

Non-monotonic reasoning in conceptual modeling and ontology design: A proposal.

Giovanni Casini¹ and Alessandro Mosca²

¹ Centre for Artificial Intelligence Research, CSIR Meraka and UKZN, South Africa
GCasini@csir.co.za

² Free University of Bozen-Bolzano, Faculty of Computer Science, Italy
mosca@inf.unibz.it

Abstract. The *Object Role Modeling language* (ORM2) is nowadays the most widespread fact-based conceptual modeling language in the business world. Recently, it has been proposed an encoding of the core fragment of ORM2 (called ORM2^{zero}) into the description logic \mathcal{ALCQT} , allowing the use of reasoning technologies in the analysis of the schemas. A number of services has been defined there based on the FO semantics of ORM2. On the other hand, in many application domains there is a need for the formalization and modeling of *defeasible information* and *non-monotonic* reasoning services. Here we formalize a possible way of introducing non-monotonic reasoning into ORM2 schemas, enriching the language with special set of new constraints.

1 Introduction

ORM2 (Object Role Modelling 2) is a graphical fact-oriented approach for modelling and querying business domain information, which allows for a verbalisation in a controlled natural language easily understandable by non-technical users [?]. The introduction of a formal semantics for ORM2 and the identification of a decidable fragment of it (see [?] and [?]) opened the doors for the exploitation of reasoning technologies to support the schema design quality, as well as the possibility to exploit ORM2 as ontology design language. In the last years, especially in the field of ontology design, a lot of attention has been devoted to the implementations of forms of *defeasible reasoning*, and various proposals (e.g., [?, ?, ?, ?, ?]) have been made in order to integrate nonmonotonic reasoning mechanisms into description logics (DLs), the main logic formalism used in ontologies representation.

The paper explores a possible way of implementing non-monotonic reasoning in the ORM2 formalism. We focus here on the DL-reformulation [?] of the non-monotonic consequence relation called *Rational Closure* (RC) [?], and we show (i) how to model defeasible information in ORM2 and (ii) how to check consistency and draw conclusions from schemas with defeasible information. Due to the lack of space, here we do not enter deeply into the technical details of the proposal, but rather we describe the main idea and its implications in terms of knowledge representation, and we refer to the related literature when needed.

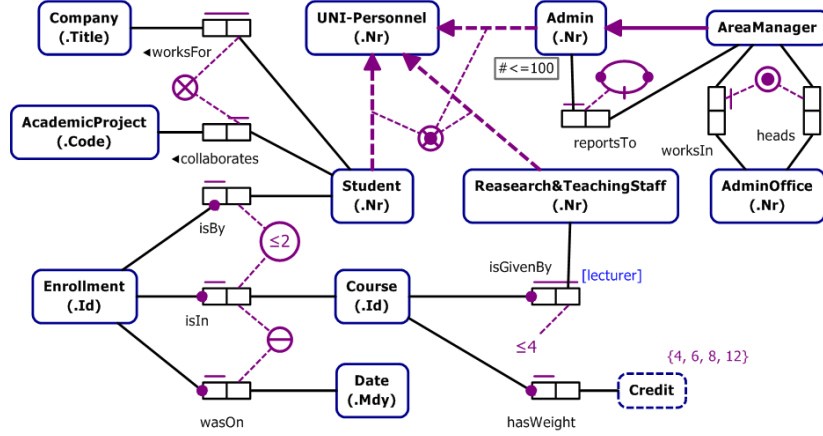


Fig. 1. An ORM2 schema example.

