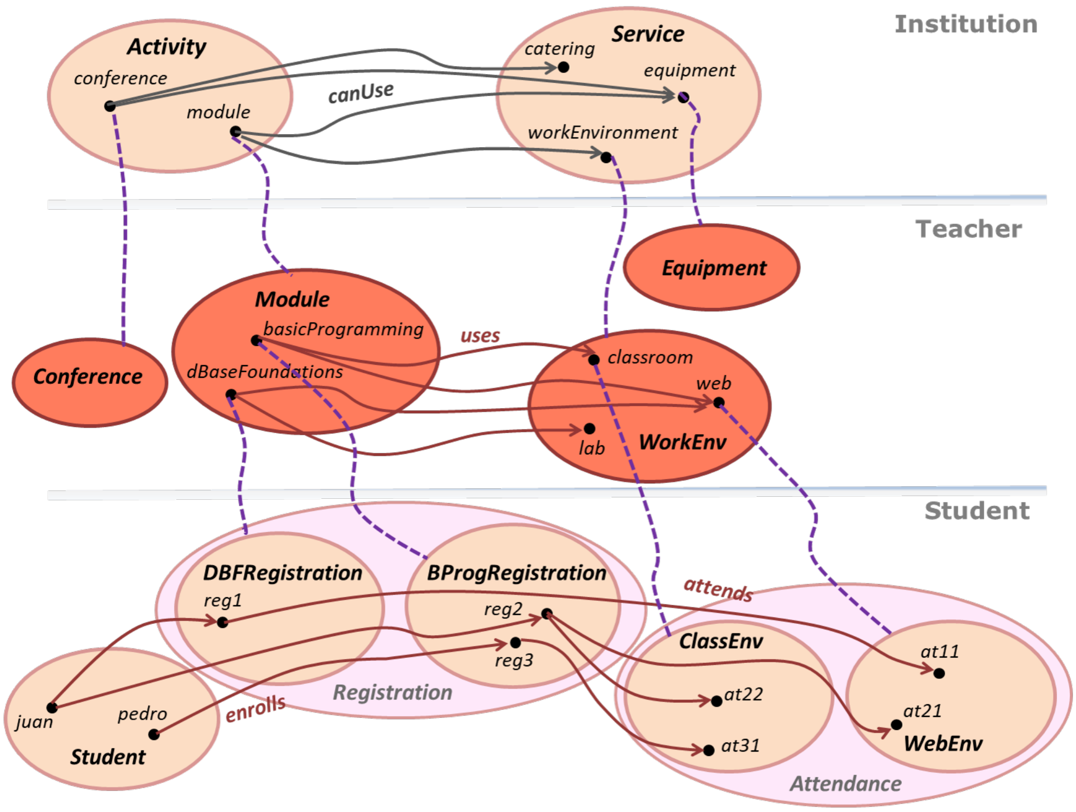

Figure 3.8: Example of educational model

Basically, *students* enroll in different modules, and for each module they must attend at least one of the work environments enabled by the teacher.

Two key points are identified in the scenario described above:

- the *three levels of users* with different requirements have *different perspectives* of the scenario described above (see Definition 1), since each level of user visualizes learning activities and associated services with different granularity, e.g. the institution visualizes a module as an activity whereas teachers visualize the particular modules they teach, and
- decisions taken by each level of user depends on decisions taken by the immediate upper level of user, e.g. students can attend work environments enabled by the teacher.

3.4.1 Meta-modelling in the educational domain

Figure 3.8 illustrates a conceptualization of the educational scenario introduced above with three ontologies, one for each level of user [Rohrer19].

In the Institution level, each activity is represented by the individuals *module* and *conference* as instances of the concept *Activity*, whereas in the Teacher level activities are represented by concepts *Module* and *Conference*, with dotted lines that represent the meta-modelling relation between individuals and concepts. Note that defining *Module* as a subclass of *Activity* does not seem the most suitable solution, since it does not represent the Institution perspective of a module, which is perceived as an atomic object, i.e. as one of its activities[6]. Moreover, services are represented by the individuals *workEnvironment* and *equipment* in the Institution level, and by the concepts *WorkEnv* and *Equipment* at the Teacher level.

From the requirements of the case study some relations between instances of roles are identified. For instance, as the institution enable the use of work environments and equipments, the teacher of each module can decide to use some equipment or some of the available work environments but no other service. In particular, for the Institution level the individual *module* is linked to individuals *workEnvironment* and *equipment* by the role *canUse*. According to the requirements in the Teacher level individuals of the concept *Module* can only be related to individuals of the concepts *WorkEnv* and *Equipment* by the role *uses*.

Regarding Teacher and Student levels, Figure 3.8 shows that there are meta-modelling correspondences between individuals such as *dBaseFoundations* and *basicProgramming* and corresponding concepts *DBFRegistration* and *BProgRegistration*, and moreover instances of the role *uses* must correspond to sets of instances of the role *attends*. For example, instances of the concept *BProgRegistration* can only be related to instances of *ClassEnv* and *WebEnv*.

Table 3.1 shows the Tbox axioms that express some requirements of the educational scenario but are not deduced from the Figure 3.8, for each level of user. As Tbox and Abox axioms are not enough to represent neither meta-modelling relations between individuals and concepts nor correspondences between roles of different levels (e.g. *canUse* and *uses*), some description logics meta-modelling approach from the ones described in Chapter 2 is required.

Axiom	Description
(1) $Activity \sqsubseteq \exists canUse.Service$	All activities have enabled at least one service.
(2) $Module \sqsubseteq \geq 2 uses.WorkEnv$	All modules use at least two different environments.
(3) $Registration \sqsubseteq \exists attends.Attendance$	All students attends at least one work environment for each module they are enrolled.

Table 3.1: Educational domain Tbox.

Applying meta-modelling to the educational case study has some advantages. On the one hand, the kind of requirements for the Teacher level such as at least two different environments must be available for each module is represented by the Tbox axiom (2) in Table 3.1. On the other hand, the kind of requirements such as modules (and other activities) can use the services enabled by the institution, that can vary over the time, are represented by Abox axioms such as $canUse(module, workEnvironment)$ in the Institution level. Then, by using meta-modelling relations, the solution of Figure 3.8 must ensure that Abox assertions $canUse(module, equipment)$, $canUse(module, workEnvironment)$ are transferred to the level of teachers as the Tbox axiom $Module \sqsubseteq \forall uses.(WorkEnv \sqcup Equipment)$.

[6]Solutions without meta-modelling were thought but they do not really represent different perspectives of the same real object, e.g. using a fictitious role to simulate meta-modelling correspondences.

This Tbox axiom restricts that all instances of *Module* can only be related to individuals of the concepts *WorkEnv* and *Equipment*. The key point here is avoiding the engineer to declare Tbox axioms at the Teacher level which are dynamic and depends on Abox axioms at the level of the Institution.

Note that as in the case study about geographic objects described in Section 3.2 horizontal lines in Figure 3.8 delimit different user perspectives (of the Institution, teacher and student users) but they do not represent the different meta-modelling levels. For instance, the upper ontology that represents the Institution perspective has the meta meta-concept *Activity* with instances at different levels: the meta-concept *module* and the concept *conference*.

3.4.2 Comparison of meta-modelling approaches

This section analyzes the expressivity of approaches *Henkin-fixed layered*, *Hilog-fixed layered* and *Hilog-global layered* for representing the meta-modelling described for the educational case study. Moreover, it is analyzed how each approach prevents from inconsistencies and ensure the well-foundedness of the interpretation domain.

Henkin-fixed layered approach. Since the interpretation domain is fixed-layered, concepts *Conference*, *Equipment*, *Student*, *Registration* (with its subclasses) and *Attendance* (with its subclasses) are typed with the level 1, whereas concepts *Module* and *WorkEnv* are typed with 2. However, concepts *Activity* and *Service* have instances of different meta-modelling levels, i.e. $Conference^1$ and $Module^2$ belong to *Activity*, and $Equipment^1$ and $WorkEnv^2$ belong to *Service*. Hence, *Activity* and *Service* cannot be represented due to in a fixed-layer semantics concepts of a given level have only instances of the immediate lower level. Roles *enrolls*, *attends* and *uses* can be represented but the role *canUse* cannot since its domain and range (*Activity* and *Service*) cannot be typed. The well-foundedness of the interpretation domain is ensured since it is a fixed-layered approach and due to the extensionality of the Henkin semantics, inconsistencies such as adding the axiom $Conference \equiv Module$ when individuals *conference* and *module* are different can be detected. In this case study, rules defined by users at higher levels that are represented in the Abox by relations between instances with meta-modelling (e.g. work environments for modules are asigned by teachers) must be transferred to the Tbox as relations between corresponding concepts by meta-modelling (students enrolled in modules can only attend to work environments defined by teachers). This kind of rule transference between levels is not solved by Henkin-fixed layered approaches.

Hilog-fixed layered approach. As in the previous approach fixed-layered semantics does not allow typing concepts *Activity* and *Service* at the third level since they have instances belonging to the first level. The well-foundedness of the interpretation domain is also ensured since it is a fixed-layered approach. However, since Hilog semantics does not satisfy extensionality, it does not detects the inconsistency that must arise when having different individuals *conference* and *module*, someone introduces $Conference \equiv Module$. It is important to detect this kind of inconsistencies in this scenario since in the real world individuals in higher levels represent the same objects than corresponding concepts in lower levels. Again, rule transference between levels (as definitions of teachers which affect students) is not solved by Hilog-fixed layered approaches.

Hilog-global layered approach. Given that the interpretation domain is global-layered, concepts and roles can all be represented (they must not be typed). However, the definition of the (global) domain does not ensure well-foundedness, e.g. for axioms such as $Module \sqsubseteq BProgRegistration$ which means that *BProgRegistration* belongs to itself, no inconsistency is detected. By the lack of extensionality, the Hilog semantics do not detect inconsistencies that arise from introducing $Conference \equiv Module$ when individuals *conference* and *module* are different. Rule transference between levels is not solved by this meta-modelling approach neither.

3.4.3 Conclusions about meta-modelling for education

Modelling the case study about education described above with meta-modelling gives solution to requirements of different levels of users. Ontologies at different levels correspond to different perspectives, and as Definition 1 says, captures the perception of each user regarding the granularity of a set of real objects. Moreover, the meta-modelling approach captures the granularity of relations between objects, in particular rules defined by higher levels of users as instances of relations which constrain relations between concepts for lower levels of users.

3.5 Accounting

The scenario introduced in this section corresponds to a real implemented application, the accounting module of the "Integrated Rental Guarantee Management System" (SIGGA) at the General Accounting Agency of the Ministry of Economy and Finance in Uruguay, called SIGGA [7]. The underlying business of SIGGA is that Uruguayan government acts as a guarantor for employees who want to rent a property. Home rental contracts are signed between landlords and renters who are employees. The application helps manage renter payments, as salary discounts or direct cash payment, and the corresponding payments to landlords. The accounting module is in charge of recording the accounting entries for the business rules of SIGGA which are modelled by the relational scheme despicted in Figure 3.9.

From a conceptual point of view, there are two *perspectives* of the business according to Definition 1: the *definitional perspective* that corresponds to expert users on accounting who define what kind of accounting entry (with debit and credit accounts) must be done for each financial movement, and the *operational perspective* that corresponds to users that operate the application by registering concrete accounting entries. Users at the definitional perspective perceive the different kinds of accounting entries as atomic objects whereas users at the operational perspective perceive the sets of accounting entries of the same kind as concepts or classes. In the definitional perspective, the different kinds of accounting entries are called *entry definitions*, and are specified according to the business rules by a set of valid details at debit and credit. These details are called *detail definitions*, and are univocally

[7]Sistema Integrado de Gestión de Garantía de Alquileres (SIGGA), Contaduría General de la Nación, www.cgn.gub.uy.

EntryDefs

entryDef	description
10	Renter payment
20	Monthly calculation of rent
30	Home damage expenses

DetDefs

entryDef	detailDef	account	D/C
10	1	11111	'D'
10	2	11112	'D'
10	3	11211	'C'
10	4	11311	'C'
20	1	11211	'D'
20	2	11311	'D'
20	3	21111	'C'
30	1	12222	'D'
30	2	21111	'C'

Accounts

account	description
11111	Cash
11112	Bank
11211	Renter Debt
21111	Landlords
12222	Damage Expenses
11311	Renter Fee

Entries

entry	entryDef	date	Observations
250	20	01/12/2017	Juan Pérez calc. of rent
251	10	13/12/2017	María García payment
252	10	14/12/2017	Juan Pérez payment

Dets

entry	detail	entryDef	detailDef	amount
250	1	20	1	4,500
250	2	20	2	1,500
250	3	20	3	6,000
252	1	10	1	6,000
252	2	10	3	4,500
252	3	10	4	1,500

Table	Primary key	Foreign keys	Description
EntryDefs	entryDef		models accounting entry definitions
DetDefs	entryDef, detailDef	entryDef from EntryDefs account from Accounts	models detail definitions
Accounts	account		models accounts
Entries	entry, entryDef	entryDef from EntryDefs	models accounting entries
Dets	entry, detail, entryDef	entry, entryDef from Entries entryDef, detailDef from DetDefs	models details of accounting entries

Figure 3.9: Relational tables for the accounting application of SIGGA

associated to an *account* such as Cash or Renter Debt that represents the essence of the financial movement. For example, in the upper part of Figure 3.9 the entry definition 10 "Renter payment", has two detail definitions at debit for accounts Cash and Bank, and two detail definitions at credit for accounts Renter Debt and Renter Fee. Whereas, in the operational perspective, particular *accounting entries* with their *details* are registered. For example, Figure 3.9 shows the "Juan Perez payment" (entry 252) on 14/12/2017 of $6,000 in cash, for accounts "Renter Debt" ($4,500) and "Renter Fee" ($1,500), according to the entry definition 10.

A retrospective looking of the SIGGA application shows some drawbacks in the implemented solution mainly due to limitations in the expressivity of the relational model to represent some relevant requirements. To describe the identified problems and introduce the proposed ontology-based solution, in the following SIGGA requirements are classified in three groups: (i) functional requirements that are solved by the relational model, (ii) functional requirements that are not solved by the relational model but can be expressed by a single-level ontology (without meta-modelling), and (iii) functional and non functional requirements that are not solved by the relational model nor by a single-level ontology.

ReqGroup. 1. Functional requirements solved by the relational model. A relevant requirement in this group is that *accounting entries have details for accounts in accordance with the definitions of expert users*, i.e. definitions of different kind of accounting entries by expert users must hold for concrete accounting entries recorded during the operation of the application by operator users. For instance, each time a renter pays a debt in cash, the recorded accounting entry must have details for the account "Cash" at debit, and "Renter Debt" or "Renter Fee" at credit in accordance with the "Renter payment" entry definition. The structural restrictions of the relational model showed at the bottom of Figure 3.9 solve this kind of requirements.

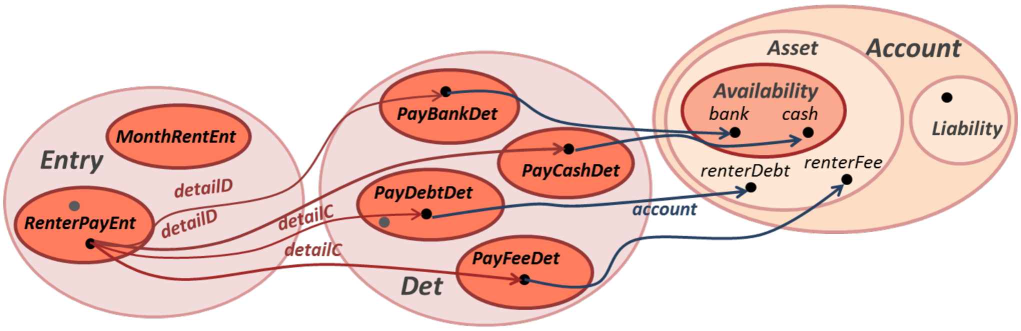

Figure 3.10: Example of SIGGA OM design

ReqGroup. 2. Functional requirements not solved by the relational model but expressed by a single-level ontology. For instance, it is required that each *concrete accounting entry has at least one debit detail and one credit detail*. The relational model of Figure 3.9 cannot express this requirement; instead, it can be solved by non-structural restrictions (i.e. only transactionally). However, the single-level ontology depicted in Figure 3.10, named SIGGA OM, with the Tbox presented in Table 3.2 can represent the requirement. In the SIGGA OM model subclasses of *Entry* represent all accounting entries of a given kind. According to definitions of expert users they have details at debit (by the role *detailD*) and credit (by the role *detailC*) that belong to a subclass of the concept *Det* and are associated to a given account (concept *Account*) at debit or credit by the role *account*. In particular, the axiom (1) of Table 3.2 expresses that each accounting entry (instance of *Entry*) has at least one debit detail and one credit detail, i.e. instances connected to each accounting entry by roles *detailD* and *detailC*. Note that the SIGGA OM ontology of Figure 3.10 also solves the requirement of the previous group about that accounting entries must follow expert definitions. In particular, axioms (3), (4) and (5) represent the definition of "Renter Payment" accounting entries, since details are restricted to the accounts at debit and credit defined by experts. For example, each renter payment must have a debit detail for the account "Cash" or "Bank" and credit details for accounts "Renter Debt" or "Renter Fee". However, axioms (3), (4) and (5) represent both the definitional perspective and the operational perspective and have to be declared for each kind of accounting entry such as definitions 10, 20 and 30 in Figure 3.9.

ReqGroup. 3. Requirements not solved by the relational model nor by a single-level ontology. Given the great amount of accounting entry definitions, an important

Axiom	Description
(1) $Entry \sqsubseteq \exists detailD.\top \sqcap \exists detailC.\top$	Entries are balanced double entry records.
(2) $Det \sqsubseteq = 1 account.Account$	A detail has associated a single account.
(3) $RenterPayEnt \sqsubseteq \forall detailD.(PayCashDet \sqcup PayBankDet)$ $RenterPayEnt \sqsubseteq \forall detailC.(PayDebtDet \sqcup PayRentFee)$	Subclasses of *Entry* are described by the subclasses of *Det* they have associated at debit and credit
(4) $PayCashDet \sqsubseteq \forall detail^{-}.RenterPayEnt$ $PayBankDet \sqsubseteq \forall detail^{-}.RenterPayEnt$ $PayDebtDet \sqsubseteq \forall detail^{-}.RenterPayEnt$ $PayFeeDet \sqsubseteq \forall detail^{-}.RenterPayEnt$	Subclasses of *Det* correspond to only one subclass of *Entry*.
(5) $PayCashDet \sqsubseteq \exists account.\{cash\}$ $PayBankDet \sqsubseteq \exists account.\{bank\}$ $PayDebtDet \sqsubseteq \exists account.\{renterDebt\}$ $PayFeeDet \sqsubseteq \exists account.\{renterFee\}$	Subclasses of *Det* are described by the accounts they have associated at debit and credit.

Table 3.2: Example of SIGGA OM Tbox

functional requirement for expert users is to be able to easily validate the correctness and completeness of their definitions. For instance, if the renter "Juan Pérez" incurs in a debt for damages in the house, the debt must be registered according to the entry definition 30 of Figure 3.9 with a debit for the "Damage Expenses" account and a credit for the "Landlords" account. But as there is no row in the table DetDefs associating the entry definition 10 (Renter payment) to the account "Damage Expenses", the "renter payment" entry which registers that "Juan Pérez" pays the damage expenses cannot be done. What happens is that the expert did not include the "Damage Expenses" account at credit in the entry definition 10. So, it should be expressed that the "renter payment" entry definition must have credit details for all accounts that are debits of renters in some entry definition. This kind of completeness rules on expert definitions, not on "data" at the level of operators, are not expressed by the relational model of Figure 3.9. Regarding the SIGGA OM model of Figure 3.10, this kind of relations between expert definitions cannot be expressed either. In fact, it would be needed to explore all accounting entry definition axioms such as (3) but it is not possible to declare this restriction in the ontology.

Moreover, there are correctness requirements that should be solved for expert definitions. For instance, for all accounting entries that move availability accounts (cash, bank), it must hold that, if an availability account is at debit, no non availability account can be also at debit (and the same at credit). This requirement is illustrated for the entry definition 10 of Renter payment in Figure 3.9 which has only cash and bank accounts at debit whereas accounts such as Renter Debt or Landlords cannot be at debit. Unlike the requirement of completeness described above, this correctness rule can be checked at operational level for each accounting entry of Renter payment such as the "Juan Pérez payment". However, it is better to check for this condition only once for an accounting entry definition instead of checking the same condition over all concrete accounting entries (such as the "Juan Pérez payment") which agree with that definition.

Due to the identification of two perspectives and with the aim that the domain conceptualization to be clear and explicit, an important *non functional requirement* is *to differenciate definitional and operational views as two abstraction levels.* Besides a greater expressiveness, a model with the capability to represent that tables

EntryDefs, DetDefs and Accounts represent the definitional perspective whereas tables Entries and Dets represent the operational level, avoids errors of design that can arise from misinterpretations. For example, at the moment of reusing the SIGGA relational schema, it is useful to distinguish a foreign key from the table Entries to the table EntryDefs of Figure 3.9, that relates conceptually equivalent knowledge with different granularity, from a foreign key from DetDefs to EntryDefs, that links two conceptually different objects at the same level. In particular, the relational model has not the expressibity to differentiate both abstraction levels.

3.5.1 Meta-modelling in the accounting domain

A possible solution to the group of requirements 3 is illustrated by the model of Figure 3.11, named SIGGA OMM, with the Tbox of Table 3.3 [Rohrer18].

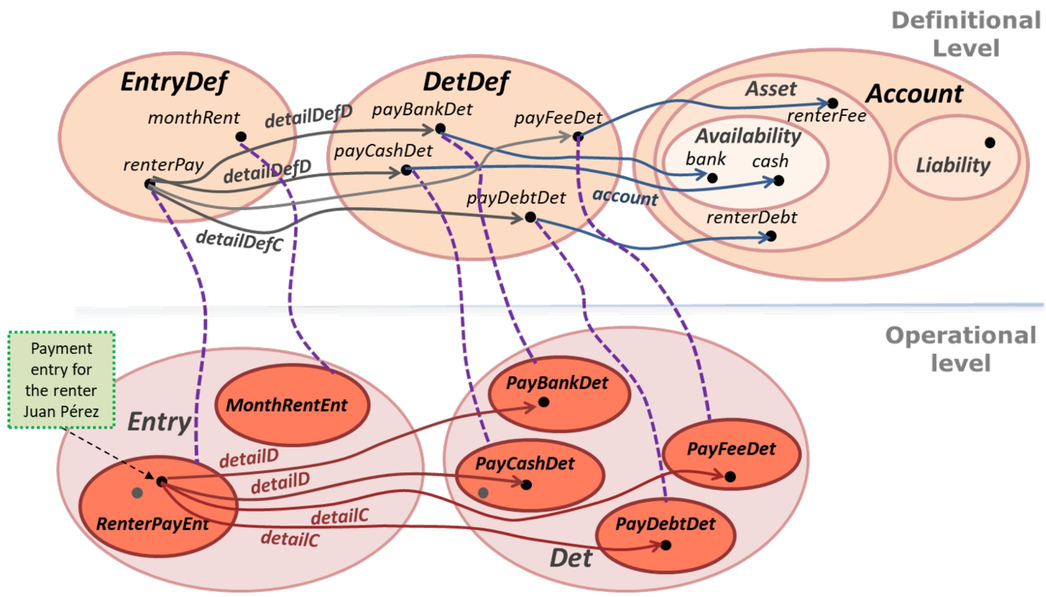

Figure 3.11: Example of SIGGA OMM design

It is easy to see that the non functional requirement is satisfied because the model of Figure 3.11 explicitly represents the two perspectives by two ontologies related by meta-modelling: the one in the upper part of the figure captures the definitional perspective whereas the ontology in the lower part represents the operational perspective. The definitional ontology has a concept *EntryDef* that represents the set of accounting entry definitions, a concept *DetDef* that represents debit and credit details that must be registered for each entry definition, and the concept *Account*. Roles *detailDefD* and *detailDefC* connect accounting entry definitions to debit and credit detail definitions. At the lower part of the figure it is depicted a model similar to SIGGA OM which represents accounting entries and details at the operational level. Note that accounts represented by the concept *Account* are in the definitional ontology since experts are in charge of defining accounts associated to debit and credit details. Moreover, note that Tbox axioms such as (3), (4) and (5) of Table

Axiom	Description
(1) $Entry \sqsubseteq \exists detailD.\top \sqcap \exists detailC.\top$	Entries are balanced double entry records.
(2) $EntryDef \sqsubseteq \exists detailDefD.\top \sqcap \exists detailDefC.\top$	Accounting entry definitions are balanced double entry records.
(3) $DetDef \sqsubseteq = 1 account.Account$	A detail definition has associated a single account.
(4) $Account \sqcap \neg Avalilability \sqcap \exists account^{-}.(\exists detailDefD^{-}.\top) \sqsubseteq \exists account.^{-}.(\exists detailDefC^{-}.(\exists detailDefD.\exists account.Avaliability))$	All accounts that generate debts must be in the definitions that have availability accounts (cash, bank), at credit
(5) $EntryDef \sqcap \exists detailDefD.(\exists account.Availability) \sqsubseteq EntryDef \sqcap \forall detailDefD.(\exists account.Availability)$	If an availability account is at debit, no non availability account can be also at debit.

Table 3.3: SIGGA OMM Tbox.

3.2 are not declared in SIGGA OMM. The solution of Figure 3.11 captures that definitions given by experts, expressed as relations between accounting entry definitions and their valid details at debit and credit, must hold at operational level as relations between sets of concrete accounting entries and their details. Hence, in the definitional ontology, each accounting entry definition is represented as an instance of *EntryDef* whereas in the operational ontology it is represented as a subclass of *Entry* that classify concrete accounting entries which agrees with the expert definition. For example, the "renter payment" entry definition represented by the individual $renterPay$ is the same real object than the concept $RenterPayEnt$. Moreover, the individual $renterPay$ has associated (by roles *detailDefD* and *detailDefC*) detail definitions at debit represented by individuals *payBankDet* and *payCashDet* and detail definitions at credit represented by individuals *payDebtDet* and *payFeeDet*. Individuals *payBankDet*, *payCashDet*, *payDebtDet* and *payFeeDet* correspond to concepts *PayBankDet*, *PayCashDet*, *PayDebtDet* and *PayFeeDet* in the operational ontology, in such a way the meta-modelling relation ensures that instances of these concepts are details of accounting entries that belong to the concept $RenterPayEnt$. Hence, the meta-modelling also hold between relations, i.e. relations are transfered between different perspectives, which solves the requirement about that accounting entries have details for accounts in accordance with the expert definitions (also solved by the relational model).

Regarding functional requirements of the group of requirements 3 about completeness of expert definitions, Tbox axioms such as (4) in Table 3.3 express rules of completeness for entry definitions such as the Renter Payment, for the example of damage expenses.The axiom expresses that all non availabiliy accounts (e.g. Cash and Bank) that are at debit in an entry definition, must be at credit in some entry definition for payment. To check the correctness of expert definitions, the Tbox axiom (5) in Table 3.3 ensures that if an availability account is at debit, no non availability account can be also at debit.

Note that the SIGGA OMM model does not require that the ontology engineer introduces assertions such as (3), (4) and (5) of Table 3.2.

3.5.2 Comparison of meta-modelling approaches

This section compares the meta-modelling approaches *Henkin-fixed layered*, *Hilog-fixed layered* and *Hilog-global layered* regarding the expressivity for representing meta-modelling in the accounting domain, for the model illustrated in Figure 3.11. Moreover, the meta-modelling approaches are analyzed to evaluate if they prevent from some inconsistencies and ensure the well-foundedness of the interpretation domain.

Henkin-fixed layered approach. Since the interpretation domain is fixed-layered, concepts *Entry*, *Det*, *Account* and their subclasses are typed with 1 since all their instances are atomic objects. Concepts *EntryDef* and *DetDef* are typed with 2 given that its instances are individuals that corresponds by meta-modelling to concepts of level 1, for instance the concept *RenterPayEnt*. It is important to observe that even though concepts *Entry*, *Det* and *Account* belong all to the meta-modelling level 1, they do not represent the same perspective (*Entry* and *Det* are in the operational perspective whereas *Account* is in the definitional perspective). The inter-layer role *account* cannot be represented by the approach of Pan (it does not allow roles with the domain and the range from different levels) but can be represented by the approach of Homola, i.e. $account^{\mathcal{I}} \subseteq \Delta_1 \times \Delta_0$. Due to it is a fixed-layered approach, the axiom $EntryDef \sqsubseteq RenterPayEnt$, which means that *RenterPayEnt* belongs to itself, is not allowed since it combines concepts from different meta-modelling levels. Hence, the Henkin-fixed layered approach ensures the well-foundedness of the interpretation domain. Moreover, properties of intensional regularity and extensionality of the Henkin semantics prevents from other inconsistencies due to meta-modelling as the one illustrated in the example below.

Example 1. Suppose that the following axioms are added to the ontology of Figure 3.11.

$$bank \neq cash$$
$$PayBankDet \equiv PayCashDet$$

For any interpretation $\mathcal{I}$, $PayBankDet^{\mathcal{I}} = PayCashDet^{\mathcal{I}}$ must hold, and by extensionality $payBankDet^{\mathcal{I}} = payCashDet^{\mathcal{I}}$. Then, from $DetDef \sqsubseteq = 1 account.Account$ (axiom (3) in Table 3.3) it follows that $bank^{\mathcal{I}} = cash^{\mathcal{I}}$, which is an inconsistency since $bank \neq cash$.

....

As in the educational domain, definitions or "rules" given by experts at the higher level, represented in the Abox by relations between instances with meta-modelling (e.g. entry definitions related to valid detail definitions) should be transferred as relations between corresponding concepts in the Tbox. This means that Tbox axioms such as (3) of Table 3.2 should be infered to ensure the agreement of concrete accounting entries with definitions of experts. This kind of rule transference between levels is not solved by Henkin-fixed layered approaches.

