Supporting Metamodeling in Ontologies Using Rules

Haya Majid Qureshi[0000−0002−6622−4950] and Wolfgang Faber[0000−0002−0330−5868]

University of Klagenfurt, Austria
firstname.lastname@aau.at

Abstract. Metamodeling is a feature in ontologies, which allows for having classes as members of other classes. While providing very flexible modeling power, it also comes with decidability issues. Because of this, metamodeling are often outright forbidden or unsupported in modeling frameworks. Some work exists that provides explicit support for metamodeling by augmenting the ontology. We introduce a new approach that provides metamodeling support in ontologies by means of a transformation. It rewrites the metamodeling features automatically into rules (M-rules) and strips them from the ontology. The resulting reduced ontology $\mathcal{O}'$ together with the M-rules then form a hybrid knowledge base, which can be processed using NoHR. This framework allows for effective reasoning and querying over the combined knowledge of ontology and rules.

Keywords: Ontologies · Metamodeling · Rules.

1 Introduction

Metamodeling is a process for specifying conceptual modeling requirements, where classes have other classes as an instance, called metaconcepts. Relations between metaconcepts are called metaproperties. The benefit for adding metamodeling in ontologies provide explicit availability of knowledge. This can allow for more understandable (less subject to misinterpretation) and reusable models. Consider an example adopted from [3], concerning the modeling of biological species, where general of below statments imply subclass relations, stating that all GoldenEagles are Eagles, all Eagles are Birds and Harry is an instance of GoldenEagle, which further can be infered as an instance of Eagle and Birds.

SubClassOf(:Eagle :Birds)
SubClassOf(:GoldenEagle :Eagle)
ClassAssertion(:GoldenEagle :Harry)

However, in species domain contrary to specific properties of some species, one can also articulate expressions about species themselves i.e. "GoldenEagle is listed in the IUCN Red List of endangered species" states that GoldenEagle as a whole class is an endangered species, and not just Harry. To formally model this expression, we can set GoldenEagle as an instance of new class EndangeredSpecies. Making GoldenEagle a type of EndangeredSpecies is concise and clear. Note that making GoldenEagle a subclass of EndangeredSpecies would not be correct, as it results in incorrect conclusions like "Harry is an EndangeredSpecies". In the aassertion above, EndangeredSpecies is a metaconcept for GoldenEagle.

OWL Full can express this type of metamodeling, but it is so expressive that it leads to undecidability [6]. OWL 2 DL provides no explicit semantics support for the example above. This means that the adoption of metamodeling features in OWL require some other, external mechanism. One promising option is the hybrid mechanism of enriching ontologies via rules [5]. Ontology languages and Logic Programming with their distinct features and benefits have been widely studied for knowledge representation on the Semantic Web. Here we focus on NoHR [2], which provides a general framework for combining non-monotonic rules with OWL 2 QL. The choice for using non-monotonic rules here are based on their proximity to common sense reasoning, which seems particularly suitable for metamodeling, where different roles of a single entity may result in different conclusions. The use of OWL 2 QL is also motivated by the Higher Order Semantics [4], in which the same objects can have multiple interpretations, depending on their use. Our approach is to automatically remove and transform the metamodeling portions of an ontology into metamodeling-style non-monotonic rules (in short M-rules). The resulting hybrid knowledge base can then be queried using NoHR. This approach allows for avoiding the semantic discrepancy and inconsistency problems in ontology and yet support metamodeling in ontologies by means of rules.

2 Preliminaries

This section provides a brief overview of the OWL 2 QL ontology language under Metamodeling Semantics [4] and the Hybrid MKNF formalism [2].

OWL 2 QL Let $V_{\mathcal{O}}$ be a vocabulary for an ontology, consisting of a tuple $V_{\mathcal{O}} = (V_N, V_C, V_{OP}, V_{DP}, V_{DT}, \ell)$, where V_N is the set of IRIs (International Resource Identifiers), and V_C, V_{OP}, V_{DP} and V_{DT} are subsets of V_N used to denote the entity names of classes, object properties, data properties, datatypes. Any entity in V_N may simultaneously have more then one role, for instance it can either be a class, or an individual, or data property, or an object property or a data type. Axioms are formed using $Exp^{\mathcal{O}}$ over $V_{\mathcal{O}}$, the set of finite expressions denoting the entities of the ontology.