2 Fact-oriented modelling in ORM2

The ‘Fact-oriented modelling’ is a conceptual modelling approach that views the world in terms of simple facts, *i.e.* assertions about *objects playing certain roles* (e.g. ‘Alice is enrolled in the Computer Science program’). In ORM2, objects may be **entities** (e.g. a person or a car) and **values** (e.g. a *character string* or a *number*). Moreover, entities and values are described in terms of the **types** they belong to, where a type is a set of instances (entities and values types are called together ‘object types’). Each entity in the domain of interest is, therefore, an instance of a particular type. The roles played by the entities in a given domain are introduced by means of **predicates**, where each predicate has a given set of **roles** according to its arity. A role is connected to exactly one object type, indicating that the role is played only by the (possible) instances of that type. According to the ORM2 design procedure, after the specification of the relevant object types and predicates, *constraints* must be considered. Hereafter, we give an informal introduction of the constraint graphical representation, together with their intended semantics. Fig. 1 shows an example of an ORM2 schema in an ‘academic domain’ (where the soft rectangles are entity types, the dashed soft rectangles are value types, and the sequences of one or more role-boxes are predicates). The example is not complete w.r.t. the set of all the ORM2 constraints but it aims at giving the feeling of the expressive power of the language. The following are among the constraints included in the schema (see [?] for a comprehensive introduction of the ORM2 constructs, together with their graphical representation):

1. **Subtyping:** solid and dashed arrows representing ‘is-a’ relationships among types.
2. **Partition:** a combination of an **exclusive** constraint (a circled ‘X’) saying that ‘Research&TeachingStaff, Admin, and Student are *mutually disjoint*’, and a **total** constraint (a circled dot) for ‘Research&TeachingStaff, Admin, and Student completely *cover* their common super-type’.

3. **Internal frequency occurrence:** *if* an instance of `Research&TeachingStaff` plays the role of being lecturer in the relation `isGivenBy`, that instance can play the role at most 4 times. *At most one* cardinalities (depicted as continuous bars) are special cases of frequency occurrences called **internal uniqueness** constraints.
4. **External frequency occurrence:** attached to the roles played by `Student` and `Course` for ‘Students are allowed to enrol in the same course *at most* twice’.
5. **External uniqueness:** it is used between the role played by `Course` in `isIn` and the role played by `Date` in `wasOn`, saying that ‘For each combination of `Course` and `Date`, *at most one* `Enrollment` `isIn` that `Course` and `wasOn` that `Date`’.
6. **Mandatory participation:** a dot saying that ‘Each `Course` is given by *at least one* instance of `Research&TeachingStaff`’.

3 The \mathcal{ALCQI} encoding of $\text{ORM2}^{\text{zero}}$

With the main aim of relying on effective technologies to reason about ORM2 schemas, an encoding in the DL \mathcal{ALCQI} , for which tableaux-based reasoning algorithms with a tractable computational complexity exist, has been devised [?]. \mathcal{ALCQI} corresponds to the basic DL \mathcal{ALC} equipped with *qualified cardinality restrictions* and *inverse roles*, and it is a fragment of the OWL2 web ontology language (a complete introduction of the syntax and semantics of \mathcal{ALCQI} can be found in [?]). Apart from the fact that the full ORM2 language is undecidable for several reasons [?], the difference in expressivity between \mathcal{ALCQI} and ORM2 is substantial: \mathcal{ALCQI} does not admit neither set-comparison, nor frequency occurrence statements, about arbitrary role sequences. And moreover, n -ary relations are not part of the \mathcal{ALCQI} language, and reified relations must be introduced in the encoding¹. Due to this limitations, the fragment called $\text{ORM2}^{\text{zero}}$, which is maximal with respect to the expressiveness of \mathcal{ALCQI} and still expressive enough to capture the most frequent usage patterns of the conceptual modelling community, has been identified.

$\text{ORM2}^{\text{zero}}$ corresponds to the fragment of ORM2 equipped with typing, mandatory, subtyping (i.e., exclusive, exhaustive, and partition) constraints, and a restricted version of the set-comparison (i.e., subset, equality, and exclusion) and of the frequency occurrence constraints. The encoding of the semantics of $\text{ORM2}^{\text{zero}}$ is shown in Table 3 where: (i) the \mathbb{E} are *entity type* concepts; (ii) the \mathbb{V} are *value type* concepts; (iii) the \mathbb{A}_R are reified n -ary *relations*; (iv) \mathbb{D} are *domain symbol* concepts; and (v) $1, \dots, n_{\max} + 1$ are DL roles. There, additional *background axioms* are needed in order to force the interpretation of the \mathcal{ALCQI} KB to be correct w.r.t. the corresponding $\text{ORM2}^{\text{zero}}$ schema (the complete proof of the correctness theorem is available at [?]).