Hilog-fixed layered approach. As in the previous approach, concepts *Entry*, *Det*, *Account* and their subclasses are typed with 1 and concepts *EntryDef* and *DetDef* are typed with 2. The inter-layer role *account* can be represented since the approach of Homola allows roles with domain and range from different levels, and the well-foundedness of the interpretation domain is also ensured since it is a fixed-layered approach. However, from the lack of extensionality of the Hilog semantics, the inconsistency that must arise in Example 1 is not detected. The reason is that $payBankDet = payCashDet$ is not inferred and then $bank = cash$ does not follow from $DetDef \sqsubseteq = 1 account.Account$. Note that this inconsistency causes that cash and bank payments are merged in the operational level whereas experts have defined that they must be distinguished. Rule transference between levels (from expert definitions to the work of operators) is not solved by Hilog-fixed layered approaches.

Hilog-global layered approach. As in previous case studies the flexibiliy of the global domain allows expressing all concepts and roles which must not be typed. However, it does not ensure well-foundedness, e.g. it allows the axiom $EntryDef \sqsubseteq RenterPayEnt$ which means that *RenterPayEnt* belongs to itself. Again, rule transference between levels is not solved and the inconsistency illustrated in Example 1 is not detected due to the lack of extensionality of the Hilog semantics.

3.5.3 Conclusions about meta-modelling for accounting

The accounting case study presented in this section corresponds to an implemented application modelled by a relational model. The text below analyzes the value added by the meta-modelling ontology-based solution to the relational model of SIGGA.

For the *operational perspective*, an ontology-based approach allows to express basic rules of the domain which the relational model cannot express, such as that accounting entries have at least one debit detail and one credit detail (group of requirements 2). This kind of requirements can only be expressed in the relational model by non structural restrictions. A valuable contribution of an ontology-based approach is the classification of accounting entries according to definitions of experts, for example the concept *RenterPayEnt* for the set of all renter payment entries. Moreover, this solution has the flexibility of just adding new Tbox axioms to categorize entries and details by different criteria.

For the *definitional perspective*, there are relevant requirements that are not solved by the relational model nor by a single-level ontology. It is the case of explicitly expressing completeness and correctness rules for the definitions of experts, e.g. the rules that restrict details at debit and credit of accounting entry definitions. This kind of requirements are represented by Tbox axioms such as (4) and (5) in Table 3.3, in the definitional ontology of Figure 3.11 (SIGGA OMM). In this way, the meta-modelling solution provides a mechanism to define rules on the rules for concrete accounting entries.

Regarding requirements which involve both *definitional and operational perspectives* below some benefits of the meta-modelling solution SIGGA OMM are described.

- On the one hand, it allows distinguishing definitional and operational views as two abstraction levels, and on the other hand both levels are integrated by the meta-modelling correspondence that unifies different representations (individuals and concepts) of the same real objects. This non-functional requirement cannot be expressed by the relational model nor by a single-level ontology-based approach. Considering the existing instance of the relational database of SIGGA, a meta-modelling ontology-based approach would enriche the relational model by representing two levels of knowledge.

- The meta-modelling approach makes even more explicit some requirements that are solved by the relational model. For example, the essential requirement about that accounting entries have details for accounts in accordance with the expert definitions is satisfied by the structural constraints of the SIGGA database instance. However, it is enhanced by the meta-modelling approach which adds information about the abstraction level each table of the SIGGA database belongs to.

- The SIGGA OMM model provides an intuitive and flexible way for introducing business rules in the ABox as "data", at the level of experts. Experts define accounting entries with debit and credit accounts for different financial movements, and change the definitions according to different factors, as the economical policy, or change of authorities. The fact that the definitional ontology in the OMM model "describes definitions" turns out natural defining relations and imposing restrictions on them. Thus, some rules can be declared for accounting entry definitions so that experts can check for inconsistencies in their definitions before the application is running.

- Declaring rules for expert definitions has the advantage that it is possible to check for a condition only once for an accounting entry definition instead of checking the same condition over all concrete accounting entries which agree with that definition. For instance, the Tbox axiom (5) in Table 3.3 can also be declared for the single-level model OM as follows:

$$Entry \sqcap \exists detailD.(\exists account.Availability) \sqsubseteq$$
$$Entry \sqcap \forall detailD.(\exists account.Availability)$$

  However, for OM all daily transactions such as the ones of renter payment must be validated whereas for the OMM model it is enough to check these conditions at the moment of introducing the definition of the renter payment entry, because the meta-modelling ontology-based approach ensures that concrete accounting entries agree with corresponding definitions.

3.6 Recommender systems

This section describes the conceptualization of recommender systems of contents for a given domain, based on the quality of the contents and the user profile. The model for recommender systems described below is basically the same regardless of the particular domain of recommendation.

3.6.1 A model for systems that recommends web contents

The main motivation for modelling recommender systems is the fact that many frequently queried websites about delicate domains such as health or education do no contain data of good quality. Within the present work, the first model for

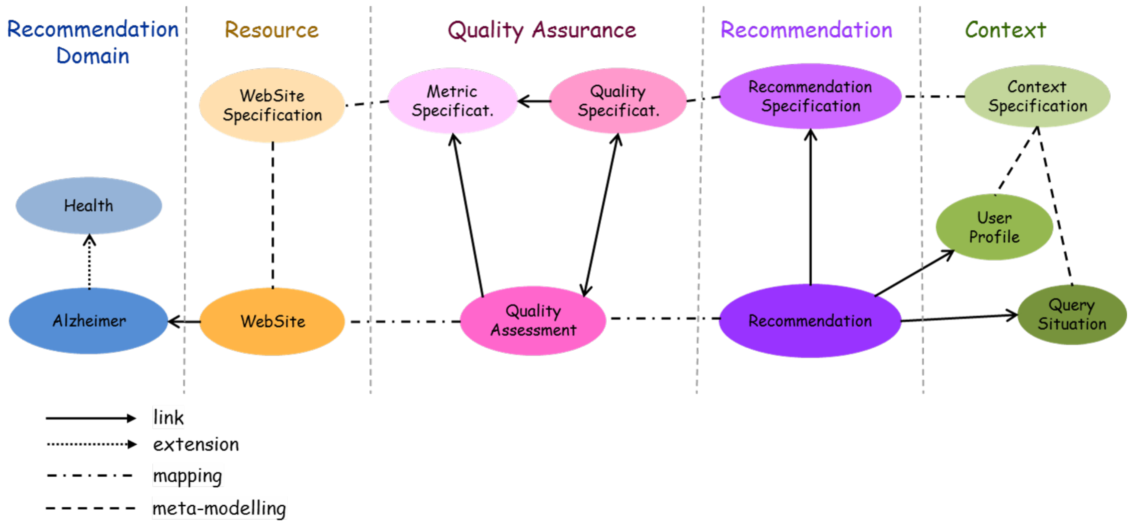

Figure 3.12: Ontology network for a recommender system

recommending health contents was developed under the project SALUS/CYTED [SALUS]. The main idea behind this health recommender system is that the recommendation of contents is obtained taking into account the quality of the published information and the profile and context of the user that consumes such information [Rohrer10a, Rohrer10b, Rohrer10c, Diaz11, Rohrer11]. This approach was also adopted for modelling recommender systems about education [Diaz12].

Besides the domain under recommendation, e.g. the health domain, recommender systems also involve several other domains: the resource domain, the quality assurance domain, the context user domain and the recommendation domain, all of them integrated in an ontology network through the four kind of relations described in Section 3.1. Figure 3.12 shows the conceptualization of the system, with different domains represented in different columns and different ontologies represented by ovals. The four relations *link*, *extension*, *mapping* and *meta-modelling* are characterized by different styles of lines connecting ontologies of the same and of different domains. Ontologies in the upper part of the figure conceptualize the *perspective of system administrators and experts on the recommendation domain*. They define metrics for evaluating the quality of contents and rules to decide the set of contents that are recommended for a given user. Ontologies in the lower part of the figure conceptualize the *perspective of users who consume domain contents and also of agents that execute the recommendation process* for a given repository of contents and a given user[8]. Below domains are briefly described as well as how they are related with each other.

Recommendation domain. In the example of Figure 3.12 the recommendation domain corresponds to the health domain. It comprises the *Health* ontology, with a general vocabulary about health, e.g. SNOMED, and the *Alzheimer* ontology, a specialization of the *Health* ontology for the alzheimer disease. Then, both ontologies are related by the *extension* relationship described in Section 3.1.

[8]Even though the idea of *perspective* is generally associated to a human user, in fact, Definition 1 points to represent a set of objects with a given granularity.

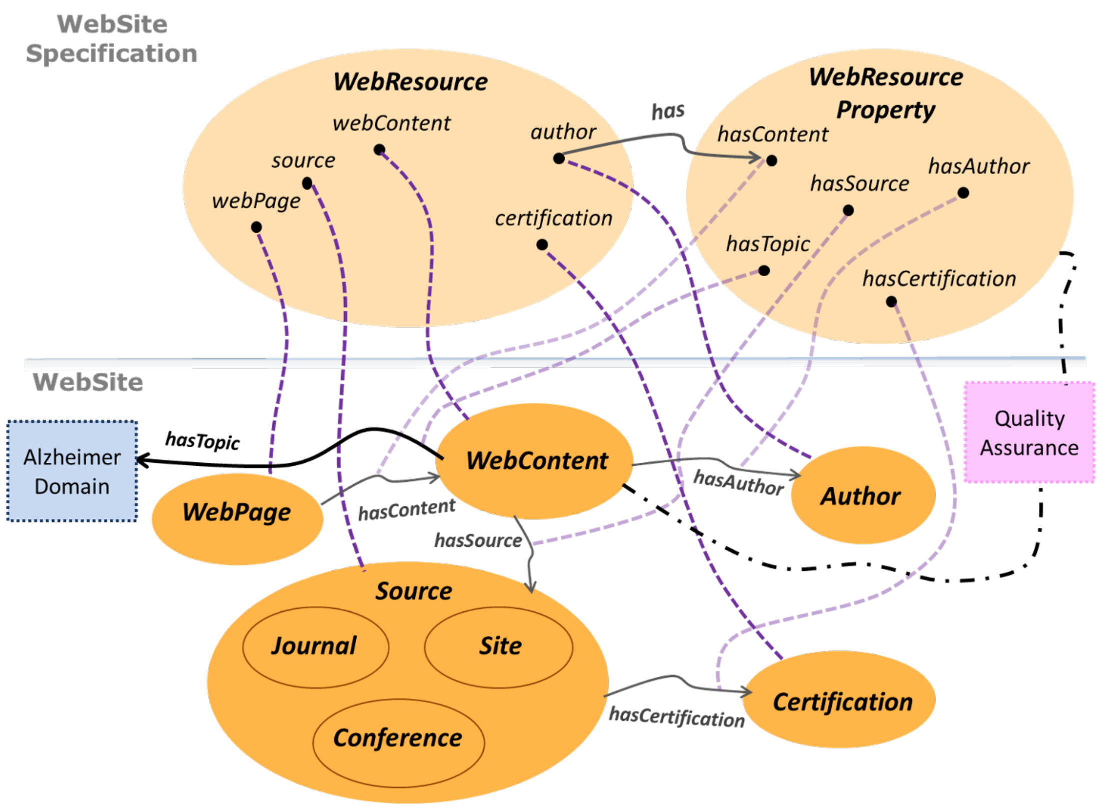

Figure 3.13: Ontology network for the resource domain

Resource. This domain is conceptualized by the ontologies *WebSite Specification* and *WebSite* illustrated in Figure 3.13. It is related to the Recommendation domain and Quality Assurance domains.

The *WebSite Specification* ontology conceptualizes different kind of web resources and its properties. A *WebResource* is any resource identified by an URL, such as a web page, a web content (e.g. a pdf file) or an author. The concept *WebResourceProperty* models the properties of a web resource, which is represented by its instances *hasContent*, *hasSource*, *hasAuthor* and *hasTopic*, among others.

The *WebSite* ontology conceptualizes concrete web resouces such as a web page identified by a given URL and its author, and describes their contents. Concepts such as *WebPage*, *WebContent* and *Author* are related by meta-modelling to instances *webPage*, *webContent* and *author* of the concept *WebResource* in the *WebSite Specification* ontology. Moreover, roles *hasContent*, *hasSource*, *hasAuthor* and *hasTopic* correspond by meta-modelling to instances *hasContent*, *hasSource*, *hasAuthor* and *hasTopic* of the concept *WebResourceProperty* in the *WebSite Specification* ontology. Note that this case study also considers meta-modelling for roles, which even though is out of the scope of the present work, it is covered by some of the meta-modelling approaches described in Chapter 2.

Ontologies of the Resource domain described above are related to ontologies of the Recommendation domain and Quality Assurance domains through *link* and *mapping* relations respectively. A link relation given by the role *hasTopic* connects instances of the concept *WebContent* to instances of some health concepts in the *Alzheimer* ontology. The concept *WebResourceProperty* is mapped to some concept of the Quality Assurance domain to model the properties of contents that are considered in the definition of metrics to assess the quality of web resources. The concept *WebContent* is mapped to some concept in the conceptualization of the quality assessment of resources.

Quality Assurance. Figure 3.12 illustrates a model with three ontologies. Ontologies *Metric Specification* and *Quality Specification* conceptualize the perspective of the expert on the recommendation domain, who defines metrics to assess web resources for different quality dimensions such readability, believability or timeliness. The ontology *Quality Assessment* conceptualizes the process of evaluating the quality of a particular resource (e.g. a web page about Alzheimer). This process is executed by an automatic agent and returns a quality level. It applies the metrics defined by experts which take as input the values of the properties for the particular resource that is assessed, e.g. its author or the reputation of its source.

Context. The three ontologies in the Context column of Figure 3.12 basically conceptualize the profile and the context of the user that looks for some information about a given topic, e.g. the alzheimer disease. The ontology *Context Specification* corresponds to the perspective of system administrators and experts. It describes the resources associated to the user that must be considered to recommend him/her some web content, e.g. user properties such as the age or the academic level, or query properties such as the goal of the query. Ontologies *User Profile* and *Query Situation* model such properties for particular users that make queries. Then, they have concepts such as *QueryGoal* to describe queries, and roles such as *hasAgeRange* and *hasAcademicLevel* to describe users. These roles are related by meta-modelling to instances about age and academic level defined by experts as the key user characteristics for recommendation.

Recommendation. This domain corresponds to the recommendation itself, i.e. given a repository of web contents of a given domain and a user that makes a query about that domain, the proposed model conceptualizes how a set of contents from this repository is recommended for the user. Experts and system administrators define the rules of recommendation which are based on definitions about the quality assessment of contents and the user and query properties that must be considered. These rules are conceptualized by the ontology *Recommendation Specification.* This ontology is related to the *Quality Specification* ontology by a mapping relation between concepts that represent the quality dimensions involved in the recommendation rule. The ontology *Recommendation Specification* is also related by mapping to the *Context Specification* ontology to be aligned with the user and query properties that are considered in the rules of recommendation. From the perspective of the execution of a recommendation process for a given repository and a given user, the ontology *Recommendation* models precisely this process. Then, concrete instances of recommendation are connected to ontologies *Recommendation Specification*, *User Profile* and *Query Situation* by *link* relations, and to the ontology *Quality Assessment* by a *mapping* relation. The relation with the *Recommendation Specification* ontology allows representing the rules of recommendation that are applied. The relation with the ontology *Quality Assessment* connects to the quality evaluation of resources from the repository. Links to ontologies *User Profile* and *Query Situation* are introduced to get values of properties that correspond to the user that makes the query. A recommendation level for each content is modelled in the *Recommendation* ontology as the result of the recommendation process.

3.6.2 Comparison of meta-modelling approaches

In this section the three semantic approaches for meta-modelling are analyzed regarding their expressivity and capability of ensuring the consistency of the model illustrated in Figure 3.12.

Henkin-fixed layered approach. Due to the fixed-layered domain, concepts of ontologies belonging to the *recommendation domain*, *quality assurance* and *recommendation* domains are all typed with the level 1. Level 1 is also associated to concepts of the *WebSite* ontology in the *resource* domain and to concepts of ontologies *User Profile* and *Query Situation* in the *context* domain. In the *resource* domain, concepts *WebResource* and *WebResourceProperty* of the *WebSite Specification* ontology illustrated in Figure 3.13 should be typed with level 2 since their instances are concepts and roles respectively. However, the concept *WebResourceProperty* cannot be interpreted by a fixed layered approach since it is mapped to a concept of the *quality assurance* domain that has level 1. Moreover, it has meta-modelling with roles, which is only solved by the approach of Pan et al. (see Table 2.3 of Chapter 2). A similar situation occurs with the mapping relation between the *Recommendation Specification* and the *Context Specification* ontologies. However, the fixed layered approach ensures the well-foundedness of the interpretation domain. Properties of intensional regularity and extensionality of the Henkin style semantics also ensure the coherence between levels in the sense that equality and difference of objects are kept through meta-modelling levels.

Hilog-fixed layered approach. As for the previous approach, on the one hand, the same problem occurs for representing concepts of the *WebSite Specification* and the *Context Specification* ontologies, due to the domain of interpretation has fixed layers. On the other hand, the well-foundedness of the domain is ensured. However, as the Hilog semantics does not satisfy extensionality, if for the model of Figure 3.13 concepts *WebContent* and *WebPage* are equated, the equality of the individuals *webContent* and *webPage* is not entailed.

Hilog-global layered approach. The global-layered domain allows representing all concepts without being typed. However, it does not ensure that the domain is well-founded, e.g. if someone adds a mapping relation between ontologies *WebSite Specification* and *WebSite* given by the axiom $WebResource \sqsubseteq WebContent$, the concept $WebContent$ will be interpreted as belonging to itself.

As a final conclusion about the capability of meta-modelling approaches to represent the model of Figure 3.12, approaches of fixed layers are not suitable for modelling recommender systems. Ontologies of Quality Assurance and Recommendation domains have only one meta-modelling level but they must be related to different meta-modelling levels of ontologies from Resource and Context domains. Regarding existing global approaches, they have the problem of not defining the domain in such a way the well-foundedness is ensured. Hence, even though Henkin approaches guarantees a coherence between different perspectives, both fixed and global domain definitions have drawbacks to model all requirements of the recommender systems illustrated in Figure 3.12.

3.7 Meta-modelling for terms denoting concepts

There exist scenarios of use of meta-modelling such as the terminology of animal kingdom species, where terms or names of objects are represented as individuals whereas corresponding real objets are represented as concepts. Below a couple of examples are presented [Kubincova16a].

Example 1.

$$\begin{array}{ll} \mathit{CervusCamelopardalis} \neq \mathit{GiraffaCamelopardalis} & \\ \mathit{CervusCamelopardalis} \equiv \mathit{GiraffaCamelopardalis} & \\ \mathit{Deprecated(CervusCamelopardalis)} & \neg \mathit{Deprecated(GiraffaCamelopardalis)} \end{array}$$

Axioms above express that names *CervusCamelopardalis* and *GiraffaCamelopardalis* are different but concepts *CervusCamelopardalis* and *GiraffaCamelopardalis* are equivalent, and that the name “Cervus camelopardalis” is deprecated whereas “Giraffa camelopardalis” is not.

. . . .

Example 2.

$$\begin{array}{ll} Sommeromys \equiv SommeromysMacrorhinos & \\ Genus(Sommeromys) & Species(SommeromysMacrorhinos) \\ Genus \sqcap Species \sqsubseteq \bot & \end{array}$$

The first axiom expresses that *Sommeromys* and *SommeromysMacrorhinos* represent the same real objects. However, *Sommeromys* denotes a genus whereas *SommeromysMacrorhinos* denotes a species, and the set of species terms are different from the set of genus terms.

. . . .

In both examples, names play the role of individuals in Abox axioms and the same names play the role of concepts in Tbox axioms. However, these names do not represent the same objects because individuals represent character strings whereas concepts represent the sets of objects denoted by the strings. Note that Examples 1 and 2 can be represented by Hilog approaches, both fixed and global layered, for which only the intensional regularity property holds. However, the Henkin-fixed layered approach does not represent these examples since for the Henkin semantics the extensionality property must also hold.
For Example 1, as *CervusCamelopardalis* $\equiv$ *GiraffaCamelopardalis* is declared then *CervusCamelopardalis* $=$ *GiraffaCamelopardalis* must hold (see Subsection 2.3.2 in Chapter 2). This contradicts
CervusCamelopardalis $\neq$ *GiraffaCamelopardalis*. In the same way, for Example 2,

since *Sommeromys* $\equiv$ *SommeromysMacrorhinos* is declared, then *Sommeromys* $=$ *SommeromysMacrorhinos* must hold, which contradicts the fact that *Sommeromys* and *SommeromysMacrorhinos* as individuals belong to disjoint concepts *Genus* and *Species*.

Even though Examples 1 and 2 can be a simple solution to represent scenarios where different strings denote the same set of objects, it is not the kind of meta-modelling addressed in the present thesis work. By contrast, in the present work meta-modelling is used to represent the same objects with different granularity, e.g. the views of expert and operator users regarding accounting entries in the accounting domain.

3.8 Conclusions

This chapter shows the use of meta-modelling for the conceptualization of a set or "class" of case studies taken from real applications and projects. The kind of meta-modelling that fits for the presented scenarios is the one that allows representing domain objects with different granularity, and moreover to transfer relations between the objects represented that way.

Three different notions are involved in the conceptualization of the case studies: *ontology network* and *meta-modelling level* which have already been introduced in the literature, and the notion of *perspective* which from the literature review done, we conclude that it has not been explicitly introduced before. The notion of *ontology network* helps managing the complexity of a business by dividing the domain into cohesive components and also helps reusing existing components in the modelling of a new application. The *meta-modelling level* allows expressing the required granularity for representing a given domain object, i.e. as individual, concept, meta-concept, meta-meta concept, and so on. Moreover, to capture the perception of the same set of objects by differents agents (humans or not), in the present work the notion of *perspective* is introduced. The idea of perspective does not necessarily agrees with the notion of meta-modelling level since, given a set of objects and a perspective of them, each object of the set can belong to a different meta-modelling level. Moreover, for a given set of objects, different users (with different perspectives) can visualize a subset of them with the same granularity whereas they visualize another subset with different granularity. For the accounting scenario described above, in the simplified conceptualization of Figure 3.11 accounts are modelled as individuals for the perspective of experts since they are in charge of defining the set of accounts. However, accounts are also represented as individuals in the perspective of operators users; e.g. suppose they are interested in a report containing the accounting entries with their details at debit and credit and the associated accounts. In particular, fixed-layered description logics meta-modelling approaches do not allow representing different perspectives with the flexibility required by the described scenarios.

Most of case studies presented in previous sections cannot be modelled by fixed-layered approaches. But the case studies can be modelled with the Hilog-global layered approach due to the flexibility of not restricting that all instances of a concept must belong to the immediate lower meta-modelling level. However, the defined semantics for the Hilog-global layered approach does not ensure the well-foundedness of the interpretation domain, and the lack of extensionality does not transfer the equivalence of concepts to the level of individuals. Moreover, none of the approaches solves the transference of rules between levels. Table 3.4 summarizes the main characteristics of the three meta-modelling approaches existing in the literature, with the drawbacks highlighted. The right lower quadrant of the figure shows that a Henkin-global layered approach is missing.

Intensions-extensions / Domain layers	Fixed	Global
Hilog	**Concepts and roles not all represented** Well-foundedness ensured **Concepts equiv. not transf. to individuals** **Relations not transferred between levels**	Concepts and roles all represented **Well-foundedness not ensured** **Concepts equiv. not transf. to individuals** **Relations not transferred between levels**
Henkin	**Concepts and roles not all represented** Well-foundedness ensured Concepts equiv. transferred to individuals **Relations not transferred between levels**	

Table 3.4: Summary of meta-modelling semantics

The global domain would provide the flexibility to represent sets of objects according to different user perspectives and the strength of the Henkin semantics would detect some inconsistencies not detected by Hilog approaches. Moreover, a proper definition of the global interpretation domain would ensure its well-foundedness, either for a single ontology or for the whole ontology network (as case studies of the present chapter are conceptualized). Chapter 4 addresses a meta-modelling approach that fills the right lower quadrant of Table 3.4.

Chapter 4

A Henkin meta-modelling approach

This chapter presents a description logics meta-modelling approach which allows fully representing the class of case studies described in Chapter 3. On the one hand, this class of scenarios requires a definition of the interpretation domain such that domain elements of (any) different meta-modelling levels can interact, as long as the domain keeps well-founded. On the other hand, as the case studies of Chapter 3 conceptualize real objects with different granularity, the meta-modelling approach introduced in the present chapter follows a Henkin semantics which ensures that equalities are tranferred between layers in both directions, from individuals to concepts and viceversa. This kind of semantics avoids non expected inferences that are inconsistencies for the domains addressed in the present work [Motz14, Motz15]. Additionally, in scenarios about education and accounting, rules defined by users at higher levels must be transferred to lower levels, so this requirement is also considered in the meta-modelling approach described below [Severi19].

This chapter is organized as follows. Section 4.1 introduces the Henkin descripcion logics $\mathcal{SHIQM}$ which deal with the main requirements of the case studies in Chapter 3, specifically a flexible interaction between layers, and the well-foundedness of the domain. Section 4.2 presents the description logics $\mathcal{SHIQM}^*$ which extends $\mathcal{SHIQM}$ to allow the transference of rules declared in the Abox at higher meta-modelling levels, as Tbox axioms in lower levels. Finally, Section 4.3 presents some conclusions.

4.1 The description logic $\mathcal{SHIQM}$

The description logics $\mathcal{SHIQ}$ is an expressive fragment of the description logics $\mathcal{SROIQ}$ which is the foundation of OWL2. As well as for $\mathcal{SROIQ}$, an ontology in $\mathcal{SHIQ}$ has a Tbox, an Rbox and an Abox, but unlike $\mathcal{SROIQ}$, $\mathcal{SHIQ}$ has not nominals and its Rbox is more restricted since it does not allow for generalized role inclusion axioms [Hitzler09, Horrocks00]. The present section describes an extension of $\mathcal{SHIQ}$ called $\mathcal{SHIQM}$ that adds a new statement to represent the meta-modelling correspondence between individuals and concepts, e.g. between *renterPay* and *RenterPayEnt* for the accounting domain (see Figure 3.11 in Chapter 3). Syntax and semantics are described, as well as the algorithm for checking consistency of an ontology in $\mathcal{SHIQM}$, and moreover the intuition and some details behind the proofs of correcteness. Definitions presented in the following subsections and the proofs of correctness are fully developed in [Motz15].

4.1.1 Syntax and semantics of $\mathcal{SHIQM}$

As in $\mathcal{SHIQ}$, assume three pairwise disjoint sets: the set of individuals $a, b, \ldots$, the set of atomic concepts $A, B, \ldots$, and the set of atomic roles $R, S, \ldots$. The set of atomic roles contains all role names R and all inverse of role names R^-. To avoid roles such as R^{--}, the function $Inv(R)$ is defined such that $Inv(R) = R^-$ for R a role name, and $Inv(R) = S$ for $R = S^-$. A role is *transitive* if it has a declaration of the form $Trans(R)$.

Let $\sqsubseteq^*$ be the transitive-reflexive closure of $\sqsubseteq$ over $\mathcal{R} \cup \{Inv(R) \sqsubseteq Inv(S) \mid R \sqsubseteq S \in \mathcal{R}\}$. A role R is a *subrole* of S if $R \sqsubseteq^* S$. A role is *simple* if it is neither transitive nor has any transitive subroles.

As in $\mathcal{SHIQ}$, concepts[1] are defined by the following definition.

Definition 2 (Concepts in $\mathcal{SHIQM}$)
Concepts in $\mathcal{SHIQM}$ are defined by the grammar:
$C, D ::= A \mid \top \mid \bot \mid (\neg C) \mid (C \sqcap D) \mid (C \sqcup D) \mid (\forall R.C) \mid (\exists R.C) \mid (\geqslant n\ S.C) \mid (\leqslant n\ S.C)$
where n is a non-negative integer and S is a simple role.

$\mathcal{SHIQM}$ extends $\mathcal{SHIQ}$ with *meta-modelling statements* defined as follows.

Definition 3 (Meta-modelling equality statement)
A meta-modelling statement *of the form $a =_{\mathsf{m}} A$ is an equality statement where a is an individual and A is an atomic concept.*

Figure 4.1 shows a fragment of the geographic domain presented in Chapter 3. Meta-modelling axioms $river =_{\mathsf{m}} River$ and $lake =_{\mathsf{m}} Lake$ represent meta-modelling relations between individuals *river*, *lake* and corresponding concepts *River*, *Lake*, illustrated by dashed edges.

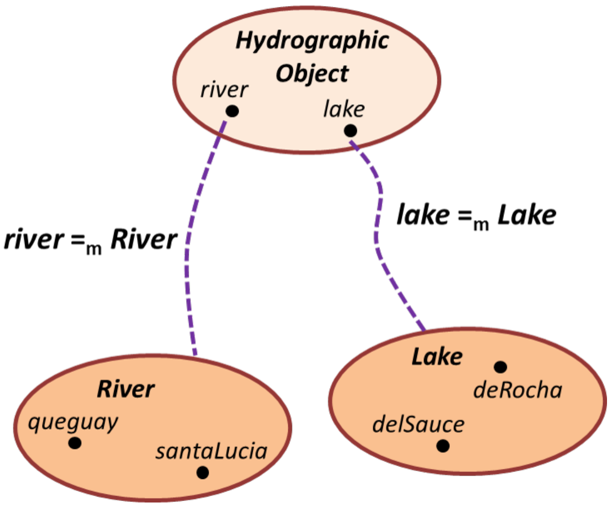

Figure 4.1: Meta-modelling relations for the hydrographic domoain

An *ontology* $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ *in* $\mathcal{SHIQM}$ has a Tbox $\mathcal{T}$, an Rbox $\mathcal{R}$, an Abox $\mathcal{A}$ and an Mbox $\mathcal{M}$ which are defined as follows.

[1]Recall that a concept is a class in OWL.