The Metamodeling Semantics for OWL 2 QL is based on an interpretation structure, constituted by a tuple $\Sigma = (\Delta, \cdot^{I}, \cdot^{E}, \cdot^{R}, \cdot^{A}, \cdot^{T})$, where Δ is the disjoint union of the two non-empty sets Δ_o (the *o*bject domain) and Δ_v (the *v*alue domain), and interpretation functions are partial, which means a single object can simultaneously be an individual $\cdot^I$, a class $\cdot^E$, an object property $\cdot^R$, a data property $\cdot^A$, and a data type $\cdot^T$. An interpretation $\mathcal{I}$ is a pair $(\Sigma, \mathcal{I})$ for ontology $\mathcal{O}$, where Σ is the interpretation structure and $\mathcal{I}$ is the interpretation function. The semantics of logically implied axioms are defined in accordance with the notion of axiom satisfaction with the interpretation $\mathcal{I}$. Moreover, $\mathcal{I}$ is said to be a model of $\mathcal{O}$ if there exists at least an interpretation that is a model of $\mathcal{O}$. Finally, an axiom α, is logically implied by $\mathcal{O}$, denoted as $\mathcal{O} \models \alpha$, if it holds for every model of $\mathcal{O}$.

NoHR Hybrid Knowledgebase We adopt hybrid knowledge bases consisting of well-founded MKNF [2] within NoHR, as combined knowledge representation for first-order translatable OWL 2 QL ontologies and non-monotonic rules. A hybrid MKNF knowledge base $\mathcal{K}$ is a pair $(\mathcal{O}, \mathcal{P})$, where

- $\mathcal{O}$ is any decidable description logic (DL) language, in our case OWL 2 QL, and
- $\mathcal{P}$ is finite set of MKNF rules of the form:

$$KH \leftarrow KB_1^+, \ldots, KB_n^+, notB_1^-, \ldots, notB_m^-$$

where H is the rule head and B_i with arity $1 \leq i \leq m$ are first order atoms in the rule body. A rule with an empty body is a Fact. A rule is a DL-safe rule, if all variables in the rule occur in at least one non-DL atom. If all rules are DL-safe, then $\mathcal{K}$ is DL-safe. The semantic transformation of $\mathcal{K}$ is based on the MKNF formula $\Pi(K)$, to which the semantics of MKNF can be applied. As defined in [1], well-founded MKNF can be queried based on $\mathcal{SLG}(\mathcal{O})$, that guarantees posing ground queries to the DL part of K. These queries return an atom set, which, together with already proved information and $\mathcal{O}$, is necessary to derive an atom of the original query. For a more technical discussion we refer to [2].

3 M-Ontology to M-Rules Framework

It is not easy to do metamodeling in ontologies, as there is little language support for ontological metamodeling. However, OWL 2 provides support for a basic level of metamodeling in ontologies via the use of *Punning*, but standard reasoners give no semantic support for those punned entities and treat them as different type of entities. Moreover, metamodeling features in ontology can easily pose semantic discrepancy in $\mathcal{O}$, that results in undecidability in query answering.

To tackle these issues, our approach is to give semi-metamodeling support to ontologies by expressing the metamodeling information in non-monotonic M-rules (short for metamodeling-rules). For this we remove the metamodeling portion from the original ontology and transform it into the $sine{-}\mathcal{M}-\mathcal{O}$ntology $(\mathcal{O}')$ and also generate M-rules for the removed portion.

Definition 1. *A sine−$\mathcal{M}$ − $\mathcal{O}$ntology ($\mathcal{O}'$) is a subset of a metamodelled-ontology $\mathcal{M} - \mathcal{O}$, without Clashing Axioms.*

Definition 2. *Consider entities C_1 and c_1 in an M-ontology. If there exists some axiom of the form (at-the-least) $C_1 \sqsubseteq C_2$ and $A(c_1)$, then these are called clashing axioms, where $C_1 \equiv c_1$ and appears more than one type of position with the same IRI.*

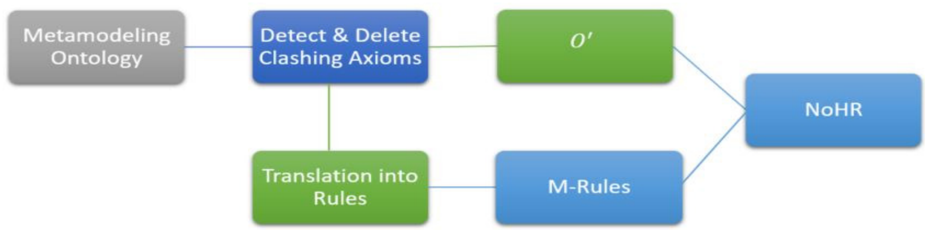

Fig. 1. Proposed Framework

Our framework makes use of two languages: MS for OWL 2 QL as an ontology language, and NoHR as reasoner for hybrid knowledge base.