Tbox

$River \sqcap Lake \sqsubseteq \bot$

Mbox

$river =_{\mathsf{m}} River$ $lake =_{\mathsf{m}} Lake$

Abox

$HydrographicObject(river)$ $HydrographicObject(lake)$ $River(queguay)$ $River(santaLucia)$ $Lake(deRocha)$ $Lake(delSauce)$

Figure 4.2: Tbox, Rbox, Abox and Mbox for the ontology of Figure 4.1

1. $\mathcal{T}$ is a finite set of axioms of the form $C \sqsubseteq D$, with C, D any two concepts.
2. $\mathcal{R}$ is a finite set of role inclusion axioms of the form $R \sqsubseteq S$ and transitive role declarations $Trans(R)$, with R, S atomic roles.
3. $\mathcal{A}$ is a finite set of statements of the form $C(a)$, $R(a, b)$, $a = b$, or $a \neq b$.
4. $\mathcal{M}$ is a finite set of meta-modelling statements $a =_{\mathsf{m}} A$

Note that $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ is an ontology in $\mathcal{SHIQ}$. Figure 4.2 shows the Tbox, Abox and Mbox of the ontology that corresponds to Figure 4.1.

Regarding the semantics of $\mathcal{SHIQM}$, as well as in $\mathcal{SHIQ}$ the notion of *interpretation* is defined as follows.

Definition 4 (Interpretation in $\mathcal{SHIQM}$)
An interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ *in $\mathcal{SHIQM}$ consists of a* domain Δ *and a function $\cdot^{\mathcal{I}}$ which maps every concept to a subset of Δ, every role to a subset of $\Delta \times \Delta$ and every individual to an element of Δ, such that for all concepts C, D, roles R, S, and non-negative integers n, the following equations are satisfied, where $\sharp X$ denotes the cardinality of a set X:*
$(R^{-})^{\mathcal{I}} = \{(x, y) \mid (y, x) \in R^{\mathcal{I}}\}$
$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$(\neg C)^{\mathcal{I}} = \Delta \backslash C^{\mathcal{I}}$
$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.(x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.(x, y) \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\}$
$(\geqslant n\ R.C)^{\mathcal{I}} = \{x \mid \sharp\{y.(x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\}$
$(\leqslant n\ R.C)^{\mathcal{I}} = \{x \mid \sharp\{y.(x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\}$

An interpretation $\mathcal{I}$ *satisfies a TBox* $\mathcal{T}$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each $C \sqsubseteq D$ in $\mathcal{T}$.
An interpretation $\mathcal{I}$ *satisfies an RBox* $\mathcal{R}$ iff (i) $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ for each $R \sqsubseteq S$ in $\mathcal{R}$ and (ii) if $\{(x,y),(y,z)\} \subseteq R^{\mathcal{I}}$ then $(x,z) \in R^{\mathcal{I}}$ for each $Trans(R)$ in $\mathcal{R}$.
An interpretation $\mathcal{I}$ *satisfies an ABox* $\mathcal{A}$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for each $C(a)$ in $\mathcal{A}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ for each $R(a,b)$ in $\mathcal{A}$, $a^{\mathcal{I}} = b^{\mathcal{I}}$ for each $a = b$ in $\mathcal{A}$, and $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for each $a \neq b$ in $\mathcal{A}$.
The following definition introduces the *interpretation of an Mbox*.

Definition 5 (Satisfiability of meta-modelling statements)
An interpretation $\mathcal{I}$ satisfies $a =_{\mathsf{m}} A$ *if* $a^{\mathcal{I}} = A^{\mathcal{I}}$. *An interpretation* $\mathcal{I}$ satisfies and Mbox $\mathcal{M}$ *if it satisfies each statement in* $\mathcal{M}$.

Note that it is possible to equate $a^{\mathcal{I}}$ to $A^{\mathcal{I}}$ since the domain Δ can now contain atomic and non atomic objects. Hence, besides being a subset of Δ, $A^{\mathcal{I}}$ is also an element of Δ.

Example 1. According to Definition 5 individuals *river* and *lake* of Figure 4.1 are interpreted as follows:

$$river^{\mathcal{I}} = River^{\mathcal{I}} = \{queguay, santaLucia\}$$
$$lake^{\mathcal{I}} = Lake^{\mathcal{I}} = \{deRocha, delSauce\}$$

The concept *HydrographicObject* is then interpreted as follows:

$$\begin{aligned} &HydrographicObject^{\mathcal{I}} \\ &= \{river^{\mathcal{I}}, lake^{\mathcal{I}}\} \\ &= \{\{queguay, santaLucia\}, \{deRocha, delSauce\}\} \end{aligned}$$

. . . .

The example above shows that the domain of interpretation Δ cannot longer consist of only atomic objects. Individuals *queguay*, *santaLucia*, *deRocha* and *delSauce* are interpreted as atomic objects, however *river* and *lake* are interpreted as the sets $\{queguay, santaLucia\}$ and $\{deRocha, delSauce\}$. Hence, the concept *HydrographicObject* is interpreted as a set of sets. Moreover, as is intuitively introduced in chapters 2 and 3, for the meta-modelling approach addressed in the present work it is important to ensure the well-foundedness of the domain Δ.

Example 2. Suppose the axiom $HydrographicObject \sqsubseteq River$ is added to the ontology of Figure 4.1. Then, for any interpretation $\mathcal{I}$:

$$River^{\mathcal{I}} = river^{\mathcal{I}} \in HydrographicObject^{\mathcal{I}} \subseteq River^{\mathcal{I}}$$

The set $River^{\mathcal{I}}$ belongs to itself which turns the domain non well-founded. It is non-sense for the most common scenarios and applications, in particular for the scenarios described in Chapter 3.

. . . .

The definition below formalizes the notion of *well-founded set*.

Definition 6 (Well-founded Set)
A set X *is* well-founded *if for all sets* $Y \neq \emptyset$ *such that* $Y \subseteq X$, Y *has a minimal element* m*., i.e. there is no* $y \in Y$ *such that* $y \in m$.

The following definition introduces a set that is well-founded and moreover allows the definition of a domain that contains flexible layers.

Definition 7 (S_n for $n \in \mathbb{N}$)
Given a non empty set S_0 of atomic objects, S_n is defined by induction on $\mathbb{N}$ as follows: $S_{n+1} = S_n \cup \mathcal{P}(S_n)$

It is easy to prove that $S_n \subseteq S_{n+1}$ for all $n \in \mathbb{N}$. A set $X \subseteq S_n$ can contain elements x such that $x \in S_i$ for any $i \leq n$ which allows the definition of a global domain with flexible layers. This means that elements with different levels of meta-modelling can coexist in a set $X \subseteq S_n$, e.g., the set of geographic objects in Figure 3.1 has two elements with meta-modelling and one with no meta-modelling at all. The *sets S_n* are proved to be *well-founded* [Motz14, Motz15]. Note that even though the analysis of case studies of Chapter 3 show the need of ensuring the well-foundedness of the domain, Hilog-global layered approaches do not ensure it and even argue that it is not needed [Kubincova16a, page 54].

Below definitions of model and consistency for $\mathcal{SHIQM}$ are given.

Definition 8 (Model of an Ontology in $\mathcal{SHIQM}$)
An interpretation $\mathcal{I}$ is a model of an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ in $\mathcal{SHIQM}$ *(denoted as $\mathcal{I} \models \mathcal{O}$) if the following holds:*

1. *the domain Δ of the interpretation is a* subset *of some S_n for some $n \in \mathbb{N}$.*
2. *$\mathcal{I}$ is a model of the ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ in $\mathcal{SHIQ}$.*
3. *$\mathcal{I}$ is a model of $\mathcal{M}$.*

The first part of the definition expresses that the domain Δ can now contain sets since the set S_n is defined recursively using the powerset operation. The second part of Definition 8 refers to the $\mathcal{SHIQ}$-ontology without the Mbox axioms. The third part of the definition adds a condition that restricts the interpretation of an individual that has a corresponding concept through meta-modelling to be equal to the concept interpretation.

Definition 9 (Consistency in $\mathcal{SHIQM}$)
We say that an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ in $\mathcal{SHIQM}$ is consistent *if there exists a model of $\mathcal{O}$.*

The following example illustrates how the second and third conditions of Definition 8 interact.

Example 3. If the axiom *river* = *lake* is added to the ontology of Figure 4.1, the $\mathcal{SHIQ}$ ontology without the Mbox is consistent. However, the $\mathcal{SHIQM}$ ontology with the Mbox is not consistent because *River* and *Lake* are non-empty and disjoint.

Definition 5 ensures *intensional regularity* and *extensionality* properties. Both properties are recalled below for individuals a, b equated to concepts A, B by meta-modelling statements $a =_{\mathsf{m}} A$, $b =_{\mathsf{m}} B$[2]:

1. If $\mathcal{O} \models a = b$ then $\mathcal{O} \models A \equiv B$ (*intensional regularity*).
2. If $\mathcal{O} \models A \equiv B$ then $\mathcal{O} \models a = b$ (*extensionality*).

As $a^{\mathcal{I}} = A^{\mathcal{I}}$ and $b^{\mathcal{I}} = B^{\mathcal{I}}$ it is easy to see that both properties hold. Hence, the presented approach follows a Henkin semantics which along with the definition of the domain Δ as a subset of S_n allows classifying $\mathcal{SHIQM}$ as a *Henkin-global layered meta-modelling approach* (see the classification of the description logics meta-modelling approaches in Section 2.3.2 of Chapter 2).

4.1.2 A tableau algorithm for checking consistency in $\mathcal{SHIQM}$

For checking consistency of an ontology in $\mathcal{SHIQM}$ the present work adopts an approach which consists in extending the tableau algorithm for $\mathcal{SHIQ}$. Basically, the algorithm for $\mathcal{SHIQM}$ adds three new expansion rules to deal with equalities and inequalities of individuals with meta-modelling and moreover a condition that checks the well-foundedness of the interpretation domain.

As in $\mathcal{SHIQ}$, the tableau algorithm for $\mathcal{SHIQM}$ constructs a structure called *completion forest*, as a way of obtaining a canonical model and deriving logical consequences of an ontology, by applying a set of rules and checking for contradictions. Moreover, the proposed algorithm ensures that the domain of the obtained model is well-founded by checking for the existence of cycles in the completion forest. The tableau algorithm for $\mathcal{SHIQM}$ applies the same rules as for $\mathcal{SHIQ}$ (Figure 4.3) and also the three new rules for meta-modelling (Figure 4.4).

Given a $\mathcal{SHIQM}$-ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$, the Tbox $\mathcal{T}$ and the Abox $\mathcal{A}$ are assumed to be converted into *negation normal form.* This means that for general concepts as described in Definition 2, negation occurs in front of atomic concepts only and Tbox axioms $C \sqsubseteq D$ are converted into the negation normal form $\neg C \sqcup D$ (see [Motz15]).

In the following, first of all a set of definitions are presented, which are then used to describe the tableau algorithm for checking consistency of a $\mathcal{SHIQM}$-ontology $\mathcal{O}$. Finally, the intuition behind the rules is given and illustrated with some examples.

Definition 10 (Completion forest for $\mathcal{SHIQM}$)
A completion forest $\mathcal{F}$ for a $\mathcal{SHIQM}$ ontology consists of

1. *a set of nodes, labelled with individual names from the Abox and the Mbox, or variable names (fresh individuals which do not belong to the ABox nor the Mbox),*
2. *directed edges between some pairs of nodes,*
3. *for each node labelled x, a set $\mathcal{F}(x)$ of concept expressions,*

[2]Chapter 2 enunciates the properties by treating the same names as individuals and concepts, however the presentation of $\mathcal{SHIQM}$ in this chapter assumes disjoint sets of individual and concept names.

4. *for each pair of nodes x and y, a set $\mathcal{F}(x, y)$ containing role names or inverses of role names, and*

5. *two relations between nodes, denoted by $\approx$ and $\not\approx$. These relations keep record of the equalities and inequalities of nodes. The relation $\approx$ is reflexive, symmetric and transitive whereas $\not\approx$ is symmetric. The relation $\not\approx$ is* compatible with $\approx$, *i.e., if $x' \approx x$ and $x \not\approx y$ then $x' \not\approx y$ for all x, x', y. Every time a pair in $\approx$ is added, $\approx$ is closed under reflexivity, symmetry and transitivity, and every time a pair is added in either $\not\approx$ or $\approx$, $\not\approx$ is closed under compatibility with $\approx$.*

Nodes labelled with individual names which are present in the input ABox and Mbox and the ones created by the $\not\approx$-rule of Figure 4.4 are named root nodes*.*

Note that in $\mathcal{SHIQ}$ root nodes are only labelled with the individual names in the Abox.

Definition 10 introduces a new equality relation $\approx$. To clarify the meaning of equality relations considered in the present work, we have that: (i) $=$ and $\equiv$ are the equality of individuals and the equivalence of concepts in the ontology, which correspond to the most commonly adopted DL notation, (ii) $=_{\mathsf{m}}$ represents the meta-modelling correspondence between individuals and concepts, and (iii) $\approx$ represents the equality of nodes in the forest.

Definition 11 (Successor, predecessor, neighbour and ancestor in $\mathcal{SHIQM}$) *Given a completion forest $\mathcal{F}$, successor, predecessor, neighbour and ancestor are defined as follows.*

- *If nodes x and y are connected by an edge (x, y) with $R \in \mathcal{F}(x, y)$ and $R \sqsubseteq^* S$, then y is called an* S-successor *of x and x is called an* S-predecessor *of y.*

- *If y is a S-successor or an $Inv(S)$-predecessor of x then y is called an* S-neighbour *of x.*

- *A node y is a* successor *(resp.* predecessor *or* neighbour*) of x if it is a S-successor (resp. S-predecessor or S-neighbour) of x for some role S.*

- Ancestor *is the transitive closure of predecessor.*

The definition of *blocking* given below is the same as for $\mathcal{SHIQ}$. It is used to ensure termination when applying the rules of Figure 4.3.

Definition 12 (Blocking for $\mathcal{SHIQM}$)
A node is blocked *iff it is not a root node and it is either directly or indirectly blocked. A node x is* directly blocked *iff none of its ancestors are blocked, and it has ancestors x', y and y' such that*

1. *y is not a root node and*

2. *x is a successor of x' and y is a successor of y' and*

3. *$\mathcal{F}(x) = \mathcal{F}(y)$ and $\mathcal{F}(x') = \mathcal{F}(y')$ and*

4. $\mathcal{F}(x', x) = \mathcal{F}(y', y)$.

In this case, we say that y blocks x.
A node y *is* indirectly blocked *iff one of its ancestors is blocked, or it is a successor of a node* x *and* $\mathcal{F}(x, y) = \emptyset$*; the latter condition avoids wasted expansions after an application of the rule* $\leqslant$ *- rule.*

Note that the blocking strategy described above is a *pairwise blocking* since the condition is that the pair (y', y) must repeat the pair (x', x). This condition is required due to the existence of inverse and transitive roles in $\mathcal{SHIQ}$. However, for less expressive logics such as $\mathcal{ALC}$, it is enough that x repeats y.

The following definitions describe the construction of the initial completion forest, the detection of contradictions and cycles in the forest, and the notion of complete forest.

Definition 13 (Initialization)
The initial completion forest *for* $\mathcal{O}$ *is defined by the following procedure.*

1. *For each individual* a *in the ontology* ($a \in \mathcal{A} \cup \mathcal{M}$) *set* $a \approx a$.
2. *For each* $a = b \in \mathcal{A}$, *set* $a \approx b$. *We also choose an individual as a representative of each equivalence class.*
3. *For each* $a \neq b$ *in* $\mathcal{A}$, *set* $a \not\approx b$.
4. *For each* $a \in \mathcal{A} \cup \mathcal{M}$, *we do the following:*
   (a) *in case* a *is a representative of an equivalence class then set* $\mathcal{F}(a) = \{C \mid C(a') \in \mathcal{A}, a \approx a'\}$;
   (b) *in case* a *is not a representative of an equivalence class then set* $\mathcal{F}(a) = \emptyset$.
5. *For all* $a, b \in \mathcal{A} \cup \mathcal{M}$ *that are representatives of some equivalence class, if* $\{R \mid R(a', b') \in \mathcal{A}, a \approx a', b \approx b'\} \neq \emptyset$ *then create an edge from* a *to* b *and set* $\mathcal{F}(a, b) = \{R \mid R(a', b') \in \mathcal{A}, a \approx a', b \approx b'\}$.

Definition 14 (Contradiction for $\mathcal{SHIQM}$)
$\mathcal{F}$ *has a* contradiction *if either*

- A *and* $\neg A$ *belongs to* $\mathcal{F}(x)$ *for some atomic concept* A *and node* x *or*
- *there are nodes* x *and* y *such that* $x \not\approx y$ *and* $x \approx y$.
- *there is a node* x *such that* $\leqslant n\, S.C \in \mathcal{F}(x)$, *and* x *has* $n+1$ S*-neighbours* $y_1, \ldots y_{n+1}$ *with* $C \in \mathcal{F}(y_i)$, $y_i \not\approx y_j$ *for all* $i, j \in \{1, \ldots n+1\}$ *with* $i \neq j$.

Definition 15 (Cycles)
We say that the completion forest $\mathcal{F}$ *has a cycle with respect to an Mbox* $\mathcal{M}$ *if there exist a sequence of meta-modelling axioms* $A_0 =_{\mathsf{m}} a_0$, $A_1 =_{\mathsf{m}} a_1$, $\ldots$ $A_n =_{\mathsf{m}} a_n$ *all in* $\mathcal{M}$ *such that*

$$\begin{array}{ll} A_1 \in \mathcal{F}(x_0) & x_0 \approx a_0 \\ A_2 \in \mathcal{F}(x_1) & x_1 \approx a_1 \\ \vdots & \vdots \\ A_n \in \mathcal{F}(x_{n-1}) & x_{n-1} \approx a_{n-1} \\ A_0 \in \mathcal{F}(x_n) & x_n \approx a_n \end{array}$$

Example 4. Suppose we have a $\mathcal{SHIQM}$-ontology with two individuals a and b, the assertions $B(a)$ and $A(b)$, and the meta-modelling axioms:

$$a =_{\mathsf{m}} A \quad b =_{\mathsf{m}} B.$$

In the completion forest, $\mathcal{F}(a) = \{B\}$ and $\mathcal{F}(b) = \{A\}$ has a cycle since $A \in \mathcal{F}(b)$ and $B \in \mathcal{F}(a)$.

. . . .

Definition 16 ($\mathcal{SHIQM}$-Complete)
*A forest $\mathcal{F}$ is $\mathcal{SHIQM}$-*complete *(or just* complete*) if none of the rules of figures 4.3 and 4.4 is applicable.*

Given a $\mathcal{SHIQM}$-ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ in negation normal form, the tableau algorithm for $\mathcal{SHIQM}$ follows the procedure described below.

Steps of the tableau algorithm for $\mathcal{SHIQM}$

1. The *initial completion forest* $\mathcal{F}$ is built by following the procedure described by Definition 13.

2. *Expansion rules of figures* 4.3 and 4.4 *are non-deterministically applied.* For each rule execution, the algorithm checks for contradictions, cycles and the completion of the forest (see definitions 14, 15 and 16). As in $\mathcal{SHIQ}$, the application of rules extends the completion forest $\mathcal{F}$, but for $\mathcal{SHIQM}$ the new $\approx$-rule for meta-modelling (see Figure 4.4) also extends the Tbox $\mathcal{T}$. Expansion rules are applied *until one of the following conditions hold* :

   (a) a $\mathcal{SHIQM}$-complete $(\mathcal{T}, \mathcal{F})$ *without contradictions nor cycles* is obtained, or

   (b) *all the choices have yield a* $(\mathcal{T}, \mathcal{F})$ that has *either contradictions or cycles.*

3. The algorithm ends when one of the conditions (a) or (b) of the step 2. holds. When it ends, it will say whether the ontology $\mathcal{O}$ is consistent or not. $\mathcal{O}$ *is consistent* in the case (a) and $\mathcal{O}$ is *inconsistent* in the case (b).

As for all fragments of the description logic $\mathcal{SROIQ}$ underlying OWL, the core part of the algorithm is the second step of the procedure described above, i.e. the application of the expansion rules [Hitzler09, Horrocks99, Horrocks00]. To apply the rules, the algorithm iterates on the nodes of the forest $\mathcal{F}$, and in each iteration executes those rules that meet the conditions. Note that the algorithm is non-deterministic regarding the order of the rule application and moreover some rules are themselves non-deterministic (see $\sqcup$, choose, $\leqslant$ a nd $\leqslant$ - root rules in Figure 4.3 and the close-rule in Figure 4.4). For these rules, implementations of the algorithm have to guess the choices and possibly have to backtrack to choice points if a choice already made has led to a contradiction or a cycle.

The algorithm builds a forest $\mathcal{F}$ to obtain a canonical model of the ontology $\mathcal{O}$. Hence, nodes x of $\mathcal{F}$ belong to the domain of that model and must satisfy the concepts associated to $\mathcal{F}(x)$. Step 1. of initialization associates concepts to

$\sqcap$-rule:
If x is not indirectly blocked, $C \sqcap D \in \mathcal{F}(x)$ and $\{C, D\} \not\subseteq \mathcal{F}(x)$ then add $\{C, D\}$ to $\mathcal{F}(x)$.

$\sqcup$-rule:
If x is not indirectly blocked, $C \sqcup D \in \mathcal{F}(x)$ and $\{C, D\} \cap \mathcal{F}(x) = \emptyset$ then add either C or D to $\mathcal{F}(x)$.

$\exists$-rule:
If x is not blocked, $\exists R.C \in \mathcal{F}(x)$ and x has no R-neighbour y with $C \in \mathcal{F}(y)$ then
1. add a new node with label y (where y is a new node label),
2. set $\mathcal{F}(x, y) = \{R\}$,
3. set $\mathcal{F}(y) = \{C\}$.

$\forall$-rule:
If x is not indirectly blocked, $\forall R.C \in \mathcal{F}(x)$ and x has an R-neighbour y with $C \notin \mathcal{F}(y)$ then add C to $\mathcal{F}(y)$.

Tbox-rule:
If x is not indirectly blocked, C is a TBox statement and $C \notin \mathcal{F}(x)$, then add C to $\mathcal{F}(x)$.

trans-rule:
If x is not indirectly blocked, $\forall S.C \in \mathcal{F}(x)$, S has a transitive subrole R, and x has an R-neighbour y with $\forall R.C \notin \mathcal{F}(y)$, then add $\forall R.C$ to $\mathcal{F}(y)$.

choose-rule:
If x is not indirectly blocked, $\leqslant n\ S.C \in \mathcal{F}(x)$ or $\geqslant n\ S.C \in \mathcal{F}(x)$ and there is an S-neighbour y of x with $\{C, \sim C\} \cap \mathcal{F}(y) = \emptyset$, then add either C or $\sim C$ to $\mathcal{F}(y)$.

$\geqslant$ - rule:
If x is not blocked, $\geqslant n\ S.C \in \mathcal{F}(x)$ and there are no n S-neighbours $y_1, \ldots, y_n$ of x with $C \in \mathcal{F}(y_i)$, $y_i \not\approx y_j$ for $i, j \in \{1, \ldots n\}$ and $i \neq j$, then
1. create n new nodes $y_1, \ldots y_n$.
2. set $\mathcal{F}(x, y_i) = \{S\}$, $\mathcal{F}(y_i) = \{C\}$ and $y_i \not\approx y_j$ for $i, j \in \{1, \ldots n\}$, $i \neq j$.

$\leqslant$ - rule:
If x is not indirectly blocked, $\leqslant n\ S.C \in \mathcal{F}(x)$, there are more than n S-neighbours y_i of x with $C \in \mathcal{F}(y_i)$, and x has two S-neighbours y, z such that y is neither a root node nor an ancestor of z, $y \not\approx z$ does not hold, and $C \in \mathcal{F}(y) \cap \mathcal{F}(z)$, then besides setting $y \approx z$, we also do:
1. add $\mathcal{F}(y)$ to $\mathcal{F}(z)$,
2. if z is an ancestor of x, then add $\{R^- \mid R \in \mathcal{F}(x, y)\}$ to $\mathcal{F}(z, x)$,
3. if z is not an ancestor of x, then add $\mathcal{F}(x, y)$ to $\mathcal{F}(x, z)$,
4. set $\mathcal{F}(x, y) = \emptyset$, and
5. set $u \not\approx z$ for all u with $u \not\approx y$.

$\leqslant$ - root-rule:
If $\leqslant n\ S.C \in \mathcal{F}(x)$, there are more than n S-neighbours y_i of x with $C \in \mathcal{F}(y_i)$, and x has two S-neighbours y, z which are both root nodes, $y \not\approx z$ does not hold, and $C \in \mathcal{F}(y) \cap \mathcal{F}(z)$, then besides setting $y \approx z$, we also do:
1. add $\mathcal{F}(y)$ to $\mathcal{F}(z)$,
2. for all directed edges from y to some w, create an edge from z to w if it does not exist with $\mathcal{F}(z, w) = \emptyset$.
3. add $\mathcal{F}(y, w)$ to $\mathcal{F}(z, w)$,
4. for all directed edges from some w to y, create an edge from w to z if it does not exist with $\mathcal{F}(w, z) = \emptyset$,
5. add $\mathcal{F}(w, y)$ to $\mathcal{F}(w, z)$,
6. set $\mathcal{F}(y) = \emptyset$ and remove all edges from/to y.
7. set $u \not\approx z$ for all u with $u \not\approx y$.

Figure 4.3: Expansion rules for $\mathcal{SHIQ}$

$\approx$-rule:
Let $a =_\mathsf{m} A$ and $b =_\mathsf{m} B$ in $\mathcal{M}$. If $a \approx b$ and $A \sqcup \neg B, B \sqcup \neg A$ does not belong to $\mathcal{T}$ then add $A \sqcup \neg B, B \sqcup \neg A$ to $\mathcal{T}$.

$\not\approx$-rule:
Let $a =_\mathsf{m} A$ and $b =_\mathsf{m} B$ in $\mathcal{M}$. If $a \not\approx b$ and there is no root node z such that $(A \sqcap \neg B \sqcup B \sqcap \neg A) \in \mathcal{F}(z)$ then create a new root node z with $\mathcal{F}(z) = \{A \sqcap \neg B \sqcup B \sqcap \neg A\}$

close-rule:
Let $a =_\mathsf{m} A$ and $b =_\mathsf{m} B$ in $\mathcal{M}$ where $a \approx x$, $b \approx y$, x and y are their respective representatives of the equivalence classes. If neither $x \approx y$ nor $x \not\approx y$ then we add either $x \approx y$ or $x \not\approx y$. In the case $x \approx y$, we also do the following:
1. add $\mathcal{F}(y)$ to $\mathcal{F}(x)$,
2. for all directed edges from y to some w, create an edge from x to w if it does not exist with $\mathcal{F}(x, w) = \emptyset$,
3. add $\mathcal{F}(y, w)$ to $\mathcal{F}(x, w)$,
4. for all directed edges from some w to y, create an edge from w to x if it does not exist with $\mathcal{F}(w, x) = \emptyset$,
5. add $\mathcal{F}(w, y)$ to $\mathcal{F}(w, x)$,
6. set $\mathcal{F}(y) = \emptyset$ and remove all edges from/to y.

Figure 4.4: Additional Expansion Rules for $\mathcal{SHIQM}$

$\mathcal{F}(x)$ from Abox assertions (see Definition 13), and in the step 2. the Tbox-rule of Figure 4.3 associates Tbox concepts C (Tbox axioms in negation normal form) if $C \notin \mathcal{F}(x)$, so $C(x)$ must hold. Then, in each iteration the algorithm explores the set of concepts associated to nodes x in $\mathcal{F}$ and then the rules that meet the conditions will be applied until no more rules can be applied or some contradiction or cycle is found and no more alternatives (of non-deterministic rules) can be explored. Note that to detect contradictions of the form of Definition 14, it is needed to fragment complex concepts until atomic concepts A and $\neg A$ can be detected in $\mathcal{F}(x)$, which is done by the rules of Figure 4.3. The intuition behind these rules is given below.

Given a node x, if the concept $C \sqcap D$ belongs to $\mathcal{F}(x)$ then $C(x)$ and $D(x)$ must hold, so C and D are both added to $\mathcal{F}(x)$. This is exactly what the $\sqcap$-rule does.

The $\sqcup$-rule is a non-deterministic rule that each time a concept $C \sqcup D$ belongs to $\mathcal{F}(x)$, either C or D is added to $\mathcal{F}(x)$ (it is enough that one of the disjuncts holds), e.g. C is added. In case a contradiction is found backtracking is executed. Then, C is dropped from $\mathcal{F}$ and the other disjunct D is added; in case a contradiction is also reached the algorithm will continue exploring alternatives in case $C \sqcup D$ was added to $\mathcal{F}$ by a choice of a non-deterministic rule previously applied; the algorithm stops in case no more alternatives can be explored.

Rules choose, $\leqslant$ and $\leqslant$ - root are also non-deterministic. The choose-rule adds C or the negation normal form of $\neg C$ ($\sim C$) to a node y connected to x by a role S in case $\leqslant n\ S.C$ or $\geqslant n\ S.C$ belongs to $\mathcal{F}(x)$ and neither C nor $\sim C$ belong to $\mathcal{F}(y)$. This can lead to the application of other rules such as the $\leqslant$ - rule, e.g. if C is added; in case a contradiction arises then the algorithm try $\sim C$. Both $\leqslant$ and $\leqslant$ - root rules equate a pair of nodes connected to x by a role S in case $\leqslant n\ S.C \in \mathcal{F}(x)$ and more than n nodes are connected to x by S. In case that the equation of two given nodes leads to a contradiction then any other pair of nodes are equated as long as they are not different. $\leqslant$ and $\leqslant$ - root rules differ in that the $\leqslant$ - root rule is applied in case the nodes that are equated are both root nodes, otherwise the $\leqslant$ - rule is applied.

The rules described above add concepts to $\mathcal{F}(x)$ but do not add nodes to the forest. However, the $\exists$-rule extends $\mathcal{F}(x)$ in depth by adding a new node each time a concept of the form $\exists R.C$ belongs to $\mathcal{F}(x)$ and there is no node y such that $R(x, y)$ and $C(y)$ hold, as the concept $\exists R.C$ expresses. The $\exists$-rule connects the new node y to x by R and adds C to $\mathcal{F}(y)$. The $\geqslant$ - rule extends $\mathcal{F}(x)$ in the same way as the $\exists$-rule, by adding n (different) nodes to $\mathcal{F}(x)$. Note that $\exists$ and $\geqslant$ rules are coherent with the open world assumption of the semantic web, i.e. if some instances are not in the knowledge base, it does not mean they indeed do not exist in the real world; the algorithm treat them as missing data and not as an inconsistency.

The $\forall$-rule merely adds C to the forest of a node y in case a concept of the form $\forall R.C$ is in $\mathcal{F}(x)$, $R(x, y)$ holds but $C \notin \mathcal{F}(y)$. The trans-rule adds $\forall R.C$ to y in case R has a transitive subrole, since by the transitivity, if $R(y, z)$ for a node z, then $C(z)$ also must hold.

Note that most of rules of Figure 4.3 are executed as long as the node x is not blocked. As Definition 12 says, when two pairs of neighbour nodes have the same concepts and roles in $\mathcal{F}$, then concepts and roles will repeat and if rules continued to be applied the algorithm would not stop.

The new expansion rules of Figure 4.4 are introduced in the present work to deal with equalities and inequalities of individuals with meta-modelling. If $a =_{\mathsf{m}} A$ and $b =_{\mathsf{m}} B$ then the individuals a and b represent concepts. Any equality at the level of individuals should be transferred as an equality between concepts and similarly with the difference. The $\approx$-rule transfers the equality $a \approx b$ to the level of concepts by adding two concepts in negation normal form to the Tbox which are equivalent to $A \equiv B$. This rule is necessary to detect the inconsistency of Example 3 where the equality $river = lake$ is transferred as an equality $River \equiv Lake$ between concepts. A particular case of the application of the $\approx$-rule is when $a =_{\mathsf{m}} A$ and $a =_{\mathsf{m}} B$. In this case, the algorithm also adds $A \equiv B$. The $\not\approx$-rule is similar to the $\approx$-rule. However, in the case that $a \not\approx b$, $A \not\equiv B$ cannot be added because the negation of $\equiv$ is not directly available in the language. So, it is replaced by an equivalent statement, i.e., an element z is added, that witnesses this difference. Again, note that concepts added to the ABox are in negation normal form.
The rules $\approx$ and $\not\approx$ are not sufficient to detect all inconsistencies. With only these rules, the inconsistency of Example 1 in Section 3.2.2 of Chapter 3 (see Figure 3.1) cannot be detected. The idea is that the equality $A \equiv B$ between concepts must also be transfered as an equality $a \approx b$ between individuals. However, it is not enough to transfer the equalities that are in the Tbox, it is also needed to transfer the semantic consequences, e.g., $\mathcal{O} \models A \equiv B$. However, trying to solve $\mathcal{O} \models A \equiv B$ is equivalent to solve the unsatisfactibility of $\mathcal{O} \cup (A \sqcap \neg B \sqcup B \sqcap \neg A)(z)$ for a fresh domain element z. Hence, the algorithm would not terminate because the problem of semantic consequences is reduced to the problem of satisfactibility that is the one we are trying to solve[3]. The solution to this problem is to explicitly try either $a \approx b$ or $a \not\approx b$, and this is exactly what the close-rule does. The close-rule adds either $a \approx b$ or $a \not\approx b$. It is similar to the choose-rule of Figure 4.3 which adds either C or $\neg C$. For a model $\mathcal{I}$ of the ontology, we have that either $a^{\mathcal{I}} = b^{\mathcal{I}}$ or $a^{\mathcal{I}} \neq b^{\mathcal{I}}$, and since the tableau algorithm works with representatives, the rule actually equates or makes different the representatives.

[3]Note that reduction to satisfactibilitiy not only is used to solve complexity

Examples below illustrate the application of most of the rules, and moreover why blocking and checking for cycles are needed. Example 5 shows the application of a set of rules for $\mathcal{SHIQ}$, the three new rules introduced for $\mathcal{SHIQM}$ and also blocking. The algorithm return that the ontology is consistent. Example 6 shows how the algorithm applies some rules for $\mathcal{SHIQ}$ and $\mathcal{SHIQM}$, and checks for the existence of cycles. As a cycle is detected, the algorithm returns that the ontology is inconsistent because the domain of the canonical model is non well-founded. The reader can skip the examples if the algorithm is already clear for him/her.

Example 5.
$\mathcal{O} = \{(\mathcal{T}, \mathcal{A}, \mathcal{M})\}$
$\mathcal{T} = \{A \sqsubseteq \exists R.A, B \sqsubseteq \exists R.B, \exists R.\top \sqsubseteq A \sqcap B, \top \sqsubseteq \forall R.(A \sqcap B)\}$
$\mathcal{A} = \{A(p), B(q), R(p,q)\}$
$\mathcal{M} = \{a =_{\mathsf{m}} A,\ b =_{\mathsf{m}} B\}$

The initial forest $\mathcal{F}_0$ is:
$\mathcal{F}_0(a) = \emptyset$, $\mathcal{F}_0(b) = \emptyset$, $\mathcal{F}_0(p) = \{A\}$, $\mathcal{F}_0(q) = \{B\}$, $\mathcal{F}_0(p,q) = \{R\}$
Starting from $\mathcal{F}_0$, the close-rule can be applied to nodes a, b, maybe by selecting $a \not\approx b$ and the Tbox-rule can be applied to all nodes, by adding concepts $\neg A \sqcup \exists R.A$, $\neg B \sqcup \exists R.B$, $\forall R.\bot \sqcup A \sqcap B$, $\bot \sqcup \forall R.(A \sqcap B)$ to the forest (Tbox axioms en negation normal form). Then, the $\sqcup$-rule can be applied to all nodes by choosing one of the disjuncts of Tbox concepts, and also the $\not\approx$-rule which creates a new node z and set $(A \sqcap \neg B \sqcup B \sqcap \neg A)$ to $\mathcal{F}(z)$. In case a contradiction is detected, the algorithm makes backtracking and takes the other disjunct. For example, for the node p, the choice of the first disjunct in the concept $\neg A \sqcup \exists R.A$ leads to a contradiction since $\{A, \neg A\} \subseteq \mathcal{F}(p)$, then the algorithm try $\exists R.A$. Below we show the forest after applying the $\sqcup$-rule (and possibly backtracking). Note that the symbol $\supseteq$ is used instead of $=$ to highlight only the concepts that are relevant to illustrate the application of the rules.

$\mathcal{F}(a) \supseteq \{\neg A, \neg B\}$
$\mathcal{F}(b) \supseteq \{\neg A, \neg B\}$
$\mathcal{F}(p) \supseteq \{A, \exists R.A, \exists R.B, A \sqcap B, \forall R.(A \sqcap B)\}$
$\mathcal{F}(q) \supseteq \{B, \exists R.A, \exists R.B, A \sqcap B, \forall R.(A \sqcap B)\}$
$\mathcal{F}(p,q) = \{R\}$
$a \not\approx b$
$\mathcal{F}(z) \supseteq \{A \sqcap \neg B \sqcup B \sqcap \neg A\}$

After applying the Tbox, $\sqcup$ and $\sqcap$ rules to z, a contradiction is detected since either $\{A, \neg A\} \subseteq \mathcal{F}(z)$ or $\{B, \neg B\} \subseteq \mathcal{F}(z)$. Hence, backtracking is applied and $a \approx b$ is chosen, and then the $\approx$-rule is applied by adding $A \sqcup \neg B, B \sqcup \neg A$ to $\mathcal{T}$.
Rules $\exists$, $\sqcap$ and $\forall$ are applied to nodes p and q. The $\forall$-rule has no effect since it would add $A \sqcap B$ to the node q but it already exists. The $\sqcap$-rule adds B to p and A to q. The $\exists$-rule creates new nodes x and y that are R-neighbours of p, and adds A and B to x and y. As forests of x, y coincid with initial forests of p, q, in the next iterations $\mathcal{F}(p) = \mathcal{F}(x)$ and $\mathcal{F}(p) = \mathcal{F}(y)$ (since $\mathcal{F}(p) = \mathcal{F}(q)$), and the forest looks as follows:

$\mathcal{F}(a) \supseteq \{\neg A, \neg B\}$
$\mathcal{F}(b) \supseteq \{\neg A, \neg B\}$
$\mathcal{F}(p) \supseteq \{A, \exists R.A, \exists R.B, \forall R.(A \sqcap B), B\}$
$\mathcal{F}(q) \supseteq \{B, \exists R.A, \exists R.B, \forall R.(A \sqcap B), A\}$
$\mathcal{F}(p, q) = \{R\}$
$a \approx b$
$\mathcal{F}(x) \supseteq \{A, \exists R.A, \exists R.B, \forall R.(A \sqcap B), B\}$
$\mathcal{F}(y) \supseteq \{B, \exists R.A, \exists R.B, \forall R.(A \sqcap B), A\}$

The application of the $\exists$-rule to x generates two new nodes x' and x'' that are R-neighbours of x, which have exactly the same forest as x, and the same happens for y. Note that if we continue applying the $\exists$-rule, more and more nodes with the same forest will be generated infinitely. In this point, conditions of Definition 12 hold for x' and x'': x blocks x' since x is not a root node, x' is a successor of x, x is a successor of p, $\mathcal{F}(x') = \mathcal{F}(x)$, $\mathcal{F}(x) = \mathcal{F}(p)$ and $\mathcal{F}(x, x') = \mathcal{F}(p, x) = R$. The same happens for x''. Figure 4.5 shows the nodes of the forest and the arcs which are all labelled with the role R (in the figure only the arc from p to q is labelled). Concepts A and B, represented by an oval, have meta-modelling with individuals a and b which are equated. Nodes p, q and theirs successors are instances of A and B (A, B are in $\mathcal{F}(p)$, $\mathcal{F}(q)$ and their successors), the figure shows blocked nodes wrapped in a dashed rounded rectangle. It appears that there is no need of apply-

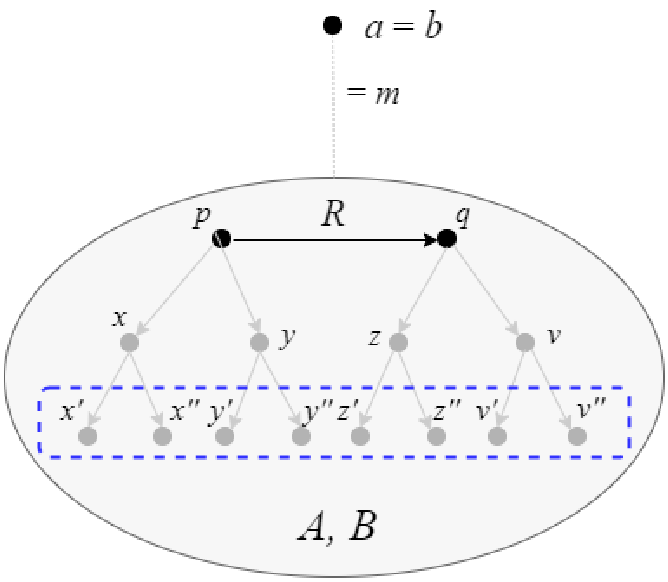

Figure 4.5: Blocked nodes in the forest of Example 5.

ing the $\exists$-rule twice from p and q; indeed in this example it is not needed, but for some combinations of inverse or transitive roles the pairwise blocking is needed (see example of page 206 in [Hitzler09]).
The algorihm does not detect cycles in the forest, just observ that instances of concepts A and B (that have meta-modelling with individuals a and b) do not have meta-modelling, so according to Definition 15 there can be no cycles in the forest.
Finally, as it is not possible to apply any more rule and $(\mathcal{T}, \mathcal{F})$ has no contradictions nor cycles, the algorithm returns that the ontology $\mathcal{O}$ is consistent.
. . . .

Example 6.
$\mathcal{O} = \{(\mathcal{T}, \mathcal{A}, \mathcal{M})\}$
$\mathcal{T} = \{X \equiv A, X \equiv C\}$
$\mathcal{A} = \{A(b), B(c)\}$
$\mathcal{M} = \{a =_{\mathsf{m}} A,\ b =_{\mathsf{m}} B,\ c =_{\mathsf{m}} C\}$

The initial forest $\mathcal{F}_0$ is:
$\mathcal{F}_0(a) = \emptyset$, $\mathcal{F}_0(b) = \{A\}$, $\mathcal{F}_0(c) = \{B\}$
Starting from $\mathcal{F}_0$, the close-rule can be applied to pairs of nodes (a, b), (a, c) and (b, c), maybe by choosing that nodes are all different ($a \not\approx b$, $a \not\approx c$, $b \not\approx c$). The Tbox-rule can be applied by adding to the forest of a, b y c the concepts $\neg A \sqcup X$, $\neg X \sqcup A$ which are equivalent to $X \equiv A$, and $\neg C \sqcup X$, $\neg X \sqcup C$ for $X \equiv C$. The result up to this point is:

$\mathcal{F}(a) \supseteq \{\neg A \sqcup X, \neg X \sqcup A, \neg C \sqcup X, \neg X \sqcup C\}$
$\mathcal{F}(b) \supseteq \{A, \neg A \sqcup X, \neg X \sqcup A, \neg C \sqcup X, \neg X \sqcup C\}$
$\mathcal{F}(c) \supseteq \{B, \neg A \sqcup X, \neg X \sqcup A, \neg C \sqcup X, \neg X \sqcup C\}$
$a \not\approx b$
$a \not\approx c$
$b \not\approx c$

The $\not\approx$-rule is applied to (a, b), (a, c) and (b, c) by generating new nodes x, y, z and adding concepts $(\neg A \sqcap B \sqcup A \sqcap \neg B)$, $(\neg A \sqcap C \sqcup A \sqcap \neg C)$, $(\neg B \sqcap C \sqcup B \sqcap \neg C)$ to their forests. Moreover the Tbox-rule is applied to x, y, z by adding $\neg A \sqcup X$, $\neg X \sqcup A$, $\neg C \sqcup X$ and $\neg X \sqcup C$. For nodes x and z after applying $\sqcup$ and $\sqcap$ rules (and making backtracking if contradictions appear), a possible selection can be concepts A, $\neg B$, X, C to x and concepts B, $\neg C$, $\neg X$, $\neg A$ to z. However, for the node y all choices lead to contradictions. For the concept $\neg A \sqcap C \sqcup A \sqcap \neg C$ either A, $\neg C$ or $\neg A$, C must be selected, and for concepts $\neg A \sqcup X$, $\neg X \sqcup A$, $\neg C \sqcup X$ and $\neg X \sqcup C$ either A, C, X or $\neg A$, $\neg C$, $\neg X$ must be selected. This results in that either $\{A, \neg A\} \subseteq \mathcal{F}(y)$ or $\{C, \neg C\} \subseteq \mathcal{F}(y)$. Hence, the algorithm have to backtrack the choice of the close-rule for (a, c) and set $a \approx c$.
The $\sqcup$-rule is applied to nodes a, b, c, and avoiding contradictions, the forest looks as follows.

$\mathcal{F}(a) \supseteq \{\neg X, \neg A, \neg C\}$
$\mathcal{F}(b) \supseteq \{A, X, C\}$
$\mathcal{F}(c) \supseteq \{B, \neg X, \neg A, \neg C\}$
$a \not\approx b$
$a \approx c$
$b \not\approx c$
$\mathcal{F}(x) \supseteq \{A, \neg B, X, C\}$
$\mathcal{F}(z) \supseteq \{B, \neg C, \neg X, \neg A\}$

According to Definition 15 at this point the algorihm detects a cycle, since we have that $b =_{\mathsf{m}} B$, $c =_{\mathsf{m}} C$, $B \in \mathcal{F}(c)$, $C \in \mathcal{F}(b)$. Figure 4.6 illustrates the cycle. Due to for all choices the algorithm detects this cycle, the ontology is inconsistent.

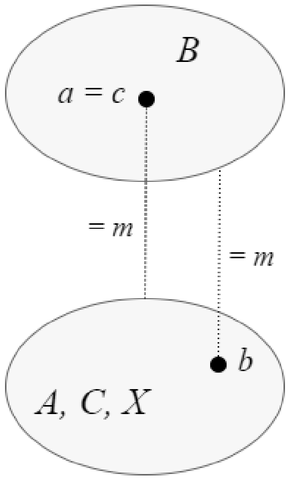

Figure 4.6: Cycle in the forest of Example 6

. . . .

4.1.3 Correctness of the tableau algorithm

This section presents the intuition and some details behind proofs of termination, soundness and completeness of the tableau algorithm described above. Main theorems and lemmas are introduced as well as the structures and principles that are used. For the complete set of proofs see [Motz15].

As in $\mathcal{SHIQ}$, the proof of *termination of the tableau algorithm for* $\mathcal{SHIQM}$ follows from the fact that the algorithm constructs a graph of interconnected root nodes and "trees" of blockable nodes rooted in some root node, and both the number of root nodes and the lenght of paths in the trees is finite. Even though the $\approx$-rule for $\mathcal{SHIQM}$ increases the Tbox, the amount of axioms added is a finite combination of the concept names in the Mbox.

Proofs of *soundness and completeness* make use of two main notions: *tableau* and set. The notion of *tableau for* $\mathcal{SHIQM}$ introduces an abstract model by extending the definition of tableau for $\mathcal{SHIQ}$ [Horrocks00]. The function set plays the role of associating each node of the forest with meta-modelling to the set of nodes that belong (are labelled with) to the corresponding concept.

The following subsections present, first of all, the definition of *tableau structure* and also the notion of *isomorphism between tableau structures.* The notion of tableau structure is independent of a description logic and an ontology. Based on this notion, the definition of *tableau* for $\mathcal{SHIQM}$ is introduced and moreover a lemma which shows that the consistency of an ontology in $\mathcal{SHIQM}$ is equivalent to having an abstract model, i.e. a tableau. Afterthat, definitions of *paths* in a completion forest and the function set are introduced, and moreover a *canonical model* built from the forest and the function set. Then, the proof of soundness shows that the canonical model constructed from the completion forest is a tableau. Finally, the proof of completeness defines a *structure preserving map* which maps each forest obtained from a rule application to a tableau for $\mathcal{SHIQM}$.

Consistency of an ontology in $\mathcal{SHIQM}$. The tableau structure.

Definition 17 (Tableau Structure)
Let $\mathbf{I}$ *and* $\mathbf{R}$ *be some arbitrary sets of individuals and roles respectively.* $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ *is a* tableau structure *for* $\mathbf{I}$ *and* $\mathbf{R}$ *if*

- $\mathbf{S}$ *is a non-empty set,*
- $\mathcal{L}$ *maps each element in* $\mathbf{S}$ *to a set of concepts,*
- $\mathcal{E} : \mathbf{R} \to 2^{\mathbf{S} \times \mathbf{S}}$ *maps each role to a set of pairs of elements in* $\mathbf{S}$*, and*
- $\mathcal{J} : \mathbf{I} \to \mathbf{S}$ *maps individuals to elements in* $\mathbf{S}$*.*

Definition 18 (Tableau for $\mathcal{SHIQM}$)
Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ *be a* $\mathcal{SHIQM}$ *ontology, with* $\mathbf{I}_{\mathcal{O}}$ *and* $\mathbf{R}_{\mathcal{O}}$ *the set of individuals and roles in* $\mathcal{O}$*.* $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ *is a tableau for* $\mathcal{O}$ *if*

1. $\mathbb{T}$ *is a tableau structure for* $\mathbf{I}_{\mathcal{O}}$ *and* $\mathbf{R}_{\mathcal{O}}$*, where* $\mathcal{J} : \mathbf{I}_{\mathcal{O}} \to \mathbf{S}$ *maps individuals occurring in* $\mathcal{A}$ *and* $\mathcal{M}$ *to elements in* $\mathbf{S}$*.*
2. $\mathbf{S} \subseteq S_n$ *for some* S_n*,*
3. *for all* $s, t \in \mathbf{S}$*,* $a, b \in \mathbf{I}_{\mathcal{O}}$*,* $R, S \in \mathbf{R}_{\mathcal{O}}$ *and concepts* C, C_1, C_2 *the properties (P1)-(P19) presented below hold. Moreover,* $\mathbb{T}' = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\restriction_{\mathbf{I}_{\mathcal{A}}})$ *is a tableau for the* $\mathcal{SHIQ}$*-ontology* $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ *which satisfies the properties (P1)-(P16), with* $\mathbf{I}_{\mathcal{A}}$ *the set of individuals in* $\mathcal{A}$*.*
   (P1) if $C \in \mathcal{L}(s)$*, then* $\neg C \notin \mathcal{L}(s)$*.*
   (P2) if $C_1 \sqcap C_2 \in \mathcal{L}(s)$ *then* $C_1 \in \mathcal{L}(s)$ *and* $C_2 \in \mathcal{L}(s)$*.*
   (P3) if $C_1 \sqcup C_2 \in \mathcal{L}(s)$ *then* $C_1 \in \mathcal{L}(s)$ *or* $C_2 \in \mathcal{L}(s)$*.*
   (P4) if $\forall S.C \in \mathcal{L}(s)$ *and* $(s,t) \in \mathcal{E}(S)$*, then* $C \in \mathcal{L}(t)$*.*
   (P5) if $\exists S.C \in \mathcal{L}(s)$*, then there is some* $t \in \mathbf{S}$ *such that* $(s,t) \in \mathcal{E}(S)$ *and* $C \in \mathcal{L}(t)$*.*
   (P6) if $\forall S.C \in \mathcal{L}(s)$ *and* $(s,t) \in \mathcal{E}(R)$ *for some* $R \sqsubseteq^* S$ *with* $Trans(R)$*, then* $\forall R.C \in \mathcal{L}(t)$*.*
   (P7) $(x,y) \in \mathcal{E}(R)$ *iff* $(y,x) \in \mathcal{E}(Inv(R))$*.*
   (P8) if $(s,t) \in \mathcal{E}(R)$ *and* $R \sqsubseteq^* S$*, then* $(s,t) \in \mathcal{E}(S)$*.*
   (P9) if $\leqslant n\ S.C \in \mathcal{L}(s)$*, then* $\sharp\{t \mid (s,t) \in \mathcal{E}(S) \text{ and } C \in \mathcal{L}(t)\} \leq n$*.*
   (P10) if $\geqslant n\ S.C \in \mathcal{L}(s)$*, then* $\sharp\{t \mid (s,t) \in \mathcal{E}(S) \text{ and } C \in \mathcal{L}(t)\} \geq n$*.*
   (P11) if $\leqslant n\ S.C \in \mathcal{L}(s)$ *or* $\geqslant n\ S.C \in \mathcal{L}(s)$*, and* $(s,t) \in \mathcal{E}(S)$*, then* $C \in \mathcal{L}(t)$ *or* $\sim C \in \mathcal{L}(t)$*.*
   (P12) if $C \in \mathcal{T}$ *then* $C \in \mathcal{L}(s)$ *for all* $s \in \mathbf{S}$*.*
   (P13) if $C(a) \in \mathcal{A}$*, then* $C \in \mathcal{L}(\mathcal{J}(a))$*.*
   (P14) if $R(a,b) \in \mathcal{A}$*, then* $(\mathcal{J}(a), \mathcal{J}(b)) \in \mathcal{E}(R)$*.*
   (P15) if $a \neq b \in \mathcal{A}$*, then* $\mathcal{J}(a) \neq \mathcal{J}(b)$*.*
   (P16) if $a = b \in \mathcal{A}$*, then* $\mathcal{J}(a) = \mathcal{J}(b)$*.*
   (P17) if $a =_{\mathsf{m}} A \in \mathcal{M}$*, then* $\mathcal{J}(a) = \{s \in \mathbf{S} \mid A \in \mathcal{L}(s)\}$*.*
   (P18) if $\mathcal{J}(a) = \mathcal{J}(b)$*,* $a =_{\mathsf{m}} A \in \mathcal{M}$ *and* $b =_{\mathsf{m}} B \in \mathcal{M}$*, then* $A \sqcup \neg B \in \mathcal{L}(s)$ *and* $B \sqcup \neg A \in \mathcal{L}(s)$ *for all* $s \in \mathbf{S}$*.*
   (P19) if $\mathcal{J}(a) \neq \mathcal{J}(b)$*,* $a =_{\mathsf{m}} A \in \mathcal{M}$ *and* $b =_{\mathsf{m}} B \in \mathcal{M}$*, then there is some* $t \in \mathbf{S}$ *such that* $(A \sqcap \neg B \sqcup B \sqcap \neg A) \in \mathcal{L}(t)$*.*

Note that properties (P17), (P18) and (P19) are added to rules (P1)-(P16) of the definition of tableau for $\mathcal{SHIQ}$ to deal with the meta-modelling statements.

The definition below is used to prove soundness of the tableau algorithm for $\mathcal{SHIQM}$; it allows reusing the proof of soundness for $\mathcal{SHIQ}$ by showing that $\mathcal{SHIQ}$ and $\mathcal{SHIQM}$ canonical models are isomorphic. The proof of soundness for $\mathcal{SHIQ}$ can be found in [Horrocks00].

Definition 19 (Isomorphism)
Let $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ and $\mathbb{T}' = (\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}')$ be two tableau structures for some $\mathbf{I}$ and $\mathbf{R}$. An isomorphism between $\mathbb{T}$ and $\mathbb{T}'$ is a bijective function $f : \mathbf{S} \to \mathbf{S}'$ such that

1. *$C \in \mathcal{L}(s)$ if and only if $C \in \mathcal{L}'(f(s))$.*
2. *$(s, t) \in \mathcal{E}(R)$ if and only if $(f(s), f(t)) \in \mathcal{E}'(R)$.*
3. *$f(\mathcal{J}(a)) = \mathcal{J}'(a)$.*

for all $s, t \in \mathbf{S}$, $a \in \mathbf{I}$, $R \in \mathbf{R}$ and concepts C. $\mathbb{T}$ and $\mathbb{T}'$ are isomorphic *if there exists an isomorphism between them.*

It is proved that *isomorphic tableau structures satisfy the same properties* and hence, if one is a model so is the other one [Motz15].

The following lemma is the basis for proofs of soundness and completeness.

Lemma 1 *Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ be a $\mathcal{SHIQM}$-ontology. $\mathcal{O}$ is consistent iff there exists a $\mathcal{SHIQM}$-tableau for $\mathcal{O}$.*

To prove the $\Leftarrow$ direction it is enough to show that given a tableau $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ for $\mathcal{O}$, the interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ described below, with $\Delta^{\mathcal{I}} := \mathbf{S}$, is a model of $\mathcal{O}$.

$$\begin{array}{lll} A^{\mathcal{I}} & := & \{s \in \mathbf{S} \mid A \in \mathcal{L}(s)\} \\ a^{\mathcal{I}} & := & \mathcal{J}(a) \\ R^{\mathcal{I}} & := & \begin{cases} \mathcal{E}(R)^{+} & \text{if } Trans(R) \\ \mathcal{E}(R) \cup \bigcup_{P \sqsubseteq^{*} R, P \neq R} P^{\mathcal{I}} & \text{otherwise} \end{cases} \end{array}$$

where $\mathcal{E}(R)^{+}$ is the transitive closure of $\mathcal{E}(R)$. The first condition of the Definition 8 of model for $\mathcal{SHIQM}$ coincides with the second condition of Definition 18. For the second and third conditions, first, it is proved that $C \in \mathcal{L}(s)$ implies $s \in C^{\mathcal{I}}$, and by using the tableau properties (P1)-(P19) it is also showed that $\mathcal{I}$ satisfies the Tbox, Abox, Rbox and the Mbox of $\mathcal{O}$. For example, the third condition of Definition 8 holds since by (P17) and the definition of $\mathcal{I}$:

$$a^{\mathcal{I}} = \mathcal{J}(a) = \{s \in \mathbf{S} \mid A \in \mathcal{L}(s)\} = A^{\mathcal{I}}$$

Given a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{O}$, for the proof of the $\Rightarrow$ direction a tableau $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is defined as follows.

$$\begin{array}{lll} \mathbf{S} := \Delta^{\mathcal{I}} & & \\ \mathcal{L}(s) & := & \{C \in \mathsf{clos}(\mathcal{O}) \mid s \in C^{\mathcal{I}}\} \\ \mathcal{E}(R) & := & R^{\mathcal{I}} \\ \mathcal{J}(a) & := & a^{\mathcal{I}} \end{array}$$

where clos is defined as:

$$\mathsf{clos}(\mathcal{O}) = \bigcup_{C(a)\in\mathcal{A} \text{ or } C\in\mathcal{T}\cup\mathsf{concepts}(\mathcal{M})} \mathsf{clos}(C)$$

with

$\mathsf{concepts}(\mathcal{M}) = \{A \sqcap \neg B \sqcup B \sqcap \neg A, A \sqcup \neg B, B \sqcup \neg A \mid a =_{\mathsf{m}} A, b =_{\mathsf{m}} B \in \mathcal{M}\}$ and $\mathsf{clos}(C)$ defined as the smallest set that contains the concept C (assumed to be in negation normal form) and is closed under (syntactic) sub-concepts and $\sim$. $\sim C$ is the negation normal form of $\neg C$ (see Section 3 of [Motz15]). Then, from Definition 8 of model for $\mathcal{SHIQM}$ follows that $\mathbb{T}$ is a tableau (satisfies properties (P1) to (P19)). For example, to prove (P17), if $a =_{\mathsf{m}} A$, by the third condition of Definition 8 $A^{\mathcal{I}} = a^{\mathcal{I}}$ and also:

$$a^{\mathcal{I}} = \mathcal{J}(a) = \{s \in \Delta^{\mathcal{I}} \mid s \in A^{\mathcal{I}}\} = \{s \in \mathbf{S} \mid A \in \mathcal{L}(s)\}$$

Soundness for $\mathcal{SHIQM}$. The function set.

As in $\mathcal{SHIQ}$, to prove soundness of the tableau algorithm for $\mathcal{SHIQM}$ a canonical model is constructed from the complete forest without contradictions nor cycles obtained by applying the rules of figures 4.3 and 4.4. Then, it is showed that such canonical structure is a tableau, according to Definition 18. However, it is also required to connect nodes for individuals with meta-modelling to the set of nodes that interpret the corresponding concept. Hence, the function set is introduced to set this connection.