The proposed framework consist of three functions:

1. The first function checks for clashing axioms in $\mathcal{M} - \mathcal{O}$ and generates the *sine-M* Ontology. To retain the same semantic knowledge between $\mathcal{M} - \mathcal{O}$ and *sine*−$\mathcal{M}$ Ontology, we omit the clashing axioms appearing in the $\mathcal{A}$Box, because omission of general concepts and roles stores in $\mathcal{T}$Box will cause semantic inconsistencies and change the meaning of the represented domain.
2. The second function takes eliminated axioms with attached information as input and translates them into *M-rules*. The basic building blocks for M-rules are the OWL 2 QL assertions under MS, where an assertion is simply an axiom consisting of sets of individuals, classes or roles. The form of assertions for M-O that are suitable for metamodeling can have one of following forms: $p_1(e_1, e_2)$ and $p_2(e_1, e_2, e_3)$, where p_1 can be a class assertion $C(x)$ or disjoint assertion $C_1 \sqcap C_2 \sqsubseteq \bot$ on concepts and p_2 is a role or property $C(x, y)$ and e_1, e_2, e_3 are expressions over $V_{\mathcal{O}}$.Currently we have introduced and focused on the following assertions notions for M-rules. We say that:
   - In mCAssert(e_1, e_2), e_1 represents an object argument and e_2 appears as class argument.
   - In isDisjCAssert(e_1, e_2), e_1, e_2 appear as class arguments.
   - In mRAssert(e_1, e_2, e_3), represents information about Object and Data property.
     - In mOPAssert(e_1, e_2, e_3), e_2 and e_3 represent individual arguments and e_1 appears as property argument.
     - In mDPassert(e_1, e_2, e_3), e_2 and e_3 represent individual arguments and e_1 appears as literal.

3. The third function passes the $\mathcal{O}'$ and *M-rules* to the NoHR reasoner for validation and for querying the hybrid knowledge base.

The basic structure and semantics of $\mathcal{O}'$ after omission of clashing axioms remain the same as $\mathcal{M} - \mathcal{O}$.

Lemma 1. *$\mathcal{O}'$ is said to be a subset of $\mathcal{M} - \mathcal{O}$ if there exists at least some syntactic correspondence between the signature of both $\mathcal{O}'$ and M-O, for the existence of named entities, so $signature(\mathcal{M} - \mathcal{O}) \cap signature(\mathcal{O}') \neq \emptyset$.*

Lemma 2. *Let $\mathcal{M} - \mathcal{O}$ be a metamodelling ontology and $\mathcal{O}'$ be a $sine{-}\mathcal{M} - \mathcal{O}ntology$. A as a concept/unary predicate and R as a role/binary predicate, $\mathcal{O}'$ is a semantic preserving representation model of $\mathcal{M} - \mathcal{O}$.*

$$M - O \models A(a) \iff \mathcal{O}' \models A(a)$$
$$M - O \models R(a, b) \iff \mathcal{O}' \models R(a, b)$$

Example 1. Consider the example mentioned in the introduction, which shows that EndangeredSpecies is a metaconcept for GoldenEagle. First, we strip the metamodeling part from the metamodelled ontology $\mathcal{M} - \mathcal{O}$ and generate $\mathcal{O}'$:

Table 1. $\mathcal{O}'$ subset of $\mathcal{M} - \mathcal{O}$

$\mathcal{M}$-Ontology	$sine{-}\mathcal{M} - \mathcal{O}$ntology ($\mathcal{O}'$)
Ontology BirdKingdom	Ontology BirdKingdom
Classes	Classes
EndangeredSpecies	EndangeredSpecies
Birds	Birds
Eagle	Eagle
GoldenEagle	GoldenEagle
BaldEagle	BaldEagle
Individual	Individual
Harry Types: GoldenEagle	Harry Types: GoldenEagle
Tim Types: BaldEagle	Tim Types: BaldEagle
GoldenEagle Types: EndangeredSpecies	
ObjectProperty	ObjectProperty
Harry Lives_in CPZ	Harry Lives_in CPZ

In Table 1 we can see that the translation of $\mathcal{M} - \mathcal{O}$ to $\mathcal{O}'$did not delete the EndangeredSpecies class, even though it is a metaclass in order to retain the same schematic structure and semantic knowledge in $\mathcal{O}'$as $\mathcal{M} - \mathcal{O}$. So, the translation function just eliminates the fact that makes EndangeredSpecies a metaclass and that is in this case an axiom of the form:

ClassAssertion(:GoldenEagle :EndangeredSpecies)

Then, the translation function transforms the eliminated part from $\mathcal{M} - \mathcal{O}$ into M-rules, which captures the metamodeling semantics of OWL 2 QL. To avoid ambiguity and misinterpretation of the arguments, we have used "is" and "Of" in our predicate i.e. isInstanceOf(?X, ?Y) which gives more clarity of roles of the arguments. After translation, metamodeling axioms are appended to the M-rules.

$$\text{mCAssert(GoldenEagle,EndangeredSpecies)} \tag{1}$$

$$\text{isInstanceOf(?C,?D) :- mCAssert(?C,?D),not isTmCAssert(?C,?D)} \tag{2}$$

$$\text{isTmCAsert(?C,?D) :- mCAsert(?C,?B), mCAsert(?B,?D)} \tag{3}$$

$$\text{isInstanceOf(?X,GoldenEagle) :- GoldenEagle(?X)} \tag{4}$$

$$\text{isInstanceOf(?X,EndangeredSpecies) :- EndangeredSpecies(?X)} \tag{5}$$

$$\text{EndangeredSpecies(?X) :- isInstanceOf(?X,EndangeredSpecies)} \tag{6}$$

We have used one of the notions introduced above, mCAssert(e_1, e_2), where m stands for meta and the argument e_1 represents the class as an individual. *isTmCAssert* is the transitive closure on meta-classes and the key idea for its definition is to stop cycles, which may occur when both arguments of mCAssert are the same or when the arguments form a loop. Next we defined the *isInstanceOf* predicate in the rules. The purpose of this predicate is to show the *instanceOf* hierarchy chain between entities and captures the metamodeling semantics of entities. Rules 4 and 5 help rule 2 to capture all the instances that are linked with meta-Assertions and ease the meta-querying. On the other hand, rule 6 updates the instances listed for EndangeredSpecies in $\mathcal{O}'$along with the instances listed in M-rules. To accommodate metaquerying, our translation function also checks the facts that come along with meta-asserted axioms like *disjoint* axioms. Suppose BaldEagle and GoldenEagle are disjoint in $\mathcal{O}'$and are instances of EndangeredSpecies. Then, to capture the correct semantics of $\mathcal{O}'$in M-rules, we have to define a new predicate called *isDisjCAssert*.

$$\text{isDisjCAssert(GoldenEagle,BaldEagle)} \tag{7}$$

$$\text{isDisjcOf(?X,?Y) :- isDisjCAssert(?Y, ?X)} \tag{8}$$

$$\text{isDisjCAssert(?Y, ?X) :- isDisjCAssert(?X, ?Y)} \tag{9}$$

The *isDisjCAssert* predicate ensures the semantic correspondence between the entities in the ontology and te atoms in rules. Rule 7 is the meta-assertion involving GoldenEagle and BaldEagle. Rule 9 ensures symmetry, and Rule 8 (wit 9) checks for disjoint assertions on instance level and makes GoldenEagle and BaldEagle different individuals. If the asserted meta-axioms come with role assertions, then the translation function will use mRAssert($e_1, e_2, , e_3$). Suppose in Example 1 we have a property axiom, which says that animals are listed in EndangeredSpecies by IUCN, and assume IUCN a metaclass, which makes ListedBy a metaproperty. The translation of this axiom will be:

$$\text{mOPAssert(ListedBy,EndangeredSpecies,IUCN)} \tag{10}$$

$$\text{isOPInstance(?R,?X,?Y) :- mOPAssert(?R,?X,?Y), not isTmOPAssert(?R,?X,?Y)} \tag{11}$$

$$\text{isTmOPAssert(?R,?X,?Y) :- mOPAssert(?R,?X,?Z),mOPAssert(?R,?Z,?Y)} \tag{12}$$

$$\text{ListedBy(?X,?Y) :- isOPInstance(ListedBy,?X,?Y)} \tag{13}$$

Rule 11 shows that the *isOPInstance* predicate helps capturing those entities that are linked together via a relation and *isTmOPAssert* helps avoiding cyclic relations. We omit dealing with other OWL 2 QL axioms, since we assume that the three assertions treated in the M-rules above are sufficient for most basic metamodeling tasks.

4 Query Language

As query language we consider conjunctive queries. A conjunctive query q over $\mathcal{O}$ is an expression **ask where** B (query body), consisting of a conjunction of atoms over $V_{\mathcal{O}}$. We consider a metavariable in a query to be a variable that may refer to metaclasses and metaproperties.