The *abstract canonical* model of a $\mathcal{SHIQM}$ ontology is built as the composition of two interpretations: the abstract $\mathcal{SHIQ}$-canonical interpretation (canonical model of the ontology without meta-modelling) and the function set that computes the set associated to an individual with meta-modelling recursively.

$$\begin{array}{lll}
\mathcal{SHIQM} & & \\
\quad\big\downarrow \mathcal{J} & \textit{Abstract Canonical Model for } \mathcal{SHIQ} & \\
\mathcal{J}' \quad \mathbf{S} = \mathsf{Paths}(\mathcal{F}) & & \textit{Domain for } \mathcal{SHIQ} \\
\quad\cong\big\downarrow \mathsf{set} & \textit{Recursive Computation of Sets} & \\
\mathbf{S}' = \mathsf{set}(\mathsf{Paths}(\mathcal{F})) & & \textit{Domain for } \mathcal{SHIQM}
\end{array}$$

The domain $\mathbf{S}$ of the tableau built from the completion forest $\mathcal{F}$ of the $\mathcal{SHIQ}$ ontology is the set of paths in $\mathcal{F}$ while the domain $\mathbf{S}'$ for the tableau of the $\mathcal{SHIQM}$ ontology consists of paths, sets of paths, sets of sets of paths, and so on. The idea of the function set is to associate the set of objects that an individual with meta-modelling represents. Before introducing definitions of set and the canonical structure for $\mathcal{SHIQM}$ the notion of *path* is formally defined below.

A *path* is a sequence of pairs of nodes of $\mathcal{F}$ of the form $p = \left[\frac{x_0}{x'_0}, \dots, \frac{x_n}{x'_n}\right]$, with $\mathsf{Tail}(p) = x_n$ and $\mathsf{Tail}'(p) = x'_n$. The path $\left[\frac{x_0}{x'_0}, \dots, \frac{x_n}{x'_n}, \frac{x_{n+1}}{x'_{n+1}}\right]$ is denoted as $\left[p \mid \frac{x_{n+1}}{x'_{n+1}}\right]$. The set $\mathsf{Paths}(\mathcal{F})$ is defined inductively as follows: for a root node a in $\mathcal{F}$ which is a representative, $\left[\frac{a}{a}\right] \in \mathsf{Paths}(\mathcal{F})$, and for a path $p \in \mathsf{Paths}(\mathcal{F})$ and a node z in $\mathcal{F}$

representative of some equivalence class: (i) if z is a successor of $\mathsf{Tail}(p)$ and z is neither blocked nor a root node , then $\left[p \mid \frac{z}{z}\right] \in \mathsf{Paths}(\mathcal{F})$, or (ii) if, for some node y in $\mathcal{F}$, y is a successor of $\mathsf{Tail}(p)$ and z blocks y, then $\left[p \mid \frac{z}{y}\right] \in \mathsf{Paths}(\mathcal{F})$.

The function set is defined as follows.

Definition 20 (From Basic Paths to Sets)
Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ and let $\mathcal{F}$ be a complete completion forest without contradictions nor cycles w.r.t. $\mathcal{M}$. For $p \in \mathsf{Paths}(\mathcal{F})$ we define $\mathsf{set}(p)$ as follows.

$$\begin{array}{rl} \mathsf{set}(p) = & \{\mathsf{set}(q) \mid A \in \mathcal{F}(\mathsf{Tail}(q))\} \\ & \quad \textit{if } p = [\frac{c}{c}] \textit{ for some } c \approx a =_{\mathsf{m}} A \in \mathcal{M} \\ \mathsf{set}(p) = & p \textit{ otherwise} \end{array}$$

Note that set only changes the paths of the form $p = [\frac{c}{c}]$ such that c is a representative of an individual with meta-modelling (recall that only root nodes can have meta-modelling). The other paths keep unchanged, i.e. the function set acts as the identity. The following example illustrates the idea behind the function set. For simplicity, paths of the form $p = [\frac{c}{c}]$ are just denoted with c, as all of the nodes in the example are root nodes.

Example 7. We consider the ontology network of Figure 4.1 and the following interpretation $\mathcal{I}$ which does not consider meta-modelling:

$$\begin{array}{lcl} \Delta^{\mathcal{I}} & = & \{queguay, santaLucia, delSauce, deRocha, river, lake\} \\ River^{\mathcal{I}} & = & \{queguay, santaLucia\} \\ Lake^{\mathcal{I}} & = & \{delSauce, deRocha\} \\ Hydrographic^{\mathcal{I}} & = & \{river, lake\} \end{array}$$

If we consider meta-modelling, interpretations of individuals *river* and *lake* (see Example 1) are the sets given by the function set as follows:

$$\begin{array}{lcl} \mathsf{set}(river) & = & \{queguay, santaLucia\} \\ \mathsf{set}(lake) & = & \{delSauce, deRocha\} \end{array}$$

Suppose an individual *hydrographic* equated to the concept *Hidrographic* by meta-modelling ($hydrographic =_{\mathsf{m}} Hydrographic$). The set associated to *hydrographic* is a set of sets as follows.

$$\begin{array}{lcl} \mathsf{set}(hydrographic) & = & \{\{queguay, santaLucia\}, \\ & & \{deRocha, delSauce\}\} \end{array}$$

However, as individuals *queguay*, *santaLucia*, *delSauce* and *deRocha* do not have meta-modelling:

$$\begin{array}{lcl} \mathsf{set}(queguay) & = & queguay \\ \mathsf{set}(santaLucia) & = & santaLucia \\ \mathsf{set}(delSauce) & = & delSauce \\ \mathsf{set}(deRocha) & = & deRocha \end{array}$$

Hence, the canonical interpretation $\mathcal{I}_{\mathsf{m}}$ for the ontology with meta-modelling is as follows:

$$\begin{array}{lll}
\Delta^{\mathcal{I}_{\mathsf{m}}} & = & \mathsf{set}(\Delta^{\mathcal{I}}) \\
 & = & \{\{queguay, santaLucia\}, \{delSauce, deRocha\}, \\
 & & \{\{queguay, santaLucia\}, \{deRocha, delSauce\}\}, \\
 & & queguay, santaLucia, delSauce, deRocha\} \\
(River)^{\mathcal{I}_{\mathsf{m}}} & = & \{queguay, santaLucia\} \\
(Lake)^{\mathcal{I}_{\mathsf{m}}} & = & \{deRocha, delSauce\} \\
(Hydrographic)^{\mathcal{I}_{\mathsf{m}}} & = & \{\{queguay, santaLucia\}, \{deRocha, delSauce\}\}
\end{array}$$

By defining $S_0 = \{queguay, santaLucia, delSauce, deRocha\}$ we have that:

$$\Delta^{\mathcal{I}_{\mathsf{m}}} \subset S_2$$

. . . .

Two important properties are proved for set: (i) set is a *correct recursive definition* and (ii) set is an *injective function*.

To prove that set is a *correct recursive definition* the relations $\prec$ and $\ll$ are defined. The relation $\prec$ on the set of nodes of the forest $\mathcal{F}$ is defined as $b \prec c$ iff $A \in \mathcal{F}(b)$, $c \approx a$ and $a =_{\mathsf{m}} A \in \mathcal{M}$. The relation $\ll$ on the set of paths $\mathsf{Paths}(\mathcal{F})$ is defined as $q \ll p$ iff $\mathsf{Tail}(q) \prec c$ and $p = [\frac{c}{c}]$. It is showed that if $\mathcal{F}$ has no cycles w.r.t. $\mathcal{M}$ then $\prec$ is well-founded, and from this result $\ll$ is proved to be well-founded. As $\ll$ is well-founded it is possible to define the function set recursively, with $q \ll p$ in the recursive step of that definition, which complete the proof that set is a *correct recursive definition*. Moreover, it is showed that if $S_0 = \mathsf{Paths}(\mathcal{F})$, then for all $p \in \mathsf{Paths}(\mathcal{F})$, $\mathsf{set}(p) \in S_{\sharp(\mathcal{M})}$.

To prove that set is an *injective function* the fact that $\ll$ is well-founded is also used. Hence, as set is an isomorphism, from a complete forest that has neither contradictions nor cycles it is possible to build a $\mathcal{SHIQM}$ canonical structure based on the canonical structure for $\mathcal{SHIQ}$. The definition of set is a key point for defining a canonical structure from a completion forest for $\mathcal{SHIQM}$.

Definition 21 ($\mathcal{SHIQM}$ Canonical Structure)
Let $\mathcal{F}$ be a completion forest for a $\mathcal{SHIQM}$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$. A canonical tableau structure $\mathbb{T}' = (\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}')$ is built from $\mathcal{F}$ as follows:

$$\begin{array}{ll}
\mathbf{S}' = & \{\mathsf{set}(p) \mid p \in \mathbf{S}\} \\
\mathcal{L}'(s) = & \mathcal{L}(p) \text{ with } s = \mathsf{set}(p) \\
\mathcal{E}'(R) = & \{(\mathsf{set}(p), \mathsf{set}(q)) \in \mathbf{S}' \times \mathbf{S}' \mid (p,q) \in \mathcal{E}(R)\} \\
\mathcal{J}'(a) = & \mathsf{set}(\mathcal{J}(a))
\end{array}$$

where $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is the canonical tableau structure for $\mathcal{SHIQ}$ defined below.

$$\begin{array}{ll}
\mathbf{S} = & \mathsf{Paths}(\mathcal{F}) \\
\mathcal{L}(p) = & \mathcal{F}(\mathsf{Tail}(p)) \\
\mathcal{E}(R) = & \{(p, \left[p \mid \frac{x}{x'}\right]) \in \mathbf{S} \times \mathbf{S} \mid x' \text{ is a } R\text{-successor of } \mathsf{Tail}(p)\} \cup \\
 & \{(\left[q \mid \frac{x}{x'}\right], q) \in \mathbf{S} \times \mathbf{S} \mid x' \text{ is an } Inv(R)\text{-successor of } \mathsf{Tail}(q) \cup \\
 & \{(\left[\frac{a}{a}\right], \left[\frac{b}{b}\right]) \in \mathbf{S} \times \mathbf{S} \mid a,\ b \text{ are representative root nodes and} \\
 & \ b \text{ is a } R\text{-neighbour of } a\} \\
\mathcal{J}(a) = & \begin{cases} \left[\frac{a}{a}\right] \text{ if } a \text{ is itself a representative} \\ \left[\frac{b}{b}\right] \text{ if } b \text{ is the representative of } a \approx b \end{cases}
\end{array}$$

Note that while $\mathbf{S}$ of the tableau $\mathbb{T}$ for $\mathcal{SHIQ}$ is a set of paths [Horrocks00], the domain $\mathbf{S}'$ of the tableau $\mathbb{T}'$ for $\mathcal{SHIQM}$ consists of paths, sets of paths, sets of sets of paths, and so on. Moreover, since the tableau structure $\mathbb{T}$ is built from the completion forest $\mathcal{F}$, the domains of $\mathcal{J}$ and $\mathcal{J}'$ are both the set of individuals in the ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ which includes the individuals occurring in the MBox.

The theorem of *soundness* is enunciated as follows.

Theorem 1 ($\mathcal{SHIQM}$ Abstract Canonical Model) *Let a $\mathcal{SHIQM}$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$. If the expansion rules for $\mathcal{SHIQM}$ can be applied to $\mathcal{O}$ in such a way that they yield a complete completion forest $\mathcal{F}$ that has no contradictions and has no cycles w.r.t. $\mathcal{M}$ then $\mathcal{O}$ is consistent*

From Lemma 1 a $\mathcal{SHIQM}$ ontology is consistent iff it has a tableau, then it is enough to prove that the tableau structure given in Definition 21 is a tableau for the $\mathcal{SHIQM}$-ontology $\mathcal{O}$. The first and second conditions of Definition 18 of tableau for $\mathcal{SHIQM}$ follow directly from Definition 21 of the canonical structure and from the fact that:

$$\mathbf{S}' = \mathsf{set}(\mathsf{Paths}(\mathcal{F})) \subseteq S_{\sharp(\mathcal{M})}$$

For the third condition, given that $(\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\restriction_{\mathbf{I}_\mathcal{A}})$ and $(\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}'\restriction_{\mathbf{I}_\mathcal{A}})$ are isomorphic tableau structures (set is an injective function) and moreover the rules preserve all properties, it is showed that $(\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\restriction_{\mathbf{I}_\mathcal{A}})$ and $(\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}'\restriction_{\mathbf{I}_\mathcal{A}})$ satisfy properties (P1)-(P16). Properties (P17)-(P19) for $\mathbb{T}' = (\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}')$ are proved by using definitions of $\mathcal{SHIQM}$ canonical model and then function set. For example, to prove (P17), if $a =_\mathsf{m} A$ then:

$$\begin{aligned}
\mathcal{J}'(a) &= \mathsf{set}(\mathcal{J}(a)) \\
&= \mathsf{set}(\left[\tfrac{b}{b}\right]) \text{ for } b \approx a \\
&= \{\mathsf{set}(q) \mid A \in \mathcal{F}(\mathsf{Tail}(q)\} \\
&= \{\mathsf{set}(q) \mid A \in \mathcal{L}(q)\} \\
&= \{\mathsf{set}(q) \mid A \in \mathcal{L}'(\mathsf{set}(q))\}
\end{aligned}$$

Completeness for $\mathcal{SHIQM}$.

The proof of completeness of the tableau algorithm described in Section 4.1.2 shows that, given a $\mathcal{SHIQM}$ ontology that has a tableau $\mathbb{T}$, the application of the expansion rules for $\mathcal{SHIQM}$ (figures 4.3 and 4.4) leads to a complete $(\mathcal{T}, \mathcal{F})$ without contradictions nor cycles. We have to show that the application of the expansion rules for $\mathcal{SHIQM}$ (figures 4.3 and 4.4) leads to a complete $(\mathcal{T}, \mathcal{F})$ without contradictions nor cycles. To this aim, Lemma 1 is applied and moreover the notion of structure preserving map defined below is used.

Definition 22 (Structure preserving map)
Let $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ be a $\mathcal{SHIQM}$-tableau for a $\mathcal{SHIQM}$-ontology $\mathcal{O}$ and $\mathcal{F}$ a completion forest. A structure preserving map $\pi : \mathcal{F} \to \mathbb{T}$ is defined as a function π from the set of nodes of $\mathcal{F}$ to $\mathbf{S}$ which satisfies the following conditions:

1. *$\mathcal{F}(x) \subseteq \mathcal{L}(\pi(x))$.*
2. *If y is a S-neighbour of x, then $(\pi(x), \pi(y)) \in \mathcal{E}(S)$.*

3. $x \not\approx y$ *implies* $\pi(x) \neq \pi(y)$.

4. $x \approx y$ *implies* $\pi(x) = \pi(y)$.

for all nodes x, y *in* $\mathcal{F}$.

Given a tableau $\mathbb{T}$, It is showed that if the applicaction of a rule to $(\mathcal{T}_1, \mathcal{F}_1)$ leads to $(\mathcal{T}_2, \mathcal{F}_2)$, there is a structure preserving map from $\mathcal{F}_2$ to $\mathbb{T}$ that extends the structure preserving map from $\mathcal{F}_1$ to $\mathbb{T}$ [Motz15]. This result is used to prove the theorem of completeness enunciated below.

Theorem 2 (Completeness) *Let* $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ *be a* $\mathcal{SHIQM}$*-ontology. If* $\mathcal{O}$ *is consistent, then the expansion rules for* $\mathcal{SHIQM}$ *can be applied to* $\mathcal{O}$ *such that they yield a complete completion forest with no contradictions and no cycles w.r.t.* $\mathcal{M}$.

The proof of the theorem above uses the result of Lemma 1, i.e. that $\mathcal{O}$ has a tableau. The proof starts by defining an initial structure preserving map for the forest after the initialization and shows that it satisfies the conditions of Definition 22. After that, to show that a completion forest without contradictions nor cycles is obtained, it is used the fact that for each applicaction of a rule, there is a structure preserving map that extends the one of the previous application, and moreover that the algorithm always terminates.

4.1.4 The meta-modelling level in $\mathcal{SHIQM}$

This section formalizes the notion of *meta-modelling level* for $\mathcal{SHIQM}$. Note that for meta-modelling approaches that follow a semantics with fixed layers, the meta-modelling level of an ontology is given by the maximum typing number assigned to concepts in the ontology (see Henkin fixed layered and Hilog fixed layered approaches of Chapter 2). Moreover, all models of the ontology have this meta-modelling level. However, as the meta-modelling approach introduced in the present work defines a global interpretation domain with flexible layers and the levels are not explicitly annotated, it is not so direct to calculate the meta-modelling level for a given ontology and for all their possible models. Notions presented below are related to the meta-modelling level of an ontology and of a given interpretation of an ontology [Motz15].

Definition 23 (Meta-modelling Level)
Let $\mathcal{O}$ *be a* $\mathcal{SHIQM}$ *ontology,* C *a concept and* $\mathcal{I}$ *a model of* $\mathcal{O}$.
The meta-modelling level of $\mathcal{I}$ *- denoted as* $\mathsf{level}(\mathcal{I})$ *- is the smallest* n *such that* $\Delta^{\mathcal{I}} \subseteq S_n$.
The concept C *is* at level n in the interpretation $\mathcal{I}$ *- denoted as* $\mathsf{level}(\mathcal{I}, C)$ *- if* n *is the smallest natural number such that* $C^{\mathcal{I}} \subseteq S_n$.
The meta-modelling level of $\mathcal{O}$ *- denoted as* $\mathsf{level}(\mathcal{O})$ *- is the smallest* n *where* n *is the level of some model of* $\mathcal{O}$.
The concept C *in the ontology* $\mathcal{O}$ *is* at level n *- denoted as* $\mathsf{level}(\mathcal{O}, C)$ *- if* n *is the smallest natural number such that* $C^{\mathcal{I}} \subseteq S_n$ *and* $\mathcal{I}$ *is a model of* $\mathcal{O}$.

The following example clarifies the different notions of meta-modelling levels defined above.

Example 8. Let $\mathcal{O}$ be the ontology of Figure 4.1 with the set of axioms of Figure 4.2. For simplicity, Abox axioms asserting that all individuals are different from each other are not included, even though this is what makes sense in a real setting. The model defined below interprets all individuals as different domain elements.

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \{queguay, santaLucia, delSauce, deRocha,\\
&\quad \{queguay, santaLucia\}, \{delSauce, deRocha\}\}\\
queguay^{\mathcal{I}} &= queguay\\
santaLucia^{\mathcal{I}} &= santaLucia\\
delSauce^{\mathcal{I}} &= delSauce\\
deRocha^{\mathcal{I}} &= deRocha\\
river^{\mathcal{I}} &= River^{\mathcal{I}} = \{queguay, santaLucia\}\\
lake^{\mathcal{I}} &= Lake^{\mathcal{I}} = \{delSauce, deRocha\}\\
HydrographicObject^{\mathcal{I}} &= \{river^{\mathcal{I}}, lake^{\mathcal{I}}\}\\
&= \{\{queguay, santaLucia\}, \{delSauce, deRocha\}\}
\end{aligned}$$

According to Definition 23, $\mathsf{level}(\mathcal{I}) = 1$, $\mathsf{level}(\mathcal{I}, River) = \mathsf{level}(\mathcal{I}, Lake) = 0$, $\mathsf{level}(\mathcal{I}, HydrographicObject) = 1$, and $\mathsf{level}(\mathcal{I}, \top) = 1$.
Suppose now that the assertion that set individuals *lake* and *queguay* as different is dropped and consider the interpretation below where individuals *lake* and *queguay* are the same object.

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \{santaLucia, delSauce, deRocha, \{delSauce, deRocha\},\\
&\quad \{\{delSauce, deRocha\}, santaLucia\}\}\\
santaLucia^{\mathcal{I}} &= santaLucia\\
delSauce^{\mathcal{I}} &= delSauce\\
deRocha^{\mathcal{I}} &= deRocha\\
lake^{\mathcal{I}} &= queguay^{\mathcal{I}} = Lake^{\mathcal{I}} = \{delSauce, deRocha\}\\
river^{\mathcal{I}} &= River^{\mathcal{I}} = \{queguay^{\mathcal{I}}, santaLucia\}\\
&= \{\{delSauce, deRocha\}, santaLucia\}\\
HydrographicObject^{\mathcal{I}} &= \{river^{\mathcal{I}}, lake^{\mathcal{I}}\}\\
&= \{\{queguay^{\mathcal{I}}, santaLucia\}, \{delSauce, deRocha\}\}\\
&= \{\{delSauce, deRocha\}, santaLucia\},\\
&\quad \{delSauce, deRocha\}\}
\end{aligned}$$

Note that even though the interpretation above does not correspond to a real scenario, it is a model according to Definition 8, but in this case $\mathsf{level}(\mathcal{I}) = 2$, $\mathsf{level}(\mathcal{I}, Lake) = 0$, $\mathsf{level}(\mathcal{I}, River) = 1$, $\mathsf{level}(\mathcal{I}, HydrographicObject) = 2$ and $\mathsf{level}(\mathcal{I}, \top) = 2$.
Regarding the level of $\mathcal{O}$, it is easy to see that $\mathsf{level}(\mathcal{O}) = 1$, $\mathsf{level}(\mathcal{O}, River) = \mathsf{level}(\mathcal{O}, Lake) = 0$ and $\mathsf{level}(\mathcal{O}, HydrographicObject) = 1$.

. . . .

An algorithm to compute the meta-modelling level of an ontology would need to build all the complete and consistent tableau forests and choose the model that has minimum level. Since this method is very inefficient, the present work presents a different algorithm that gives a range of values where the meta-modelling level

belongs.
The level of the model $\mathcal{I}$ given by the tableau algorithm is bounded by the cardinality of the Mbox, which follows since given $S_0 = \mathsf{Paths}(\mathcal{F})$, for all $p \in \mathsf{Paths}(\mathcal{F})$, $\mathsf{set}(p) \in S_{\sharp(\mathcal{M})}$ (see Section 4.1.3 in the present chapter) [Motz15]. Hence $\mathsf{level}(\mathcal{O}) \leq \mathsf{level}(\mathcal{I}) \leq \sharp(\mathcal{M})$. Then, a function lb that computes a lower bound for $\mathsf{level}(\mathcal{O})$ and $\mathsf{level}(\mathcal{O}, C)$ for a concept C, is recursively defined as follows.

Definition 24 (Lower Bound)
Let $\mathcal{O}$ be a $\mathcal{SHIQM}$ ontology, a an individual and C a concept in $\mathcal{O}$. The function lb *is defined for a, $\mathcal{O}$ and C as follows:*

$$\begin{array}{ll}
\mathsf{lb}(a) & = 0 \textit{ if there is no } A \textit{ such that } \mathcal{O} \models a =_{\mathsf{m}} A \\
\mathsf{lb}(a) & = max\{\mathsf{lb}(b) \mid \mathcal{O} \models a =_{\mathsf{m}} A, \mathcal{O} \models A(b)\}) + 1 \\
\mathsf{lb}(\mathcal{O}) & = max\{\mathsf{lb}(a) \mid a \in \mathcal{O}\} \\
\mathsf{lb}(\mathcal{O}, C) & = max\{\mathsf{lb}(a) \mid \mathcal{O} \models C(a)\}
\end{array}$$

It is proved that $\mathsf{lb}(\mathcal{O}, C) \leq \mathsf{level}(\mathcal{O}, C)$ and $\mathsf{lb}(\mathcal{O}) \leq \mathsf{level}(\mathcal{O})$ for C a concept in $\mathcal{O}$ [Motz15]. Then, the present work proposes the algorithm described below to compute a range of values where the meta-modelling level of an ontology belongs.

1. Run tableau for checking consistency of the ontology and getting a model $\mathcal{I}$.

2. Let $n = \mathsf{lb}(\mathcal{O})$ and $m = \mathsf{level}(\mathcal{I})$.

   If $n = m$ then the level of the ontology is n

   Otherwise the level of the ontology is between n and m.

Something similar can be done for the meta-modelling level of a concept by substituting $\mathsf{lb}(\mathcal{O})$ by $\mathsf{lb}(\mathcal{O}, C)$ and $\mathsf{level}(\mathcal{I})$ by $\mathsf{level}(\mathcal{I}, C)$.

Note that for real domains of rather closed worlds usually there are concrete requirements that lead to include a lot of restrictions in the Abox, Tbox, Rbox and Mbox of the ontology. This causes that the lower bound of the computed meta-modelling level for the ontology coincides with the meta-modelling level of any interpretation, i.e. $n = m$, as Example 8 shows for the case that all individuals are different.

4.1.5 Representing case studies with $\mathcal{SHIQM}$

This section analyzes the expressivity of the description logic $\mathcal{SHIQM}$ to represent different domains with meta-modelling. The description logic $\mathcal{SHIQM}$ is a *Henkin-global layered approach* according to the classification of semantics approaches for meta-modelling introduced in Chapter 2. The semantics of $\mathcal{SHIQM}$ interprets the set of axioms in an ontology into a single *global layered domain* $\Delta \subseteq S_n$ such that $S_{n+1} = S_n \cup \mathcal{P}(S_n)$ with S_0 a set of atomic objects. The definition of S_n enables interpreting a concept C as a set of objects which belong to different meta-modelling levels (e.g. the interpretation of the concept *Hydrographic* for the last scenario of Example 8), and also ensures the well-foundedness of the domain since S_n is a well-founded set. Moreover, due to the Henkin semantics satisfies intensional regularity and extensionality, different perspectives of the same domain can be properly represented. This is because the equality (and difference) between objects represented for different perspectives, with different granularity, is kept consistent.

Below case studies of Chapter 3 are modelled by using the description logics $\mathcal{SHIQM}$.

Geographic objects in Uruguay The geographic domain described in Chapter 3 is depicted in Figure 3.1. The description logic $\mathcal{SHIQM}$ allows representing meta-modelling correspondences such as the one between the individual *river* and the concept *River* by meta-modelling statements of the form $river =_{\mathsf{m}} River$.

Given that $\mathcal{SHIQM}$ is a global layered approach, all concepts and roles in Figure 3.1 can be represented without having to type them with the meta-modelling level. Concepts *River*, *Lake*, *Wetland*, *NaturalForest*, *Activity* and *GovernmentOffice* have level 0 since intuitively all their instances are atomic. In fact, according to Definition 23 $\mathsf{level}(\mathcal{O}, River) = 0$ since for some models $\mathcal{I}$ of the ontology $\mathcal{O}$ represented in Figure 3.1 $River^{\mathcal{I}} \subseteq S_0$ holds. Following the same criteria, concepts *HydrographicObject* and *FloraObject* are meta-concepts with level 1 since $HydrographicObject^{\mathcal{I}} \subseteq S_1$, i.e. $n = 1$ is the minimum n such that for some model $\mathcal{I}$, $HydrographicObject^{\mathcal{I}} \subseteq S_n$[4]. Intuitively, all their instances are concepts, i.e. individuals equated by meta-modelling to concepts of level 0. However, even though the concept *GeographicObject* has level 2, i.e. $GeographicObject^{\mathcal{I}} \subseteq S_2$, not all their instances are meta-concepts. In fact, $hydrographic^{\mathcal{I}} = HydrographicObject^{\mathcal{I}}$ and $flora^{\mathcal{I}} = FloraObject^{\mathcal{I}}$ belong to S_2 whereas $physiographic^{\mathcal{I}}$ belongs to S_0. Moreover, it is possible to introduce a new concept by combining concepts of different layers, e.g. $Activity \sqcup FloraObject$, with $Activity^{\mathcal{I}} \cup FloraObject^{\mathcal{I}} \subseteq S_1$. Inter-layer roles such as *over* can be also represented, $over^{\mathcal{I}} \subseteq S_0 \times S_1$.

The definition of S_n ensures that the domain of interpretation $\Delta \subseteq S_n$ is well-founded. Suppose the axiom $FloraObject \sqsubseteq NaturalForest$ is added to the ontology of Figure 3.1. For any interpretation $\mathcal{I}$, it holds that $naturalForest^{\mathcal{I}} \in FloraObject^{\mathcal{I}} \subseteq NaturalForest^{\mathcal{I}}$. It causes that the domain Δ turns out non well-founded (*naturalForest* belong to itself), and moreover is is not longer a subset of S_n since $FloraObject^{\mathcal{I}} \subseteq S_1$ and $NaturalForest^{\mathcal{I}} \subseteq S_0$.

Due to $\mathcal{SHIQM}$ is a Henkin semantics, it prevents from inconsistencies such the one of Example 1 in Section 3.2.2. The property of extensionality ensure that the equality $Wetland \equiv NaturalForest$ is transferred to the level of individuals, and then the functionality of the role *associatedWith* forces the equality $river = lake$, which by the intensional regularity causes an inconsistency since concepts *River* and *Lake* are disjoint.

Health This case study does not present additional requirements with respect to the ones of the geographic domain, hence the $\mathcal{SHIQM}$ approach is able to represent the model proposed in Chapter 3, illustrated in Figure 3.7. The main benefit of the description logic $\mathcal{SHIQM}$ for the health domain is that the Henkin semantics ensures the consistency of those objects represented in different perspectives with different granularity, e.g. avoiding non expected results such us the one presented in Example 2 of Section 3.3.2.

Education Besides meta-modelling correspondences introduced by meta-modelling statements such as $module =_{\mathsf{m}} Module$, all concepts and roles in the model of Fig-

[4]From now on, for simplicity it is considered the interpretation $\mathcal{I}$ such that for a concept C $\mathsf{level}(\mathcal{O}, C) = \mathsf{level}(\mathcal{I}, C)$ holds.

ure 3.8 can be represented with $\mathcal{SHIQM}$. In particular, concepts *Activity* and *Service* in the level 2 have instances of different meta-modelling levels. For example, $Activity^{\mathcal{I}} \subseteq S_2$ (for interpretations $\mathcal{I}$ such that $\mathsf{level}(\mathcal{I}, Activity) = \mathsf{level}(\mathcal{O}, Activity) = 2$) and for concepts *Conference* and *Module* it holds that $Conference^{\mathcal{I}} \subseteq S_0$ and $Module^{\mathcal{I}} \subseteq S_1$.

Likewise, the well-foundedness of the interpretation domain and the coherence of equality relations between objects at different levels is also ensured as follows from the semantics of $\mathcal{SHIQM}$. However, for the educational domain not only equality relations of objects must be transfered between levels. Moreover, relations between individuals given by roles must also be transferred as relations between corresponding concepts (by meta-modelling) that are also given by roles. For example, in the scenario of Figure 3.8, the user teacher is who decides associating modules (e.g. *basicProgramming*) to different work environments (e.g. *classroom* and *web*), and this decision must hold at the level of students. Then, instances the role *uses* at the teacher level must be transferred to the role *attends* that relates corresponding concepts by meta-modelling (*DBProgRegitration* to *ClassEnv* and *WebEnv*). The description logic $\mathcal{SHIQM}$ cannot express this kind of correspondences between relations of different levels. The only way to ensure this is by introducing Tbox axioms such as $DBProgRegitration \sqsubseteq \forall attends.(ClassEnv \sqcup WebEnv)$. However, it would be needed to introduce one such axiom at the student level for each Abox axiom such as $uses(basicProgramming, web)$ at the teacher level, which becomes cumbersome for the great amount of modules in the educational scenario.

Accounting For the accounting domain illustrated in Figure 3.11, the same considerations as for the educational domain are valid regarding the expressiveness of $\mathcal{SHIQM}$ to represent the scenario. In particular, for the accounting scenario the requirement that accounting definitions (or "rules") of experts introduced at the level of instances must be transferred to the operational level is indeed a relevant requirement not solved by $\mathcal{SHIQM}$. As is analized at the end of Section 3.5.3, it would turn easier to validate such definitions by introducing Tbox axioms at the expert level, avoiding the execution of the same validations at the operational level.

Table 4.1 shows almost the same as table 3.4 in Chapter 3, but now with the right lower quadrant filled by the Henkin-global layered approach $\mathcal{SHIQM}$. Even though $\mathcal{SHIQM}$ solves most of the requirements of the case studies described in Chapter 3, it remainds to solve the requirement of transferring definitions or rules from higher to lower levels.

4.2 The description logic $\mathcal{SHIQM}^*$

The main motivation for the description logics $\mathcal{SHIQM}^*$ is to solve the last requirement that is in all quadrants of Table 4.1, which is not solved either by the description logics $\mathcal{SHIQM}$ presented above. $\mathcal{SHIQM}$ does not allow that definitions or rules set by users at higher levels become restrictions on the data accessed by users at lower levels. The description logic $\mathcal{SHIQM}^*$ extends $\mathcal{SHIQM}$ by adding a new meta-modelling statement just to do this: to translate rules introduced in the

Intensions-extensions / Domain layers	Fixed	Global
Hilog	**Concepts and roles not all represented** Well-foundedness ensured **Concepts equiv. not transf. to individuals** **Relations not transferred between levels**	Concepts and roles all represented **Well-foundedness not ensured** **Concepts equiv. not transf. to individuals** **Relations not transferred between levels**
Henkin	**Concepts and roles not all represented** Well-foundedness ensured Concepts equiv. transferred to individuals **Relations not transferred between levels**	Concepts and roles all represented Well-foundedness ensured Concepts equiv. transferred to individuals **Relations not transferred between levels**

Table 4.1: Summary of meta-modelling semantincs including $\mathcal{SHIQM}$

Abox as "data" at the upper level, into the Tbox as restrictions on the data accessed at the lower level. Hence, for the scenario of the accounting domain illustrated in Figure 3.11, the description logic $\mathcal{SHIQM}^*$ allows explicitly representing two kinds of relations that hold between definitional an operational levels: (i) equalities between individuals and concepts such as $renterPay =_\mathsf{m} RenterPayEnt$ that can already be represented by $\mathcal{SHIQM}$ and (ii) the correspondence between relations from both levels such as *detailDefD* and *detailD*; *detailDefD* relates entry definitions with detail definitions at debit whereas *detailD* relates concrete accounting entries with their details at debit. The latter is the kind of meta-modelling relations that $\mathcal{SHIQM}^*$ solves, in addition to the meta-modelling provided by $\mathcal{SHIQM}$.

Next subsections describe some changes to move from $\mathcal{SHIQM}$ to $\mathcal{SHIQM}^*$. Syntax, semantics and the tableau algorithm are extended to introduce a new meta-modelling statement that is named $MetaRule$. All definitions presented below as well as the full development of proofs are introduced in [Severi19].

4.2.1 Syntax and semantics of $\mathcal{SHIQM}^*$

An *ontology* $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ *in* $\mathcal{SHIQM}^*$ consists of a Tbox $\mathcal{T}$, an Rbox $\mathcal{R}$, an Abox $\mathcal{A}$, and an Mbox $\mathcal{M}$ where $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ is an ontology in $\mathcal{SHIQ}$ and $\mathcal{M}$ is a set of two different *meta-modelling statements*: (1) *equality statements* of the form $a =_\mathsf{m} A$ that equate an individual a to a concept A, and (2) *role characteristics* of the form $MetaRule(R, S)$ for roles R and S.

The roles in $MetaRule(R, S)$ are assumed to be simple. The set of individuals in equality statements $a =_\mathsf{m} A$ is denoted $\mathsf{dom}(\mathcal{M})$, the set of roles occurring in $\mathcal{T}$, $\mathcal{R}$, $\mathcal{A}$ and $\mathcal{M}$ (that now also contains roles) together with their inverses are denoted $\mathbf{R}_\mathcal{O}$ and as for $\mathcal{SHIQM}$ the set of individuals ocurring in $\mathcal{A}$ and $\mathcal{M}$ are denoted $\mathbf{I}_\mathcal{O}$.

The intuition behind the role characteristic $MetaRule(R, S)$ is that for each statement $a =_\mathsf{m} A$ the Tbox is enriched with $A \sqsubseteq \forall S.(\sqcup X)$ where X is the set of all concepts B with (a, b) in R and $b =_\mathsf{m} B$. For the accounting domain represented in Figure 3.11, Abox axioms such as

$$\begin{array}{cc} detailDefD(renterPay, payCashDet) & detailDefD(renterPay, payBankDet) \\ detailDefC(renterPay, payDebtDet) & detailDefC(renterPay, payDamageDet) \end{array} \tag{4.1}$$

represent a dynamic rule given by an expert at the definitional level. This rule restricts the details at debit and credit of "Renter payment" accounting entries. In the example, individuals of the class $RenterPayEnt$ can only be related to individuals of classes $PayCashDet$ and $PayBankDet$ by the role $detailD$, and to individuals of classes $PayDebtDet$ and $PayDamageDet$ by the role $detailC$. The following Tbox axioms express these restrictions:

$$\begin{aligned} RenterPayEnt &\sqsubseteq \forall detailD.(PayCashDet \sqcup PayBankDet) \\ RenterPayEnt &\sqsubseteq \forall detailC.(PayDebtDet \sqcup PayDamageDet) \end{aligned} \tag{4.2}$$

To avoid declaring the above Tbox axioms for all accounting entry definitions (that are a lot for an accounting system), the following MetaRule axioms are introduced, which infer Tbox axioms 4.2.

$$\mathsf{MetaRule}(detailDefD, detailD) \qquad \mathsf{MetaRule}(detailDefC, detailC) \tag{4.3}$$

The following definition of model of an ontology in $\mathcal{SHIQM}^*$ also includes the satisfiability of role characteristics MetaRule.

Definition 25 (Model of an ontology in $\mathcal{SHIQM}^*$)
An interpretation $\mathcal{I}$ is a model of an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ in $\mathcal{SHIQM}^*$ *(denoted as $\mathcal{I} \models \mathcal{O}$) if the following holds:*

1. *the domain Δ of the interpretation is a* subset *of some S_n for some $n \in \mathbb{N}$.*
2. *$\mathcal{I}$ is a model of the ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ in $\mathcal{SHIQ}$.*
3. *$a^{\mathcal{I}} = A^{\mathcal{I}}$ holds for each equality statement $a =_{\mathsf{m}} A$.*
4. *$A^{\mathcal{I}} \subseteq (\forall S.(\sqcup X))^{\mathcal{I}}$ holds for each role characteristic* MetaRule(R, S) *and each equality statement $a =_{\mathsf{m}} A$, where $X = \{B \mid (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ and $b =_{\mathsf{m}} B \in \mathcal{M}\}$.*

The last part of Definition 25 says that for each $a =_{\mathsf{m}} A$ and MetaRule$(R, S) \in \mathcal{M}$ if $x \in A^{\mathcal{I}}$ and $(x, y) \in S^{\mathcal{I}}$, then $y \in B^{\mathcal{I}}$ for some $b =_{\mathsf{m}} B \in \mathcal{M}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

4.2.2 Checking consistency in $\mathcal{SHIQM}^*$

The tableau algorithm for checking consistency of an ontology in $\mathcal{SHIQM}^*$ extends the tableau algorithm for $\mathcal{SHIQM}$ by adding two rules that handle the $MetaRule$ role characteristic.

The definition of completion forest presented below slightly modifies the definition of completion forest for $\mathcal{SHIQM}$ to deal with role characteristics MetaRule.

Definition 26 (Completion forest for $\mathcal{SHIQM}^*$)
A completion forest $\mathcal{F}$ *for a $\mathcal{SHIQM}^*$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ consists of*

1. *a set of nodes labelled with individual names or variable names,*
2. *directed edges between some pairs of nodes,*
3. *for each node labelled x, a set $\mathcal{F}(x)$ of concept expressions,*

4. *for each pair of nodes x and y, a set $\mathcal{F}(x, y)$ containing role names and inverses of role names in $\mathbf{R}_\mathcal{O}$, or labels $\sim R$ in $\mathbf{R}_\sim$, where the set $\mathbf{R}_\sim$ is used in the algorithm to keep record of pairs of nodes in* $\mathsf{dom}(\mathcal{M})$ *not connected by an arc labelled with some role R in $\mathbf{R}_\mathcal{O}$ which is in some role characteristic $MetaRule(R, S)$, and*

5. *two relations between nodes, denoted by $\approx$ and $\not\approx$, defined as for $\mathcal{SHIQM}$.*

Note that $\mathbf{R}_\sim \cap \mathbf{R}_\mathcal{O} = \emptyset$.

Notions of *root nodes*, *successor*, *predecessor*, *neighbour*, *blocking* and *cycles* are exactly as for $\mathcal{SHIQM}$ (see definitions 11, 12 and 15). Moreover, the initialization builds an initial forest following exactly the same procedure as for $\mathcal{SHIQM}$, i.e. nodes of the initial tableau forest are created from individuals that occur in the Abox as well as in the Mbox (see Definition 13). However, the definition of *contradiction* is modified to handle the new role characteristic $\mathsf{MetaRule}$.

Definition 27 (Contradiction for $\mathcal{SHIQM}^*$)
The completion forest $\mathcal{F}$ has a contradiction *if either*

- *A and $\neg A$ belongs to $\mathcal{F}(x)$ for some atomic concept A and node x or*
- *there are nodes x and y such that $x \not\approx y$ and $x \approx y$.*
- *there is a node x such that $\leqslant nS.C \in \mathcal{F}(x)$, and x has $n + 1$ S-neighbours $y_1, \ldots y_{n+1}$ with $C \in \mathcal{F}(y_i)$, $y_i \not\approx y_j$ for all $i, j \in \{1, \ldots n + 1\}$ with $i \neq j$.*
- *R and $\sim S$ belong to $\mathcal{F}(x, y)$, for nodes x, y, roles R, S in $\mathbf{R}_\mathcal{O}$, $\sim S \in \mathbf{R}_\sim$ and $R \sqsubseteq^* S$.*

The definition of *complete forest* slightly modifies the definition for $\mathcal{SHIQM}$ since two new rules are added.

Definition 28 ($\mathcal{SHIQM}^*$-Complete)
*A forest $\mathcal{F}$ is $\mathcal{SHIQM}^*$-*complete *(or just* complete*) if none of the rules of figures 4.3, 4.4 and 4.7 is applicable.*

The procedure that follows the tableau algorithm for $\mathcal{SHIQM}^*$ is the same as for $\mathcal{SHIQM}$. The only difference is that in the step 2 of application of expansion rules. Besides applying the expansion rules for $\mathcal{SHIQ}$ of Figure 4.3 and the rules for $\mathcal{SHIQM}$ of Figure 4.4 (to deal with equality statements $a =_\mathsf{m} A$) , now the rules $\mathsf{Close\text{-}Meta}$ and $\mathsf{MetaRule}(R, S)$ of Figure 4.7 are also applied to handle the new meta-modelling statement *MetaRule*. Moreover, the $\mathsf{MetaRule}(R, S)$-rule of Figure 4.7 also extends the Tbox (in addition to the $\approx$-rule for $\mathcal{SHIQM}$).

As well as in $\mathcal{SHIQM}$, in the last step the algorithm says that the ontology in $\mathcal{SHIQM}^*$ is consistent iff the expansion rules can be applied in such a way they yield a $\mathcal{SHIQM}^*$-complete $(\mathcal{T}, \mathcal{F})$ without contradictions nor cycles. Otherwise the algorithm says that it is inconsistent. The algorithm stops when reachs *some* $\mathcal{SHIQM}^*$-complete $(\mathcal{T}, \mathcal{F})$ that has neither contradictions nor cycles or when all the choices have yield $(\mathcal{T}, \mathcal{F})$ that has either contradictions or cycles.

Below the intuition behind the new rules of Figure 4.7 is given.

Close-Meta-**rule**:
Let $a =_\mathsf{m} A$, $b =_\mathsf{m} B$ and $\mathsf{MetaRule}(R, S)$ in $\mathcal{M}$, $a \approx x$, $b \approx y$ with x, y representatives of equivalence classes for a, b. If neither $R \in \mathcal{F}(x, y)$ nor $\sim R \in \mathcal{F}(x, y)$, then add either R to $\mathcal{F}(x, y)$ or $\sim R$ to $\mathcal{F}(x, y)$.

$\mathsf{MetaRule}(R, S)$-**rule**:
Let $a =_\mathsf{m} A$ and $\mathsf{MetaRule}(R, S)$ be in $\mathcal{M}$, and suppose that the Close-Meta-rule cannot be applied. If $\neg A \sqcup \forall S.(\sqcup X)$ does not belong to $\mathcal{T}$, then add $\neg A \sqcup \forall S.(\sqcup X)$ to $\mathcal{T}$ for $X = \mathsf{Image}_\mathcal{F}(R, a)$ where $\mathsf{Image}_\mathcal{F}(R, a) = \{B \mid P \in \mathcal{F}(x, y),\ b =_\mathsf{m} B,\ P \sqsubseteq^* R,\ a \approx x,\ b \approx y$ with x, y representatives of equivalence classes for $a, b\}$.

Figure 4.7: $\mathcal{SHIQM}^*$ Expansion Rules for meta-modelling statements

The $\mathsf{MetaRule}(R, S)$-rule changes the Tbox by adding inclusion axioms of the form $A \sqsubseteq \forall S.C$ with the concept A in a equality statement $a =_\mathsf{m} A$. The filler C is a disjunction of concept names with meta-modelling obtained from the individuals related to a via R. For example, suppose $\mathsf{MetaRule}(R, S) \in \mathcal{M}$ and R only belongs to $\mathcal{F}(a, b_1)$ and $\mathcal{F}(a, b_2)$, with $a =_\mathsf{m} A$, $b_1 =_\mathsf{m} B_1$, $b_2 =_\mathsf{m} B_2$ then, the Tbox axiom added by the $\mathsf{MetaRule}(R, S)$-rule is $A \sqsubseteq \forall S.(B_1 \sqcup B_2)$.

The Close-Meta-rule creates arcs non-deterministically with either R or $\sim R$ and it is also needed for completeness. The $\mathsf{MetaRule(R, S)}$-rule is only applied when the Close-Meta-rule cannot be applied any more. This guarantees that $\mathsf{Image}_\mathcal{F}(R, a)$ is always the same, i.e. if $\mathcal{F}'$ is obtained by expanding $\mathcal{F}$ then, $\mathsf{Image}_\mathcal{F}(R, a) = \mathsf{Image}_{\mathcal{F}'}(R, a)$.

Correctness of the tableau algorithm for $\mathcal{SHIQM}^*$

The justification of *termination* is the same as for $\mathcal{SHIQM}$ taking into account that the $\mathsf{MetaRule}(R, S)$-rule also adds Tbox axioms that are bounded since they are finite combinations of roles and concept names of the Mbox.

Regarding *soundness and completeness*, main changes with respect to $\mathcal{SHIQM}$ are in definitions of the *tableau* and the *canonical structure* for $\mathcal{SHIQM}^*$, as well as the corresponding adjustments in proofs.

The definition of *tableau for a $\mathcal{SHIQM}^*$ ontology* extends the tableau for $\mathcal{SHIQM}$ by adding two properties (P20) and (P21), and also considering the set of labels $\mathbf{R}_\sim$ defined above.

Definition 29 (Tableau for $\mathcal{SHIQM}^*$)
Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ be a $\mathcal{SHIQM}^*$ ontology, with $\mathbf{I}_\mathcal{O}$ and $\mathbf{R}_\mathcal{O}$ the set of individuals and roles in $\mathcal{O}$, and $\mathbf{R}_\sim$ the set of labels $\sim R$ such that R is a role in some role characteristic $\mathsf{MetaRule}(R, S)$.
$\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ *is a tableau for $\mathcal{O}$ if*

1. $\mathbf{S} \subseteq S_n$ *for some* S_n,
2. $\mathcal{L}$ *maps each element in* $\mathbf{S}$ *to a set of concepts,*
3. $\mathcal{E} : \mathbf{R}_\mathcal{O} \cup \mathbf{R}_\sim \to 2^{\mathbf{S} \times \mathbf{S}}$,

4. $\mathcal{J} : \mathbf{I} \to \mathbf{S}$ *maps individuals to elements in* $\mathbf{S}$ *and*

5. *for all* $s, t \in \mathbf{S}$, $a, b \in \mathbf{I}_{\mathcal{O}}$, $R, S \in \mathbf{R}_{\mathcal{O}}$, $\sim R \in \mathbf{R}_{\sim}$ *and concepts* C, C_1, C_2, *besides properties (P1)-(P19) of Definition 18, the following properties hold:*

   (P20) If $\mathsf{MetaRule}(R, S)$, $a =_{\mathsf{m}} A$, $b =_{\mathsf{m}} B$ *in* $\mathcal{M}$ *then* $(\mathcal{J}(a), \mathcal{J}(b)) \in \mathcal{E}(R)$ *iff* $(\mathcal{J}(a), \mathcal{J}(b)) \notin \mathcal{E}(\sim R)$.

   (P21) If $\mathsf{MetaRule}(R, S)$, $a =_{\mathsf{m}} A$ *in* $\mathcal{M}$ *then* $\neg A \sqcup \forall S.(\sqcup X) \in \mathcal{L}(s)$ *where* $X = \{B \mid (\mathcal{J}(a), \mathcal{J}(b)) \in \mathcal{E}(P),\ b =_{\mathsf{m}} B \in \mathcal{M},\ P \sqsubseteq^* R\}$.

Lemma 1 for $\mathcal{SHIQM}$ is enunciated below for $\mathcal{SHIQM}^*$.

Lemma 2 *Let* $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ *be a* $\mathcal{SHIQM}^*$ *ontology.* $\mathcal{O}$ *is consistent iff there exists a* $\mathcal{SHIQM}^*$*-tableau for* $\mathcal{O}$.

The $\Leftarrow$ direction of this lemma is proved by showing that the interpretation $\mathcal{I}$ defined as in Lemma 1 is a model of $\mathcal{O}$ according to Definition 25, which adds the item 4 to the definition of model for $\mathcal{SHIQM}$. If $\mathsf{MetaRule}(R, S)$ and $a =_{\mathsf{m}} A$ are in $\mathcal{M}$, from the definition of $\mathcal{I}$, $X = \{B \mid (\mathcal{J}(a), \mathcal{J}(b)) \in \mathcal{E}(P), b =_{\mathsf{m}} B \in \mathcal{M}, P \sqsubseteq^* R\} = \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}, b =_{\mathsf{m}} B\}$ since R is simple, and from (P21) of Definition 29 $\neg A \sqcup \forall S.(\sqcup X) \in \mathcal{L}(s)$ for all $s \in \mathbf{S}$, hence $s \in (\neg A \sqcup \forall S.(\sqcup X))^{\mathcal{I}}$ for all $s \in \Delta^{\mathcal{I}}$ which implies that $A^{\mathcal{I}} \subseteq (\forall S.(\sqcup X)^{\mathcal{I}}$.

To prove the $\Rightarrow$ direction, the tableau $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ defined from a model $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ of $\mathcal{O}$ is modified to consider labels in $\mathbf{R}_{\sim}$ as follows.

$$
\begin{array}{lll}
\mathbf{S} := \Delta & & \\
\mathcal{L}(s) & := & \{C \in \mathsf{clos}(\mathcal{O}) \mid s \in C^{\mathcal{I}}\} \\
\mathcal{E}(R) & := & R^{\mathcal{I}} \text{ for } R \in \mathbf{R}_{\mathcal{O}} \\
\mathcal{E}(\sim R) & := & \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid (a^{\mathcal{I}}, b^{\mathcal{I}}) \notin R^{\mathcal{I}}, a =_{\mathsf{m}} A, b =_{\mathsf{m}} B, \\
 & & \mathsf{MetaRule}(R, S) \text{ in } \mathcal{M},\ R \in \mathbf{R}_{\mathcal{O}},\ \sim R \in \mathbf{R}_{\sim}\} \\
\mathcal{J}(a) & := & a^{\mathcal{I}}
\end{array}
\tag{4.4}
$$

where clos is defined as

$$\mathsf{clos}(\mathcal{O}) = \bigcup_{C(a) \in \mathcal{A} \text{ or } C \in \mathcal{T} \cup \mathsf{concepts}(\mathcal{M})} \mathsf{clos}(C)$$

with

$$
\begin{array}{rl}
\mathsf{concepts}(\mathcal{M}) = & \{A \sqcap \neg B \sqcup B \sqcap \neg A, A \sqcup \neg B, B \sqcup \neg A \mid a =_{\mathsf{m}} A, b =_{\mathsf{m}} B \in \mathcal{M}\} \cup \\
 & \{\neg A \sqcup \forall S.(\sqcup X) \mid a =_{\mathsf{m}} A \in \mathcal{M}, X \subseteq \{B \mid b =_{\mathsf{m}} B \in \mathcal{M}\}\}
\end{array}
$$

Then, (P20) and (P21) must be showed. For example, to prove (P21), if $a =_{\mathsf{m}} A$, $\mathsf{MetaRule}(R, S)$ are in $\mathcal{M}$, since $\mathcal{I}$ is a model of $\mathcal{O}$:
$A^{\mathcal{I}} \subseteq (\forall S.(\sqcup X)^{\mathcal{I}}$ with $X = \{B \mid (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}, b =_{\mathsf{m}} B\}$, and from 4.4
$X = \{B \mid (\mathcal{J}(a), \mathcal{J}(b)) \in \mathcal{E}(P), b =_{\mathsf{m}} B, P \sqsubseteq^* R\}$.
Hence, $\neg A \sqcup \forall S.(\sqcup X) \in \mathcal{L}(s)$ for all $s \in \mathbf{S}$.

The *canonical tableau structure* for $\mathcal{SHIQM}^*$ also makes use of the function set introduced in Definition 20, but now the domain of $\mathcal{E}'$ is extended with the set $\mathbf{R}_{\sim}$.

Definition 30 ($\mathcal{SHIQM}^*$ canonical tableau structure)
Let $\mathcal{F}$ be a completion forest for a $\mathcal{SHIQM}^*$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$. We define the canonical tableau structure $\mathbb{T}' = (\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}')$ *built from $\mathcal{F}$ as follows:*

$$
\begin{array}{ll}
\mathbf{S}' = & \{\mathsf{set}(p) \mid p \in \mathbf{S}\} \\
\mathcal{L}'(s) = & \mathcal{L}(p) \textit{ with } s = \mathsf{set}(p) \\
\mathcal{E}'(R) = & \{(\mathsf{set}(p), \mathsf{set}(q)) \in \mathbf{S}' \times \mathbf{S}' \mid (p,q) \in \mathcal{E}(R)\} \textit{ for } R \in \mathbf{R}_{\mathcal{O}} \\
\mathcal{E}'(\sim R) = & \{(\mathsf{set}(p), \mathsf{set}(q)) \in \mathbf{S}' \times \mathbf{S}' \mid (p,q) \notin \mathcal{E}(R)\} \textit{ for } \sim R \in \mathbf{R}_{\sim}. \\
\mathcal{J}'(a) = & \mathsf{set}(\mathcal{J}(a))
\end{array}
$$

where $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is the canonical tableau structure for $\mathcal{SHIQ}$.

Theorem 3 (Soundness) *Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$. If the expansion rules for $\mathcal{SHIQM}^*$ can be applied to $\mathcal{O}$ in such a way that they yield a complete $(\mathcal{T}, \mathcal{F})$ that has no contradictions and has no cycles w.r.t. $\mathcal{M}$ then $\mathcal{O}$ is consistent.*

As in $\mathcal{SHIQM}$, from Lemma 2 it is enough to prove that the canonical tableau structure given in Definition 30 is a tableau for $\mathcal{O}$, and for this, properties (P20) and (P21) must be showed. (P20) follows from Definition 30 of canonical structure for $\mathcal{SHIQM}^*$, and (P21) is showed using the definition of set and the fact that $\mathcal{F}$ is complete.

The proof of *completeness* is almost the same as for $\mathcal{SHIQM}$. It also makes use of a structure preserving map $\pi : \mathcal{F} \to \mathbb{T}$, taking into account $\mathsf{Close\text{-}Meta}$ and $\mathsf{MetaRule(R, S)}$ rules, and also considering Tbox axioms generated by the $\mathsf{MetaRule(R, S)}$-rule for the Tbox-rule.

Theorem 4 (Completeness) *Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ be a $\mathcal{SHIQM}^*$-ontology. If $\mathcal{O}$ is consistent, then the expansion rules for $\mathcal{SHIQM}^*$ can be applied to $\mathcal{O}$ such that they yield a complete completion forest with no contradictions and no cycles w.r.t. $\mathcal{M}$.*

4.2.3 Representing case studies with $\mathcal{SHIQM}^*$

For the educational and accounting domains presented in Chapter 3 it is not enough to keep equality relations between objects with meta-modelling through different levels. Furthermore, relations between individuals expressed by roles at higher levels must be transferred as relations between corresponding concepts (by meta-modelling) in lower levels.

For the *educational domain* illustrated in Figure 3.8, relations between instances of the role *canUse* must be tranferred to the relation given by the role *uses* between corresponding concepts, and relations between instances of the role *uses* must be translated to the role *attends*. For the latter correspondence, the instance *basicProgramming* of the concept *Module* is connected by *uses* to instances *classroom* and *web* of the concept *WorkEnv*. Hence, instances of the concept *DBProgRegitration* can only be connected to instances of concepts *ClassEnv* and *WebEnv*. This restriction can be expressed by the Tbox axiom $DBProgRegitration \sqsubseteq \forall attends.(ClassEnv \sqcup WebEnv)$, and the same for all definitions that teachers make for all modules in the university. The semantics of the MetaRule role characteristic provided by the description logic $\mathcal{SHIQM}^*$ allows just specifying the roles that must keep this correspondence by introducing MetaRule($uses, attends$). Then, MetaRule($uses, attends$) infers Tbox axioms such as $DBProgRegitration \sqsubseteq \forall attends.(ClassEnv \sqcup WebEnv)$ at the student level, in conformance with Abox axioms at the teacher level.

Likewise, the *accounting domain* conceptualized in Figure 3.11 requires a strict compliance of rules for accounting entries given by experts on accounting. Like for the educational domain, relations between instances of roles *detailDefD*, *detailDefC* must be translated between corresponding concepts for the roles *detailD*, *detailC* respectively, and to ensure this, MetaRule(*detailDefD*, $detailD$) and MetaRule(*detailDefC*, *detailC*) are introduced. However, for the accounting domain it is also important that experts can validate their definitions both for correctness and for completeness. As is expressed by the axiom (4) of Table 3.3 in Chapter 3, the "renter payment" entry definition must have credit details for all accounts that are debits for the renter in some entry definition. This can be expressed in the Tbox at the expert level as follows.

$$Account \sqcap \neg Avalilability \sqcap \exists account^{-}.(\exists detailDefD^{-}.\top) \sqsubseteq$$
$$\exists account.^{-}.(\exists detailDefC^{-}.(\exists detailDefD.\exists account.Avaliability))$$

Given that MetaRule translates rules from expert level to operational level, this kind of restriccions are translated to the operational level. To validate this directly at the operational level, it is needed to explore all Tbox axioms such as:

$$RenterPayEnt \sqsubseteq \forall detailC.(PayDebtDet \sqcup PayRentFee)$$

but it is not possible to declare this restriction in the ontology.

Another benefit of the MetaRule is that it is possible to check for a condition only once over an accounting entry definition instead of checking the same condition over all concrete accounting entries which agree with that definition. The following Tbox axiom is introduced at the level of experts at the moment they set accounting definitions. It checks that accounting entries with availability accounts at debit cannot have other accounts at debit which are not availability accounts.

$$EntryDef \sqcap \exists detailDefD.(\exists account.Availability) \sqsubseteq$$
$$EntryDef \sqcap \forall detailDefD.(\exists account.Availability)$$

Then, the following Tbox axiom does not need to be introduced. It checks for all entries introduced daily at the operational level.

$$Entry \sqcap \exists detailD.(\exists account.Availability) \sqsubseteq Entry \sqcap \forall detailD.(\exists account.Availability)$$

4.3 Conclusions

This chapter presents the theoretical foundation of the description logics $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$ that are the base for implementing a framework with the capability of unifying different abstraction levels of a domain conceptualization. They allow ontology engineers to represent real objects and relations between objects from different perspectives, giving a solution to conceptualize hierarchical user views in organizations.

Description logics $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$ are defined following a Henkin style of semantics which is at the same time a strong and a flexible semantics. The "strong" part is given by the Henkin approach. Besides following a direct semantics coherent with the semantics of OWL, the Henkin style semantics ensures the consistency in the representation of real objects at different perspectives, keeping coherence of the equality between objects and also for relations given by roles. The "flexible" part is given by the fact that all meta-modelling layers are represented within a single domain such that each layer can interact with any other as long as the domain keeps well-founded. Hence, according to the classification of meta-modelling aproaches given in Chapter 2, it is the first one meta-modelling approach classified as *Henkin-global layered*.