Definition 3. *A metaquery is an expression consisting of meta-predicates p and meta-variables v, where p can have other p as their arguments and v can appear in predicate positions.*

The semantics of queries resort to the interpretation of queried ontology and rules according to metamodeling semantics. We expressed the queries in NoHR, that allow conjunction of predicates with variables in a query and to get the correct substitution for variables, queries need to be ground before being processed by the DL reasoner.

Example 2. Consider a conjunctive query q to Example 1, where q should retrieve all those Birds that are instances of EndangeredSpecies (ES) and live in Central Park Zoo (CPZ):

isInstanceOf(?C, EndangeredSpecies), isInstanceOf(?X, ?C),Birds(?X), Lives_in(?X, CPZ)

With the use of $\mathcal{O}'$and M-rules, we get the set $q_i^{'} = \{q_0^{'}, q_1^{'}, q_2^{'}\}$ is a rewriting for q, where q_0 checks for all the instances of EndangeredSpecies defined by rules and will substitute $?C$ with *GoldenEagle*. q_1 takes the substituion of $?C$ and checks for its instances from both $\mathcal{O}'$and M-rules and will substitute $?X$ with *Harry*. q_2 takes the substition of $?X = Harry$ and checks $\mathcal{O}'$for the property *Lives_in*, if proven true, it can derive the queried goal and the answer to $q_i^{'}$ is equivalent to q for any set of ground facts $\mathcal{A}$.

The query q, according to Definition 3 is clearly a metaquery. The full power of metaquyering, where $\mathcal{T}$box and $\mathcal{A}$box atoms coexist, can be achieved with the use of the generalized predicate *isInstanceOf* from the translation function. We called *isInstanceOf* predicate generalized, as it tries to capture the semantics of first order (FO) constructs like $C(x)$, $C_1 \sqsubseteq C_2$ such that it does not require FO constructs in the query. For instance, in the above example we ask to retrieve all those birds of subclass Eagle who are instances of EndangeredSpecies (ES) and live in Central Park Zoo (CPZ). This query can be answered by the query above, with the fact that GoldenEagle is an instance of EndangeredSpecies and GoldenEagle is a subclass of Eagle as well.

5 Related Work

Higher Order Semantics [4] extends DL-Lite$_\mathrm{R}$ to take metaclasses and metaqueries into account. The interpretation structure follows the Hilog style semantics, which allows the elements in the domain Δ_o to have polymorphic characteristics. For reasoning with the extended $DL-Lite_R$, punning has been adopted. This extension guarantees a low complexity but at the cost of restricted expressivity.

Hybrid Integration of rule systems with ontology languages have shown significant advances and plays a central role in the development of the Semantic Web. Our focus here is on one of the integration techniques known as the hybrid MKNF knowledge bases [7], which consist of a finite number of MKNF rules and a decidable fragment of description logic. The well-founded semantics version of hybrid MKNF knowledge bases offers efficient reasoning and inconsistency handling and NoHR [2] is a system, which provides a platform for querying over a combination of both rules and an ontology. However, there is no work, to the best of our knowledge, that uses NoHR reasoning with metamodeling features.

6 Conclusion and Future Work

We showed that our approach not only allows metamodeling facility in ontology but it allows for writing a slightly restricted form of metaqueries as well. We plan to perform more experiments to evaluate our approach, that is, a comparison with other approaches that support metamodeling feature in OWL 2 QL ontologies. Also we plan to extend the notion of assertions to achieve the full power of meta-querying by using non-monotonic rules.

References

1. Alferes, J.J., Knorr, M., Swift, T.: Query-driven procedures for hybrid MKNF knowledge bases. ACM Transactions on Computational Logic (TOCL) **14**(2), 1–43 (2013)
2. Costa, N., Knorr, M., Leite, J.: Next step for NoHR: OWL 2 QL. In: International Semantic Web Conference. pp. 569–586. Springer (2015)
3. Guizzardi, G., Almeida, J.P.A., Guarino, N., de Carvalho, V.A.: Towards an ontological analysis of powertypes. In: JOWO@ IJCAI (2015)
4. Lenzerini, M., Lepore, L., Poggi, A.: A higher-order semantics for owl 2 ql ontologies. In: Description Logics (2015)
5. Meditskos, G., Bassiliades, N.: Rule-based owl ontology reasoning systems: Implementations, strengths, and weaknesses. In: Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, pp. 124–148. IGI Global (2009)
6. Motik, B.: On the properties of metamodeling in OWL. In: International Semantic Web Conference. pp. 548–562. Springer (2005)
7. Motik, B., Rosati, R.: Reconciling description logics and rules. Journal of the ACM (JACM) **57**(5), 1–62 (2008)