To check consistency of an ontology in $\mathcal{SHIQM}^*$ five rules are added to the tableau algorithm for $\mathcal{SHIQ}$. It allows extending the code of existing reasoners in a modular way. A first protopype that extends the reasoner Pellet by adding the three rules of Figure 4.4 for $\mathcal{SHIQM}$, was developed by Ignacio Vidal [Motz16].

Case studies presented in Chapter 3 show the benefits of the description logics $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$ to solve some requirements which are not covered by the other approaches. On the one hand, fixed layered approaches attach the meta-modelling level to the syntax of the language which becomes troublesome for the ontology engineer. Moreover, regarding expressiveness, fixed layered approaches allow a very restricted representation of different perspectives of a set of objects since it is not possible to freely mix different meta-modelling levels. On the other hand, Hilog semantics are too weak for representing the scenarios addressed in the present work. In particular, they do not ensure a coherent conceptualization of the perspectives of the same domain, that have different users. In particular, Hilog semantics with a global domain do not ensure either the well-foundedness of the domain. Note that the benefits of the description logics $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$ are also valid for ontology networks since the tableau algorithm extended with meta-modelling is applied to the union of the sets of axioms of the networked ontologies, i.e. it is ensured that the interpretation domain of the ontology network is well-founded[5]. Table 4.2 shows a summary for the main meta-modelling approaches with the right lower quadrant corresponding to the description logic $\mathcal{SHIQM}^*$.

Intensions-extensions / Domain layers	Fixed	Global
Hilog	**Concepts and roles not all represented** Well-foundedness ensured **Concepts equiv. not transf. to individuals** **Relations not transferred between levels**	Concepts and roles all represented **Well-foundedness not ensured** **Concepts equiv. not transf. to individuals** **Relations not transferred between levels**
Henkin	**Concepts and roles not all represented** Well-foundedness ensured Concepts equiv. transferred to individuals **Relations not transferred between levels**	Concepts and roles all represented Well-foundedness ensured Concepts equiv. transferred to individuals Relations between levels transferred

Table 4.2: Summary of meta-modelling semantincs including $\mathcal{SHIQM}^*$

[5]Most of the case studies of Chapter 3 are modelled by ontology networks.

Finally, a pending issue is the complexity of checking consistency of an ontology in $\mathcal{SHIQM}^*$ that has not been addressed yet. A first result was obtained for the description logic $\mathcal{ALCM}$. Checking consistency of an ontology in $\mathcal{ALC}$ extended with equality statements was proved to be ExpTime-complete, which does not change when moving from $\mathcal{ALC}$ to $\mathcal{ALCM}$ [Martinez16]. Even though for implementations with a great volume of axioms this can result unpractical, there are some alternatives to counteract it as pointed below.

- Existing OWL reasoners based on the Tableau algorithm have implemented different optimizations such as absorption, lazy unfolding and hypertableau, which significantly improve the performance of reasoners [Baader03, chapter 9][Motik09]. The extension of such implementations by adding new meta-modelling rules will take advantage of such optimizations.

- For those domains that can be modelled by adding meta-modelling to lightweight logics such as $\mathcal{EL}$, $\mathcal{RL}$ or $\mathcal{QL}$ (since the expressivity is enough), it is possible to extend algorithms specific for these lightweight logics, more efficient than tableau, which will most likely remain tractable [Krotzsch12].

- The implementation of parallel reasoning is an alternative to improve the performance of reasoners when they are executed for big ontologies [Quan19]. A possible solution is the implementation of a parallelizing framework for reasoners extended with meta-modelling.

Chapter 5

Meta-modelling ontology pattern

This chapter presents a design pattern to conceptualize an ontology[1] by using the meta-modelling approach given by the description logic $\mathcal{SHIQM}^*$, described in Chapter 4. The proposed pattern assists the ontology engineer in the modelling of domains for which there are requirements at different knowledge levels, in particular different user perspectives. The review of literature about metodological frameworks presented in Section 2.4.2 of Chapter 2 shows that this is the first design pattern that provides guidelines to conceptualize a domain by using meta-modelling approaches that extends description logics [Rohrer19].

Even though the design pattern presented here can be applied to model all case studies presented in Chapter 3, there are requirements that mainly belong to educational and accounting domains which are indeed the motivation of the proposed pattern. Both educational and accounting domains have in common the existence of levels of users who have different perspectives of a set of real objects, and moreover that users at higher levels define rules (on the set of objects) for users of lower levels. Considering a more general context of different levels of knowledge or perspectives of a given domain, the proposed design pattern provides a solution to model two kind of requirements that are identified for each knowledge level: some requirements that correspond to rather static rules such as basic principles of the domain and other more dynamic requirements, in particular rules defined at the immediate higher level.

The remainder of the present chapter is organized as follows. Section 5.1 defines static and dynamic rules in the context of the scenarios presented in Chapter 3. Section 5.2 describes the meta-modelling ontology design pattern. Finally, Section 5.3 presents some conclusiones.

5.1 Static and dynamic rules

The educational domain described in Chapter 3 presents a hierarchy of three levels of users: the institution, the teacher and the student. The institution is in charge of defining learning activities such as modules or conferences, and assigning them services such as work environments or equipment. However, depending on different factors (e.g. economics or politics) the service assignment can vary every year. These changes result in *dynamic rules* for teachers who teach modules and make use of the

[1]From now on, every time “ontology” is said, it also applies to “ontology network”

services enabled by the institution, e.g. this year they can use work environments and equipment but not catering. On the other hand, there are *static rules* such as at least two different work environments must be available for all modules. Likewise, at the level of students there are dynamic rules, e.g. students enrolled in a module can only attend to the work environments enabled by the teacher every year, and also static rules such as students must attend at least one of the enabled work environments.

Regarding the accounting domain, definitions made by expert users about different kind of accounting entries become dynamic rules for the level of operators. However, there are also static rules for operators, such as each accounting entry must have at least a debit and a credit detail.
In the context of the present work that considers the scenarios presented in Chapter 3, static and dynamic rules are defined as follows.

Definition 31 (Static and dynamic rules)
Let $E = \{e_1, \ldots e_m\}$ be a set of real objects and $P = \{p_1, \ldots p_n\}$ a set of perspectives of E in a set of ontologies $\mathbb{O} = \{O_1, \ldots O_k\}$ according to Definition 1.

- Static rules *for E and $p_i \in P$ are restrictions on elements $p_i(e)$ in $\mathbb{O}$, $e \in E$, that rarely change over the life cycle of E.*
- Dynamic rules *for E and $p_i \in P$ are restrictions on concepts $p_i(e)$ in $\mathbb{O}$, $e \in E$, that are determined by corresponding individuals $p_j(e)$ in $\mathbb{O}$, in a perspective p_j of E in $\mathbb{O}$, $p_j \neq p_i$.*

Intuitively, dynamic rules on sets of data for a given perspective (e.g. for the user student) are restrictions that are defined and changed at a higher perspective that perceives the same objects with less granularity (e.g. the user teacher perceives objects modules as individuals). These kind of rules probably change over the life cycle of the set of objects, since in general arise from conditions more sensitive to different factors that influence the business, such as the economical situation or new organization leaders.

5.2 A meta-modelling ontology design pattern

This section presents a design pattern that guides in the application of the Henkin-global layered meta-modelling approach given by the description logics $\mathcal{SHIQM}^*$ [Motz15, Severi19]. Rohrer et al. introduce the pattern following the style of Gamma et al. [Rohrer19, Gamma95].

Pattern name. Meta-modelling ontology pattern.

Intent. Taking ontologies as modelling artifacts, the *meta-modelling ontology pattern* is intended to conceptualize a domain by modelling two or more knowledge levels associated to different perspectives[2]. For each knowledge level, there exist static or dynamic rules (or both) which are restrictions on a set of objects in the domain.

[2]Note that for the case studies of Chapter 3 the perspectives correspond to different levels of users. However, the definition presented in Chapter 3 is a more general notion of perspective that is not necessarily associated to a human user.

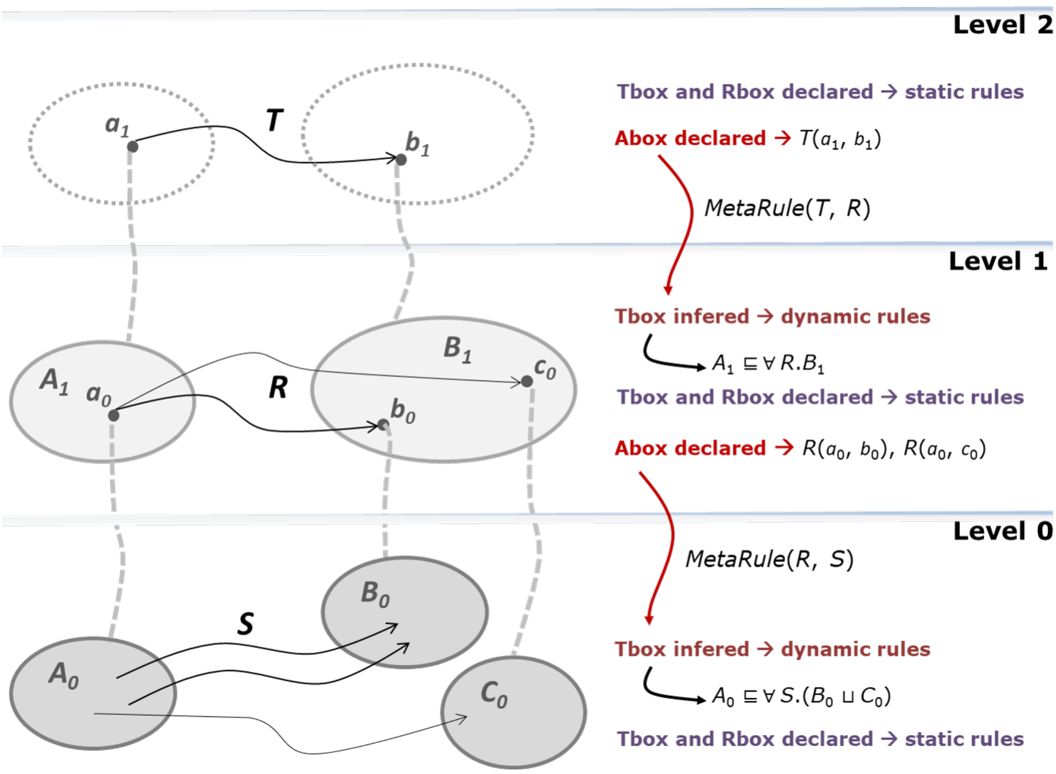

Figure 5.1: Meta-modelling ontology pattern

Motivation. The motivation for applying the *meta-modelling ontology pattern* is given by the scenarios about educational and accounting domains described in Chapter 3.

Applicability. The *meta-modelling ontology pattern* applies instead of single-level modelling approaches to conceptualize two or more knowledge levels associated to different perspectives of a set of objects of a given domain. In most cases, perspectives are associated to different levels of users. This pattern gives solution to the representation of both static and dynamic business rules at each knowledge level, according to Definition 31.

Proposed solution Figure 5.1 illustrates the structure of the proposed pattern. It shows the different knowledge levels which are assigned natural numbers.
Given the knowledge level n, *ontologies that conceptualize this level are separate from ontologies that correspond to knowledge levels different from* n and the *rules in the level* n are modelled as follows.

- *Static rules* are represented in the Tbox and Rbox of the level n.
- *Dynamic rules* that restrict the relation between concepts A and B by a role S are represented as follows.
  - Abox axioms in the level $n+1$, that relate individuals a, b by a role R.
  - Mbox axioms $a =_{\mathsf{m}} A$, $b =_{\mathsf{m}} B$ from the level $n+1$ to the level n.
  - The Mbox axiom $\mathsf{MetaRule}(R, S)$ from the level $n+1$ to the level n.

- Any other kind of dynamic rules, e.g. cardinality restrictions, that are not the focus of the presented pattern, must be represented and changed directly in the Tbox or Rbox of the level n.

The *meta-modelling ontology pattern* proposes that dynamic rules are introduced in the Abox of the level $n+1$ to impact in the Tbox of the level n. Mbox axioms equating individuals at the level $n+1$ to concepts at the level n allow expressing that individuals and corresponding concepts are the same real entities that are visualized with different granularity in different perspectives. Finally, the Mbox axiom $\mathsf{MetaRule}(R, S)$ express that relations on the role R at the level $n+1$ are constraints for the role S at the level n, on concepts semantically equivalent than individuals related by R. Hence, the reasoner extended with rules to deal with Mbox axioms will infer the Tbox axioms that express dynamic rules on the level n, as Figure 5.1 shows.

Consequences. The application of the *meta-modelling ontology pattern* has some advantages that are described below.

- In general, *static rules* are accepted and agreed aspects of the domain. For this reason, it is more likely to be reused, and the fact that they are modelled in the Tbox favors the reuse of a structure, which later is populated with instances.
- As *dynamic rules* for the level n are modelled in the Abox of the level $n+1$, this kind of rules are treated as data in the level $n+1$, what makes the presented pattern a flexible approach.
- When *static rules* (as Tbox or Rbox axioms) on a level n, $n > 0$, restrict individuals related by a role R which is in some axiom $\mathsf{MetaRule}(R, S)$, such static rules on the level n are in fact *rules on rules* on the level $n-1$. It is due to Abox axioms with R are translated as rules (Tbox) for the level $n-1$.
- Expressing *rules for a level* n *on rules for the level* $n-1$ have the advantage of checking for a condition only once over an individual of the level n, avoiding to check the same condition for all instances of the concept equated to that individual by meta-modelling. For example, let's consider for the educational domain a rule on the Teacher level which restricts that a module cannot use both web and lab work environments. Hence, there is a Tbox axiom $Module \sqsubseteq \neg(\exists uses.\{web\} \sqcap \exists uses.\{lab\})$[3] at the Teacher level. This restriction (and others) is checked at the level of each instance of $Module$ which forces that no student who is enrolled on a module can attend both web and lab work environments. However, if we do not apply the pattern and model the domain with a single knowledge level we have to check this condition (and many others) at the level of each student registration.

Limitations. The *meta-modelling ontology pattern* does not solve dynamic rules for a level n different from those defined at the level $n+1$ as relations on objects. These relations are translated to the level n as restrictions on relations between classes (infered Tbox axioms of Figure 5.1).

[3]Note that even though our extension of $\mathcal{SHIQ}$ does not consider nominals, we declare a Tbox axiom including them, for the sole purpose of illustrating the usefullness of the rules on rules.

Example of pattern instantiation. For the accounting domain presented in Chapter 3 the level 0 corresponds to the operator user whereas the level 1 corresponds to the expert user. The requirement that *each concrete accounting entry has at least one debit detail and one credit detail* (see the group of requirements 2 in Section 3.5 of Chapter 3) is a *static rule* at the level 0 since it comes from the ALE-based accounting model of debits and credits (universally adopted) [Meigs83]. By applying the pattern, the rule is expressed by the following Tbox axiom in the level 0 (see axiom (1) in Table 3.3):

$$Entry \sqsubseteq \exists detailD.\top \sqcap \exists detailC.\top$$

Conversely, the requirement that *accounting entries have details for accounts in accordance with the definitions of expert users* (see the group of requirements 1 in Section 3.5 of Chapter 3) is a *dynamic rule* at the level 0 which is introduced at the level 1 by expert users in charge of defining the accounts for debit and credit that correspond to each kind of accounting entry. For example, the expert user defines that each "Renter payment" accounting entry must have debit details for accounts "Cash" or "Bank" and credit details for accounts "Renter Debt" or "Renter Fee" (see Figure 3.11). By applying the pattern, the rule is expressed by the following Abox axiom in the level 1:

$$\begin{array}{l} detailDefD(renterPay, payCashDet) \\ detailDefD(renterPay, payBankDet) \\ detailDefC(renterPay, payDebtDet) \\ detailDefC(renterPay, payFeeDet) \end{array}$$

with the following Mbox axioms that unify both levels and translate the above relations to restrictions on relations between corresponding concepts:

$$\begin{array}{l} renterPay =_{\mathsf{m}} RenterPayEnt \\ payCashDet =_{\mathsf{m}} PayCashDet \\ payBankDet =_{\mathsf{m}} PayBankDet \\ payDebtDet =_{\mathsf{m}} PayDebtDet \\ payFeeDet =_{\mathsf{m}} PayFeeDet \\ \mathsf{MetaRule}(detailD, detailDefD) \\ \mathsf{MetaRule}(detailC, detailDefC) \end{array}$$

Hence, the following Tbox axioms are infered:

$$\begin{array}{l} RenterPayEnt \sqsubseteq \forall detailD.(PayCashDet \sqcup PayBankDet) \\ RenterPayEnt \sqsubseteq \forall detailC.(PayDebtDet \sqcup PayFeeDet) \end{array}$$

Moreover, the requirement that *expert users are able to easily validate the completeness and correctness of their definitions* (see the group of requirements 3 in Section 3.5 of Chapter 3) can be expressed by applying the pattern, by introducing Tbox axioms in the level 1. For example, the requirement that all accounting entries which move availability accounts (Cash, Bank) at debit (credit) cannot have non availability accounts also at debit (credit) is a correctness rule that can be checked at the moment of introducing the entry definition of the renter payment entry by introducing Tbox axiom:

$$\begin{array}{c} EntryDef \sqcap \exists detailDefD.(\exists account.Availability) \sqsubseteq \\ EntryDef \sqcap \forall detailDefD.(\exists account.Availability) \end{array}$$

If the above restriction does not hold then the reasoner gives an inconsistency, and hence the expert definition cannot be introduced. However, for completeness rules such as the Tbox axiom (4) in Table 3.3 of Chapter 3 (recall the example of damage expenses), the reasoner does not give an inconsistency. This happens because OWL, based on description logics, adheres to the open world assumption. To solve this, a non-monotonic extension to description logics could be used. Besides a Tbox, Rbox, Abox and Mbox, a set of assertions with closed predicates can be added to $\mathcal{SHIQM}^*$ to express that the extension of a concept or role is completely determined by this set of assertions, as it would be in a database table [Bossu85, Patel-Schneider12]. This extension is out of the scope of the present work, this will be addressed in a future work.

5.3 Conclusions

This chapter introduces a design pattern to assist the ontology engineer in the application of the ontological meta-modeling approach presented in Chapter 4, the description logic $\mathcal{SHIQM}^*$. This approach allows conceptualizing requirements for different knowledge levels (corresponding to different perspectives), in particular business rules that are static at a given level and some other rules that are dynamically introduced and changed at the immediate higher level.

According to the classification of ontology design patterns (ODPs) of the NeOn methodology (see Section 2.4 of Chapter 2), the *meta-modelling ontology pattern* described above is a *structural* design pattern since it distinguishes different layers or knowledge levels, and it is also a *content* design pattern since it provides a solution to model business rules. As the pattern presented above is a guideline to conceptualize multi-level ontologies, in fact, the *meta-modelling ontology pattern* motivates a new ODP classification in *single-level patterns* and *multi-level patterns*.

Going further, it is possible to make a first analysis of the quality of the presented design pattern. For example, according to the *ODP Quality Model* [Hammar17] the *meta-modelling ontology pattern* satisfies the quality characteristic *functional suitability* and in particular the sub-characteristic *consistency* since it ensures the consistency of the representation of real objects and relations through different perspectives. Moreover, it satifies the characteristic *maintainability*, and in particular the sub-characteristic *modularity* due to it explicitly separates (and at the same time unifies) different abstraction levels. Another characteristic that the meta-modelling ontology pattern satisfies is *compatibility*, for the subcharacteristic *co-existence*, because the other patterns apply within a single layer whereas the pattern presented here solves a kind of interaction between layers.

The *meta-modelling ontology pattern* was applied to different domains (educational and accounting) to give solution to the same problem of modelling static and dynamic rules at different levels. It is visualized as a flexible approach for representing dynamic rules as well as rules on rules. Moreover, for the accounting domain, the pattern was applied to enhance an implemented real application after identifying some weaknesses in the original design. A single-level solution was compared to the two-level model obtained following the pattern, resulting positive the capability of representing more than one abstraction level as well as the inference of (dynamic) constraints on a given level from definitions introduced (as relations on instances) in the upper level. However, the pattern presented in this chapter must be validated by independent ontology engineers to obtain a more reliable evaluation of the capability of the pattern to model different perspectives of a given domain.

Chapter 6

Conclusions, work in progress and future work

This chapter summarizes the overall work done and presents the main conlusions, the work in progress and the future work.

6.1 General conclusions of the thesis work

The motivation of the present thesis work has two main directions. On the one hand, a theoretical (and also technical) direction that led to deepen into the meta-modelling relation, among other different ways of combining ontologies. On the other hand, the work presents a practical direction that aims to solve some requirements of a set of real scenarios, in particular the need of all of them of representing one or more business objects with different granularity. Hence, this section presents some general conclusions as a brief analysis of the main contributions of the thesis work, by grouping them into theoretical and practical contributions, and finally gives a general assessment of the work done.

6.1.1 Theoretical contribution: $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$

From the theoretical point of view, the following paragraphs summarize the main contributions about the syntax, semantics and the reasoning algorithm introduced for representing meta-modelling in $\mathcal{SHIQ}$ ontologies.

Simple and flexible syntax. The description logic $\mathcal{SHIQ}$ is extended with two meta-modelling statements:

- equality statements $a =_{\mathsf{m}} A$ to represent that the individual a is the same real object than the concept A, resulting in the description logic $\mathcal{SHIQM}$, and
- role characteristics $\mathsf{MetaRule}(R, S)$ which translate Abox axioms $R(a, b)$ into the Tbox by entailing axioms $A \sqsubseteq \forall S.(\sqcup X)$ where X is the set of all concepts B such that $b =_{\mathsf{m}} B$, i.e. the sets of individuals related by S must belong to corresponding concepts by meta-modelling. MetaRule extends $\mathcal{SHIQM}$, resulting in the description logic $\mathcal{SHIQM}^*$.

Unlike some of the meta-modelling approaches described in Chapter 2, the syntax of the meta-modelling statements does not force to attach the meta-modelling level to concepts, which results cumbersome for the ontology engineer. It also provides the flexiblity of allowing to declare concepts with instances of different levels, as well as define complex concepts by combining concepts and roles of different levels.

The presented approach results more natural than others at the moment of reusing a set of independent ontologies to build a knowledge base for a given application (as the scenarios described in Chapter 3) since it allows equating an individual in one ontology to a concept of another one, just by introducing an equality statement $a =_{\mathsf{m}} A$. The same happens when it is required to translate relations between instances in one ontology to restrictions on relations between corresponding concepts of another ontology. This is solved by the role characteristic MetaRule (as is applied in educational and accounting scenarios).

Flexible, strong and well-founded semantics. According to the classification of description logics meta-modelling approaches given in Chapter 2, $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$ are Henkin-global layered approaches.

The Henkin semantics ensures the consistency between different meta-modelling levels by transferring equalities of individuals to the equivalence of corresponding concepts, and viceversa. It allows detecting some inconsistencies which results relevant for scenarios that use meta-modelling to represent a set of objects with different granularity.

Instead of defining the interpretation domain separated in layers where Δ is exactly the union of disjoint domains Δ_n, $n \in \mathbb{N}$ (as in fixed layered approaches), the description logic $\mathcal{SHIQM}$ (and then $\mathcal{SHIQM}^*$) defines a "global layered" domain which is a subset of the set S_n defined recursively as $S_{n+1} = S_n \cup \mathcal{P}(S_n)$. This definition allows that elements of different levels of meta-modelling can coexist, providing the required flexibility to properly interpret the syntax of meta-modelling statements. Moreover, the definition of Δ as a subset of S_n ensures that the interpretation domain is a well-founded set. The well-foundedness of Δ is an original contribution from the theoretical point of view. This is also justified in the practical dimension since for the scenarios analyzed in the present work a non well-founded domain of interpretation (e.g. with elements that belong to themselves) results in an inconsistent model.

A reasoning algorithm extended with new rules. The tableau algorithm for $\mathcal{SHIQ}$ is extended with three new rules for $\mathcal{SHIQM}$ to deal with equality statements, and two rules for $\mathcal{SHIQM}^*$ to deal with the role characteristic MetaRule. This extension is aligned to the main structure of the $\mathcal{SHIQ}$ tableau algorithm since existing rules keep unchanged. Moreover, the extended algorithm checks for the existence of cycles in the forest, as a mechanism to ensure that the domain is well-founded. It is an additional condition to return if the ontology is consistent. The idea is to provide a unique algorithm, and then a single piece of software that implements it, which works both for ontologies with meta-modelling and without meta-modelling. For the latter, the behaviour of the extended algorithm is the same as the one of the $\mathcal{SHIQ}$ tableau algorithm.

6.1.2 Practical contribution: representing perspectives

From the practical point of view, the meta-modelling approach was applied to a set of case studies taken from real contexts, and regarding methodological issues, the present work introduces an ontology design pattern for meta-modelling.

A class of case studies and the notion of perspective. The set of case studies analyzed in the present work have in common the requirement that either different levels of users or automatic agents visualize (and possibly change) the more relevant business objects with different granularity. This requirement leads to define the notion of "perspective" as the perception that each user or agent has about a set of objects of a given domain. In this sense, the semantics of $\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$ allows representing several perspectives of the same set of objects, which do not necessarily correspond to levels of meta-modeling. The definition of a global layered domain allows that objects represented at different levels can freely coexist in the same perspective. Moreover, the presented approach ensures the coherence or consistency of different perspectives of a given domain. This is due to it follows a Henkin style of semantics which also unifies all the perspectives by giving a unique interpretation to different representations (as individual or concept) of each business object.

Besides the benefits provided by the description logic $\mathcal{SHIQM}$ to equate individuals to concepts (exploited in all scenarios of the Chapter 3), some domains such as the educational and the accounting domains take advantage of the MetaRule role characteristic provided by $\mathcal{SHIQM}^*$. The MetaRule role characteristic gives a solution to the definition of rules as relations between instances in the perspective of higher user levels, and translates these rules to the perspective of lower user levels as restrictions on their data.

From the analysis of the case studies of the Chapter 3 it was found that it is worth defining "perspective" as a different notion. The definition of perspective cannot be replaced by the one of "ontology" since even though for a given domain each perspective can be represented with a separate ontology, this is not necessarily so. More precisely, a perspective is not the representation itself, it is a mapping between each real object (or entity) and its respresentation (individual or concept) in an ontology, i.e. the granularity with which an actor perceives a set of real objects.

A meta-modelling design pattern. The use of meta-modelling, and in particular the meta-modelling approach proposed in the present work, introduces a certain complexity in the conceptualization of a domain. With the aim of guiding the ontology engineer about when and how to use the proposed approach, the present work introduces a design pattern called *meta-modelling ontology pattern*.

The proposed pattern help the engineer in the modelling of a scenario both from scratch or by combining different ontologies in such a way some objects are conceptualized with different granularity to solve requirements of different knowledge levels, that corresponds to different perspectives. On the one hand, equality statements $a =_{\mathsf{m}} A$ must be introduced to represent that both a and A are the same business object. On the other hand, the ontology engineer is advised about how to represent the business rules that correspond to the requirements of each knowledge level. In this sense, the pattern introduces a criteria to distinguish more static rules from those more dynamic rules that are defined by users (or may be automatic agents) at a higher perspective, e.g. by users who visualize some objects with a higher abstraction level. Then, for each knowledge level, the solution given by the pattern for representing static rules is to introduce them in the Tbox and Rbox. Whereas, the solution for those dynamic rules that restrict a relation (given by a role S) between concepts A and B in a given level, consists in introducing Abox assertions $R(a, b)$ at the immediate higher level. Individuals a and b corresponds by meta-modelling to the concepts A and B, and $\mathsf{MetaRule}(R, S)$ ensures the correspondence between R and S. The main benefits of applying the pattern is that static rules are represented as "structural" assertions in each level whereas dynamic rules are introduced as "data" by the user level in charge of defining and changing the rules. Then, Tbox axioms that restrict the relation between concepts are inferred in the lower level. Hence, $a =_{\mathsf{m}} A$ and $\mathsf{MetaRule}(R, S)$ play the role of unifying different perspectives by ensuring the coherence through different levels.

The meta-modelling ontology pattern was only applied to the case studies of Chapter 3. A validation done by independent working groups is still pending.

6.1.3 General assessment of the thesis work.

In this section a general assessment of the present work is done, considering different dimensions.

Regarding the logical and formal coherence of the description logic extension, all the theoretical work was validated in more than one venue (see the list of publications in Chapter 1).

Regarding the usefuness of the extension to model real domains, this is showed in Chapter 3 by applying the presented approach to a set of different real scenarios.

With respect to the feasibility of the presented solution, even though the study of complexity for $\mathcal{SHIQM}^*$ is still pending, we know that it will be at least as for $\mathcal{SHIQ}$, which is intractable. However, different actions can be taken to improve performance of implementations of reasoners extended with the five new rules presented in Chapter 4. As mentioned at the end of the chapter, some of them are to apply optimization techniques to the implementation of the new rules, to extend reasoners for lightewight logics that are based on algorithms more efficient than tableau, or to implement parallel reasoning.

Finally, from the point of view of the ontology engineer at the moment of modelling a domain, below some pros and cons are analyzed. The main advantages of the approach are as follows.

- Regarding expressivity, like the other meta-modelling approaches the presented extension to $\mathcal{SHIQ}$ enables the capability of representing real objects with different granularity, according to the perspective of the involved actor. Moreover, the present work introduces the capability of defining rules at a given level ensuring they are met in data at lower levels.

- Regarding easy to use, there are three main aspects that combined distinguish the presented approach from others:

  - the flexible syntax, that avoids the typing of concepts with meta-modelling levels,

  - the Henkin semantics, that ensures the coherence between levels, and
  - the automatic detection of cycles, that ensures the well-foundedness of the domain.

- Regarding the modelling of scenarios that integrate different domains, unlike the approaches presented in Chapter 2, the construct $=_{\mathsf{m}}$ allows the mapping of individuals and concepts of different ontologies, facilitating the reuse of existing models. Even though the other meta-modelling approaches are simpler in the sense that the same name can be treated as individual or concept depending on its position in the statement, our approach could also work like this with a slight adjustment in the tableau algorithm, keeping the capability of easily modelling ontology networks with meta-modelling.

- A design pattern is also presented to guide the ontology engineer at the moment of applying the meta-modelling approach, regarding when and how to use it.

Regarding the drawbacks of the presented approach, below the main ones are mentioned.

- Despite the greater expressiveness that meta-modelling approaches provide, from the point of view of the ontology engineer it is also true that some complexity is introduced when modelling a domain. In particular, the approach presented in this work also adds new constructs $=_{\mathsf{m}}$ and MetaRule which even though provide a great expressivity, they also require the engineer to learn how to use them.

- Regarding the MetaRule construct, because of the open world assumption, certain rules that introduce completeness restrictions can be explicited but not effectively checked since the reasoner does not generate an inconsistency. This makes MetaRule less useful in practice; however, this can be solved by extending the present approach with closed predicates, which is a future work.

- Some of the meta-modelling approaches described in Chapter 2 provide useful capabilities such as meta-modelling for roles, and meta-modelling instanciation and subsumption, that also can be incorporated to our approach.

- A validation of the design pattern by independent working groups is still pending.

6.2 Work in progress

The extension of the Hermit reasoner for OWL by adding the five new rules presented in Chapter 4 is being implemented by a student as his final work for graduation [Horrocks12]. The implemented extension will first be tested on some prototypes for educational and accounting case studies.

$\mathcal{SHIQM}$ and $\mathcal{SHIQM}^*$ are Henkin-global layered approaches that ensure the well-foundedness of the domain. They are very flexible regarding modelling choices but at the same time ensure the consistency in the representation of a set of objects with different granularity. However, for more closed scenarios may be it is required to add more restrictions in the interaction between meta-modelling layers. As an example, some works identify meta-modelling antipatterns such as the stratification antipattern presented in the work of Brasileiro et al [Brasileiro16]. Hence, it is being analyzed how to alert the ontology engineer for the presence of this kind of antipatterns, in the form of a warning rather than an inconsistency.

6.3 Future work

As is mentioned above, it is required to go ahead with an independent validation of the meta-modelling ontology pattern. Once the adaptations arising from such validation have been made to the pattern, the idea is to define a metodology for ontology development with meta-modelling, incorporating the meta-modelling ontology pattern. In this sense, it is possible to extend an existing metodology of design of ontologies such as the NeOn methodology [SuarezFigueroa12].

With respect to the theoretical point of view, it is planned to advance in the research directions described below.

- The extension of more powerful description logics like $\mathcal{SROIQ}$ (that underlies OWL) with meta-modelling statements to allow using the meta-modelling approach to integrate OWL ontologies.
- The study of the computational complexity for logics extended with meta-modelling which are more expressive than $\mathcal{ALCM}$ [Martinez16]
- The extension of algorithms (different from tableau) for lightweight logics such as structural subsumption algorithms, saturation calculus for instance retrieval or rule-based classification, which allows implementing more efficient reasoners for ontologies with meta-modelling [Baader03, chapter 2][Krotzsch12].
- The application of optimization techniques to the new expansion rules in the implementation of reasoners (e.g. Hermit) extended with meta-modelling.
- The extension of description logics with meta-modelling for roles, i.e. an individual represent the same real object than a role. Recall the case study of recommender systems in Chapter 3; in the conceptualization of web resources the individual *hasAuthor* is related by meta-modelling to the role *hasAuthor*.
- The enrichment of the MetaRule role characteristic to express other kind of restrictions such as rules that infer stronger restrictions in the Tbox, e.g. $A \sqsubseteq \exists S.(\sqcup X)$.
- The extension of description logics with meta-modelling (either lightweight logics like $\mathcal{ELM}^*$ or more expressive logics like $\mathcal{SHIQM}^*$) with closed predicates to solve requeriments of aplications where data are assumed complete, e.g. applications that access to data from a conceptual representation of the domain through an ontology-based data access mechanism [Calvanese07, Calvanese17].

Bibliography

[AGESIC] Agencia de Gobierno Electrónico y Sociedad de la Información y del Conocimiento. https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/, Last date accessed February 2021.

[Alloca09] C. Allocca, M. D'Aquin, and E. Motta. DOOR - Towards a Formalization of Ontology Relations., booktitle = Proceedings of the International Conference on Knowledge Engineering and Ontology Development, KEOD, publisher = INSTICC Press, year = 2009.

[Almeida18] J. P. A. Almeida, Vi. Albani de Carvalho, F. Brasileiro, C. M. Fonseca, and G. Guizzardi. Multi-level conceptual modeling: Theory and applications. In *Proceedings of the XI Seminar on Ontology Research in Brazil and II Doctoral and Masters Consortium on Ontologies*. CEUR-WS.org, 2018.

[Antoniou09] G. Antoniou and F. van Harmelen. Web ontology language: OWL. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 112–131. Springer Publishing Company, Incorporated.

[Antunes19] C. R. Antunes, A. Rademaker, and M. Abel. A category-theoretic approach for the detection of conservativity violations in ontology alignments. In *Proceedings of the XII Seminar on Ontology Research in Brazil and III Doctoral and Masters Consortium on Ontologies, (ONTOBRAS)*. CEUR-WS.org, 2019.

[Atkinson15a] C. Atkinson, R. Gerbig, and T. Kühne. A unifying approach to connections for multi-level modeling. In *Proceedings of ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MoDELS)*. IEEE Computer Society, 2015.

[Atkinson15b] C. Atkinson and T. Kühne. In defence of deep modelling. *Information & Software Technology*, 64, 2015.

[BFO] BFO. http://basic-formal-ontology.org/, Last date accessed February 2021.

[Baader03] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: The-*

ory, *Implementation, and Applications*. Cambridge University Press, 2003.

[Bao09] J. Bao, G. Voutsadakis, G. Slutzki, and V. G. Honavar. Package-based description logics. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, pages 349–371. Springer, 2009.

[Barwise96] K.J. Barwise and L. Moss. On the mathematics of non-wellfounded phenomena. Technical report.

[Blomqvist09] E. Blomqvist. *Semi-automatic Ontology Construction based on Patterns*. PhD thesis, Linköping University, Sweden, 2009.

[Blomqvist16] E. Blomqvist, K. Hammar, and Valentina Presuttii. Engineering ontologies with patterns – the extreme design methodology. In P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, editors, *Ontology Engineering with Ontology Design Patterns - Foundations and Applications*, pages 23–50. IOS Press.

[Borgida03] A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, 1, 2003.

[Bossu85] G.ve Bossu and P. Siegel. Saturation, nonmonotonic reasoning and the closed-world assumption. *Artif. Intell.*, 25, 1985.

[Brasileiro16] F. Brasileiro, J. P. A. Almeida, V. A. de Carvalho, and G. Guizzardi. Applying a multi-level modeling theory to assess taxonomic hierarchies in wikidata. In *Proceedings of the 25th International Conference on World Wide Web, (WWW)*, 2016.

[Brockmans09] S. Brockmans, P. Haase, L. Serafini, and H. Stuckenschmidt. Formal and conceptual comparison of ontology mapping languages. In *Modular Ontologies*, pages 267–291. Springer, 2009.

[Buschmann96] F. Buschmann, R. Meunier, H. Rohnert, and P. Sommerlad. *Pattern-Oriented Software Architecture: A System of Patterns*. Wiley, 1996.

[Calvanese07] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati. Ontology-based database access. In *Proceedings of the Fifteenth Italian Symposium on Advanced Database Systems, (SEBD)*, 2007.

[Calvanese17] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D.e Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8, 2017.

[Carvalho17] V. A. Carvalho, J. P. A Almeida, C. M Fonseca, and G. Guizzardi. Multi-level ontology-based conceptual modeling. *Data & Knowledge Engineering*, 109, 2017.

[Carvalho18] V. A. Carvalho and J. P. A. Almeida. Toward a well-founded theory for multi-level conceptual modeling. *Software and System Modeling*, 17, 2018.

[Chen09] S. Chen and B. W. Morris. Using design patterns, analysis pattern, and case-based reasoning to improve information modeling and method engineering in systems development. *International Journal of Applied Management and Technology*, 7(1), 2009.

[Cima17a] G. Cima, G. De Giacomo, M. Lenzerini, and A. Poggi. Querying OWL 2 QL ontologies under the SPARQL metamodeling semantics entailment regime. In *Proceedings of the 25th Italian Symposium on Advanced Database Systems*. CEUR-WS.org, 2017.

[Cima17b] G. Cima, G. De Giacomo, M. Lenzerini, and A. Poggi. On the SPARQL metamodeling semantics entailment regime for OWL 2 QL ontologies. In *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS*. ACM, 2017.

[Comesania15] D. Comesa na. Modelo conceptual de información geográfica para la ide – uruguay. Master's Thesis, 2015.

[DOLCE] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. WonderWeb deliverable D18 ontology library (final). Technical report, IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web, 2003.

[DRAON] DRAOn reasoner. `http://iddl.gforge.inria.fr/`, Last date accessed February 2021.

[DeLara14] J. De Lara, Esther G., and J. S. Cuadrado. When and how to use multilevel modelling. *ACM Trans. Softw. Eng. Methodol.*, 24, 2014.

[Diaz11] A. Díaz, R. Motz, and E. Rohrer. Making ontology relationships explicit in a ontology network. In *Proceedings of the 5th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW)*. CEUR-WS.org, 2011.

[Diaz12] A. Díaz, R. Motz, E. Rohrer, and L. Tansini. An ontology network for educational recommender systems. In O. C. Santos and J. G. Boticario, editors, *Educational Recommender Systems and Technologies: Practices and Challenges*, pages 67–93. IGI Global, 2012.

[ECLIPSE] Eclipse Modeling Framework. `https://www.eclipse.org/modeling/emf/`. Last date accessed February 2021.

[El-Sappagh14] S. El-Sappagh, M. E. Mogy, and A. M. Riad. A standard fragment of EHR relational data model for diabetes mellitus diagnosis. In *Proceedings of Informatics and Systems*. IEEE, 2014.

[El-Sappagh18] S. El-Sappagh, F. Franda, F. Ali, and K. Kwak. SNOMED CT standard ontology based on the ontology for general medical science. *BMC Med. Inf. & Decision Making*, 18(1), 2018.

[Elmasri15] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Pearson, 7th edition, 2015.

[Falbo13] R. Falbo, G. Guizzardi, A. Gangemi, and V. Presutti. Ontology patterns: Clarifying concepts and terminology. In *Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns co-located with 12th International Semantic Web Conference (ISWC)*, 2013.

[Fischer-Pauzenberger17] C. Fischer-Pauzenberger and W. S. A. Schwaiger. The OntoREA Accounting Model: Ontology-based modeling of the accounting domain. *CSIMQ*, 11, 2017.

[Fowler97] Martin F. *Analysis patterns - reusable object models*. Addison-Wesley-Longman, 1997.

[Gamma95] .E Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., 1995.

[Gangemi06] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann. Modelling ontology evaluation and validation. In *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC*, 2006.

[Gangemi09] A. Gangemi and V. Presutti. Ontology design patterns. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 221–243. Springer Publishing Company, Incorporated, 2009.

[Genero02] M. Genero, G. Poels, and M. Piattini. Defining and validating measures for conceptual data model quality. In *Advanced Information Systems Engineering, 14th International Conference, (CAiSE)*, 2002.

[Ghidini17] C. Ghidini and L. Serafini. Distributed first order logic. *Artif. Intell.*, 253, 2017.

[Giacomo11] G. De Giacomo, M. Lenzerini, and R. Rosati. Higher-order description logics for domain metamodeling. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence, AAAI*. AAAI Press, 2011.

[Glimm10] B. Glimm, S. Rudolph, and J. Völker. Integrated metamodeling and diagnosis in OWL 2. In *International Semantic Web Conference*, 2010.

[Gomez-Perez10] A. Gómez-Pérez, M. Fernandez-Lopez, and O. Corcho. *Ontological Engineering: with examples from the areas of Knowledge*

Management, e-Commerce and the Semantic Web. First Edition. Springer London, 2010.

[Grau06] B. C. Grau, B. Parsia, and E. Sirin. Combining OWL ontologies using E-connections. *J. Web Semant.*, 4, 2006.

[Grau07] B. C. Grau and O. Kutz. Modular ontology languages revisited. In *Proceedings of the IJCAI'07 Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa*, 2007.

[Grau08a] B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *J. Web Sem.*, 6(4), 2008.

[Grau08b] B. C. Grau, I. Horrocks, Y. Kazakov, and Ul. Sattler. Modular reuse of ontologies: Theory and practice. *J. of Artificial Intelligence Research (JAIR)*, 31, 2008.

[Grau09a] B. C. Grau, B. Parsia, and E. Sirin. Ontology integration using E-connections. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, pages 293–320. Springer.

[Grau09b] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Extracting modules from ontologies: A logic-based approach. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, pages 159–186. Springer, 2009.

[Groner12] Jeff Z. Pan, Steffen Staab, Uwe Amann, Jrgen Ebert, and Yuting Zhao. *Ontology-Driven Software Development*. Springer Publishing Company, Incorporated, 2012.

[Gu16] Zhenzhen Gu. Meta-modeling extension of horn-sroiq and query answering. In *Proceedings of the 29th International Workshop on Description Logics, (DL)*. CEUR-WS.org, 2016.

[Gu18] Z. Gu and S. Zhang. The more irresistible Hi(SRIQ) for meta-modeling and meta-query answering. *Frontiers Comput. Sci.*, 12(5), 2018.

[Guarino09] N. Guarino. An overview of ontoclean. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 201–220. Springer Publishing Company, Incorporated.

[Guizzardi12] G. Guizzardi. Ontological foundations for conceptual modeling with applications. In *Proceeding of the 24th International Conference on Advanced Information Systems Engineering, (CAiSE)*. Springer, 2012.

[HL7SNOMED] T. Benson and G. Grieve. *Principles of Health Interoperability. SNOMED CT, HL7 and FHIR*. Springer-Verlag, 2016.

[Hammar17] K. Hammar. *Content Ontology Design Patterns: Qualities, Methods, and Tools.* PhD thesis, Linköping University, Sweden, 2017.

[Hitzler09] P. Hitzler, M. Krötzsch, and S. Rudolph. *Foundations of Semantic Web Technologies.* Chapman & Hall/CRC, 2009.

[Homola07] M. Homola. Distributed description logics revisited. In *Proceedings of the 20th International Workshop on Description Logics (DL).* CEUR-WS.org, 2007.

[Homola10a] M. Homola and L. Serafini. Towards formal comparison of ontology linking, mapping and importing. In *Proceedings of the 23rd International Workshop on Description Logics (DL).* CEUR-WS.org, 2010.

[Homola10b] M. Homola and L. Serafini. Augmenting subsumption propagation in distributed description logics. *Applied Artificial Intelligence*, 24, 2010.

[Homola13] M. Homola, J. Kluka, V. Svátek, and M. Vacura. Towards typed higher-order description logics. In *Proceedings of the 26th International Workshop on Description Logics, (DL).* CEUR-WS.org, 2013.

[Homola14] M. Homola, J. Kluka, V. Svátek, and M. Vacura. Typed higher-order variant of SROIQ - why not? In *Proceedings of the 27th International Workshop on Description Logics, (DL).* CEUR-WS.org, 2014.

[Horrocks00] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic SHIQ. In *Automated Deduction - CADE-17, 17th International Conference on Automated Deduction.* Springer, 2000.

[Horrocks12] I. Horrocks, B. Motik, and Z. Wang. The hermit OWL reasoner. In *1st International Workshop on OWL Reasoner Evaluation (ORE-2012).* CEUR-WS.org, 2012.

[Horrocks99] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *J. Log. Comput.*, 9, 1999.

[ICD] ICD. `http://www.who.int/classifications/icd/en/`, Last date accessed February 2021.

[IDEUY] Infraestructura de datos Espaciales de Uruguay. `https://www.gub.uy/infraestructura-datos-espaciales/`, Last date accessed February 2021.

[IGM] Instituto Geográfico Militar. `http://www.igm.gub.uy/`, Last date accessed February 2021.

[ISO] Systems and software engineering - systems and software quality requirements and evaluation (SQuaRE) - system and software quality models. International Standard ISO/IEC 25010:2011, International Organization for Standardization, 2011.

[Jekjantuk09] N. Jekjantuk, G. Gröner, and J. Z. Pan. Reasoning in metamodeling enabled ontologies. In *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED)*. CEUR-WS.org, 2009.

[Jekjantuk10] N. Jekjantuk, G. Gröner, and J.Z. Pan. Modelling and reasoning in metamodelling enabled ontologies. *Int. J. Software and Informatics.*, 4 (3), 2010.

[Kassel05] Gilles Kassel. Integration of the DOLCE top-level ontology into the ontospec methodology. *CoRR*, abs/cs/0510050, 2005.

[Keet08] C. M. Keet and A. Artale. Representing and reasoning over a taxonomy of part-whole relations. *Appl. Ontol.*, 3(1-2), 2008.

[Keet12] C. M. Keet. Transforming semi-structured life science diagrams into meaningful domain ontologies with didon. *J. of Biomedical Informatics*, 45(3), 2012.

[Keet13] C. M. Keet, M. C. Suárez-Figueroa, and M. Poveda-Villalón. Pitfalls in ontologies and tips to prevent them. In *KMIS*.

[Keet18] C. M. Keet. *An Introduction to Ontology Engineering*. Maria Keet, 2018.

[Konev09] B. Konev, C. Lutz, D. Walther, and F. Wolter. Formal properties of modularisation. In *Modular Ontologies*, pages 25–66. Springer-Verlag, 2009.

[Konev13] B. Konev, C. Lutz, D. Walther, and F. Wolter. Model-theoretic inseparability and modularity of description logic ontologies. *Artif. Intell.*, 203, 2013.

[Koopman12] B. Koopman, G. Zuccon, A. Nguyen, D. Vickers, L. Butt, and P. D. Bruza. Exploiting snomed ct concepts and relationships for clinical information retrieval : Australian e-health research centre and queensland university of technology at the trec 2012 medical track. In *21st Text REtrieval Conference (TREC 2012)*. National Institute of Standards and Technology (NIST), 2012.

[Krotzsch12] M. Krötzsch. Owl 2 profiles: An introduction to lightweight ontology languages. In *Reasoning Web Summer School 2012*. Springer, 2012.

[Kubincova15] P. Kubincová, J. Kluka, and M. Homola. Towards expressive metamodelling with instantiation. In *Proceedings of the 28th International Workshop on Description Logics, (DL)*. CEUR-WS.org, 2015.

[Kubincova16a] P. Kubincová. Higher-order description logics for metamodelling. Master's Thesis, 2016.

[Kubincova16b] P. Kubincová, J. Kluka, and M. Homola. Expressive description logic with instantiation metamodelling. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR*, 2016.

[Kuhne18] T. Kühne. A story of levels. In *Proceedings of MODELS 2018 Workshops, (PAINS)*. CEUR-WS.org, 2018.

[LOCMOD] Locality module extractor. `https://github.com/ernestojimenezruiz/locality-module-extractor`, Last date accessed February 2021.

[Lenzerini14] M. Lenzerini, L. Lepore, and A. Poggi. Making metaquerying practical for hi(dl - lite R) knowledge bases. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences - Confederated International Conferences: CoopIS, and ODBASE*. Springer, 2014.

[Lenzerini16a] M. Lenzerini, L. Lepore, and A. Poggi. Answering metaqueries over hi (OWL 2 QL) ontologies. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, (IJCAI)*. IJCAI/AAAI Press, 2016.

[Lenzerini16b] M. Lenzerini, L. Lepore, and A. Poggi. A higher-order semantics for metaquerying in OWL 2 QL. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR*. AAAI Press, 2016.

[Lenzerini18] M. Lenzerini, L. Lepore, and A. Poggi. Metaquerying made practical for OWL 2 QL ontologies. *Information Systems*, 2010.

[MOF] Meta-object Facility. `https://www.omg.org/mof/`. Last date accessed February 2021.

[Martinez16] M. Martinez, E. Rohrer, and P. Severi. Complexity of the description logic ALCM. In *Fifteenth International Conference on Principles of Knowledge Representation and Reasoning, KR*. AAAI Press, 2016.

[Meigs83] W. B. Meigs and R. F. Meigs. *Financial accounting*. McGraw-Hill, 4th ed. edition, 1983.

[Motik07] B. Motik. On the properties of metamodeling in OWL. *J. Logic and Computation*, 17, 2007.

[Motik09] B. Motik, R. Shearer, and I. Horrocks. Hypertableau reasoning for description logics. *J. Artif. Int. Res.*, 36(1), 2009.

[Motz14] R. Motz, E. Rohrer, and P. Severi. Reasoning for $\mathcal{ALCQ}$ extended with a flexible meta-modelling hierarchy. In *Proceedings of Semantic Technology - 4th Joint International Conference (JIST)*. Springer, 2014.

[Motz15] R. Motz, E. Rohrer, and P. Severi. The description logic SHIQ with a flexible meta-modelling hierarchy. *J. Web Semant.*, 35, 2015.

[Motz16] R. Motz, E. Rohrer, P. Severi, and I. Vidal. OWL extended with meta-modelling. In *Fourth International Semantic Web and Linked Open Data workshop co-located with 15th Ibero-American Conference on AI (IBERAMIA)*. CEUR-WS.org, 2016.

[Motz17] R. Motz, E. Rohrer, and P. Severi. Applying description logics extended with meta-modelling to SNOMED-CT. In *Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, (AMW)*. CEUR-WS.org, 2017.

[NCI] National cancer institute. `https://www.cancer.gov/`, Last date accessed February 2021.

[Noy01] N. F. Noy and De. L. Mcguinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford University, 2001.

[OGMS] Ontology for general medical science. `https://bioportal.bioontology.org/ontologies/OGMS/`, Last date accessed February 2021.

[OntoUML] OntoUML Language. `https://ontouml.org/`, Last date accessed February 2021.

[Pan05] J. Z. Pan, I. Horrocks, and G. Schreiber. OWL FA: A metamodeling extension of OWL DL. In *Proceedings of the OWLED*05 Workshop on OWL: Experiences and Directions, (OWLED)*. CEUR-WS.org, 2005.

[Parent09] C. Parent and S. Spaccapietra. An overview of modularity. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, pages 5–23. Springer, 2009.

[Patel-Schneider12] P. Patel-Schneider and E. Franconi. Ontology constraints in incomplete and complete data. In *The 11th International Semantic Web Conference, ISWC*. Springer, 2012.

[Peroni16] Silvio Peroni. A simplified agile methodology for ontology development. In *OWL: - Experiences and Directions - Reasoner Evaluation - 13th International Workshop, OWLED*, 2016.

[PrinceSales15] T. Prince Sales and G. Guizzardi. Ontological anti-patterns: empirically uncovered error-prone structures in ontology-driven conceptual models. *Data Knowl. Eng.*, 99, 2015.

[PrinceSales16] T. Prince Sales and G. Guizzardi. Anti-patterns in ontology-driven conceptual modeling: The case of role modeling in ontouml. In P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, editors, *Ontology Engineering with Ontology Design Patterns - Foundations and Applications*, pages 161–187. IOS Press.

[Quan19] Z. Quan and V. Haarslev. A parallel computing architecture for high-performance OWL reasoning. *Parallel Comput.*, 83, 2019.

[Rector09] A. L. Rector, R. Qamar, and T. Marley. Binding ontologies and coding systems to electronic health records and messages. *Applied Ontology*, 4, 2009.

[Rohrer10a] E. Rohrer, R. Motz, and A. Díaz. Ontology-based process for recommending health websites. In *Software Services for e-World - 10th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, (I3E)*. Springer, 2010.

[Rohrer10b] E. Rohrer, R. Motz, and A. Díaz. Modeling a web site quality-based recommendation system. In *iiWAS'2010 - The 12th International Conference on Information Integration and Web-based Applications and Services*. ACM, 2010.

[Rohrer10c] E. Rohrer, R. Motz, and A. Díaz. Modeling and use of an ontology network for website recommendation systems. In *On the Move to Meaningful Internet Systems: (OTM)*. Springer, 2010.

[Rohrer11] E. Rohrer, R. Motz, and A. Díaz. Modelling a web site quality-based recommendation system. *Int. J. Web Inf. Syst.*, 7(4), 2011.

[Rohrer14] E. Rohrer, P. Severi, R. Motz, and A. Díaz. Metamodelling in a ontology network. In *Proceedings of the 8th International Workshop on Modular Ontologies (WOMO)*. CEUR-WS.org, 2014.

[Rohrer18] E. Rohrer, P. Severi, and R. Motz. Applying meta-modelling to an accounting application. In *Proceedings of the XI Seminar on Ontology Research in Brazil and II Doctoral and Masters Consortium on Ontologies, (ONTOBRAS)*. CEUR-WS.org, 2018.

[Rohrer19] E. Rohrer, P. Severi, and R. Motz. Meta-modelling ontology design pattern. In *Knowledge Graphs and Semantic Web - First Iberoamerican Conference, (KGSWC)*. Springer, 2019.

[RxNorm] Rxnorm. `https://www.nlm.nih.gov/research/umls/rxnorm/index.html`, Last date accessed February 2021.

[SALUS] Anais dos workshops salus/cyted-cnpq, prosul-cnpq avalsaúde e sticamsud-capes alap. `https://seer.ufrgs.br/cadernosdeinformatica/issue/view/1113`, Last date accessed February 2021.

[SNOMED] Snomed ct. `http://www.snomed.org/`, Last date accessed February 2021.

[SNOMEDGuide] IHTSDO. SNOMED International. Leading healthcare terminology, worldwide. SNOMED CT Starter Guide, 2017.

[SWOOP] SWOOP ontology editor and reasoner. `http://code.google.com/p/swoop/`, Last date accessed February 2021.

[Sandkuhl08] K. Sandkuhl, H. Tellioglu, and S. G. Johnsen. Orchestrating economic, socio-technical and technical validation using visual modelling. In *16th European Conference on Information Systems, ECIS*, 2008.

[Sattler09] U. Sattler, T. Schneider, and M. Zakharyaschev. Which kind of module should I extract? In *Proceedings of the 22nd International Workshop on Description Logics (DL)*, 2009.

[Schmidt06] D. C. Schmidt. Model-driven engineering. *IEEE Computer*, 39(2), 2006.

[Schulz08] S. Schulz, K. Markó, and B. Suntisrivaraporn. Formal representation of complex SNOMED CT expressions. *BMC Med. Inf. & Decision Making*, 8(S-1), 2008.

[Schulz09] S. Schulz, B. Suntisrivaraporn, F. Baader, and M. Boeker. SNOMED reaching its adolescence: Ontologists' and logicians' health check. *I. J. Medical Informatics*, 78, 2009.

[Schulz11] S. Schulz, Rl. Cornet, and K. A. Spackman. Consolidating snomed ct's ontological commitment. *Applied Ontology*, 6(1), 2011.

[Serafini05a] L. Serafini, A. Borgida, and A. Tamilin. Aspects of distributed and modular ontology reasoning. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI*. Professional Book Center, 2005.

[Serafini05b] L. Serafini and A. Tamilin. DRAGO: distributed reasoning architecture for the semantic web. In *The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC*. Springer, 2005.

[Serafini09] L. Serafini and M. Homola. Modular knowledge representation and reasoning in the semantic web. In *Semantic Web Information Management*, pages 147–181. Springer, 2009.

[Severi19] P. Severi, E. Rohrer, and R. Motz. A description logic for unifying different points of view. In *Knowledge Graphs and Semantic Web - First Iberoamerican Conference, (KGSWC)*. Springer, 2019.

[Sirin07] Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2).

[Staab01] S. Staab, R. Studer, H. Schnurr, and Y. Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1), 2001.

[Staab09] S. Staab and R. Studer. *Handbook on Ontologies*. Springer Publishing Company, Incorporated, 2nd edition, 2009.

[Studer98] R. Studer, V.R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1), 1998.

[Suarez-Figueroa10] M. Suarez-Figueroa, A. Gomez-Perez, M. Poveda, J. Ramos, J. Euzenat, and C. Le Duc. Neon deliverable d5.4.3. Revision and extension of the NeOn methodology for building contextualized ontology networks. Technical report, NeOn Project, 2010.

[SuarezFigueroa12] M. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, editors. *Ontology Engineering in a Networked World*. Springer, 2012.

[Suntisrivaraporn07] B. Suntisrivaraporn, F. Baader, S. Schulz, and K. A. Spackman. Replacing SEP-triplets in SNOMED CT using tractable description logic operators. In *Proceedings of AIME 2007*. Springer, 2007.

[Tello04] A. L. Tello and A. Gómez-Pérez. ONTOMETRIC: A method to choose the appropriate ontology. *J. Database Manag.*, 15(2), 2004.

[Thorn10] C. Thörn. *On the Quality of Feature Models*. PhD thesis, Linköping University, Sweden, 2010.

[Tran08] T. Tran, P. Haase, B. Motik, B. C. Grau, and I. Horrocks. Metalevel information in ontology-based applications. In *Proceedings of the Twenty-Third Conference on Artificial Intelligence, AAAI*. AAAI Press, 2008.

[UFO] Unified Fundational Ontology. `https://ontouml.readthedocs.io/en/latest/intro/ufo.html`, Last date accessed February 2021.

[UML] Unified Modeling Language Specification Version 241. `https://www.omg.org/spec/UML/2.4.1/About-UML/`. Last date accessed February 2021.

[UMLS] Unified medical language system. `https://uts.nlm.nih.gov/home.html`, Last date accessed February 2021.

[Uschold96] M. Uschold and M. Gruninger. Ontologies: principles, methods and applications. *Knowledge Eng. Review*, 11, 1996.

[Vrandecic05] D. Vrandecic, H. S. Pinto, C. Tempich, and Y. Sure. The DILIGENT knowledge processes. *J. Knowledge Management*, 9(5), 2005.

[Wang07] Y. Wang, J. Bao, P. Haase, and G. Qi. Evaluating formalisms for modular ontologies in distributed information systems. In *Proceedings of the Web Reasoning and Rule Systems, First International Conference, RR*. Springer, 2007.

[Zimmermann07] A. Zimmermann. Integrated distributed description logics. In *Proceedings of the 20th International Workshop on Description Logics (DL)*. CEUR-WS.org, 2007.

[ZimmermannD08] A. Zimmermann and C. L. Duc. Reasoning with a network of aligned ontologies. In *Proceedings of the Web Reasoning and Rule Systems, Second International Conference, (RR)*. Springer, 2008.