It is finally important to say that in ORM2, and in conceptual modeling languages in general, the notion of consistency is slightly different from the classical logical one. As a matter of fact, from a logical point of view a KB \mathcal{K} is considered inconsistent only if we can classically derive a contradiction from it ($\mathcal{K} \models \top \sqsubseteq \perp$ in DL). Instead, dealing with conceptual schemas we generally desire that they

¹ I.e., for each relation R of arity $n \geq 2$, a new atomic concept A_R and n functional roles are introduced.

Table 1. \mathcal{ALCQI} encoding.

Background axioms:	$E_i \sqsubseteq \neg(D_1 \sqcup \dots \sqcup D_l)$ for $i \in \{1, \dots, n\}$ $V_i \sqsubseteq D_j$ for $i \in \{1, \dots, m\}$, and some j with $1 \leq j \leq l$ $D_i \sqsubseteq \bigcap_{j=i+1}^l \neg D_j$ for $i \in \{1, \dots, l\}$ $\top \sqsubseteq A_{\top_1} \sqcup \dots \sqcup A_{\top_{n_{max}}}$ $\top \sqsubseteq (\leq 1i.\top)$ for $i \in \{1, \dots, n_{max}\}$ $\forall i.\perp \sqsubseteq \forall i+1.\perp$ for $i \in \{1, \dots, n_{max}\}$ $A_{\top_n} \equiv \exists 1.A_{\top_1} \sqcap \dots \sqcap \exists n.A_{\top_1} \sqcap \forall n+1.\perp$ for $n \in \{2, \dots, n_{max}\}$ $A_R \sqsubseteq A_{\top_n}$ for each atomic relation R of arity n $A \sqsubseteq A_{\top_1}$ for each atomic concept A
TYPE($R.a, O$)	$\exists \tau(R.a)^-.A_R \sqsubseteq O$
FREQ ⁻ ($R.a, \langle \min, \max \rangle$)	$\exists \tau(R.a)^-.A_R \sqsubseteq \geq \min \tau(R.a)^-.A_R \sqcap \leq \max \tau(R.a)^-.A_R$
MAND($\{R^1.a_1, \dots, R^1.a_n, \dots, R^k.a_1, \dots, R^k.a_m\}, O$)	$O \sqsubseteq \exists \tau(R^1.a_1)^-.A_{R^1} \sqcup \dots \sqcup \exists \tau(R^1.a_n)^-.A_{R^1} \sqcup \dots \sqcup$ $\exists \tau(R^k.a_1)^-.A_{R^k} \sqcup \dots \sqcup \exists \tau(R^k.a_m)^-.A_{R^k}$
R-SET _{Sub} ⁻ (A, B)	$A_R \sqsubseteq A_S$ (entire relations)
R-SET _{Exc} ⁻ (A, B)	$A_R \sqsubseteq A_{\top_n} \sqcap \neg A_S$
R-SET _{Sub} ⁻ (A, B)	$\exists \tau(R.a_i)^-.A_R \sqsubseteq \exists \tau(S.b_j)^-.A_S$ (pair of roles)
R-SET _{Exc} ⁻ (A, B)	$\exists \tau(R.a_i)^-.A_R \sqsubseteq A_{\top_n} \sqcap \neg \exists \tau(S.b_j)^-.A_S$
O-SET _{Iso} ($\{O_1, \dots, O_n\}, O$)	$O_1 \sqcup \dots \sqcup O_n \sqsubseteq O$
O-SET _{Tot} ($\{O_1, \dots, O_n\}, O$)	$O \sqsubseteq O_1 \sqcup \dots \sqcup O_n$
O-SET _{Ex} ($\{O_1, \dots, O_n\}, O$)	$O_1 \sqcup \dots \sqcup O_n \sqsubseteq O$ and $O_i \sqsubseteq \bigcap_{j=i+1}^n \neg O_j$ for each $i = 1, \dots, n$
OBJ(R, O)	$O \equiv A_R$

satisfy a stronger form of consistency, that is, we want that none of the classes in the schema is forced to be empty.

Definition 1 (Strong consistency). A TBox \mathcal{T} is strongly consistent if none of the atomic concepts present in its axioms are forced to be empty, that is, if $\mathcal{T} \not\models A \sqsubseteq \perp$ for every atomic concept A appearing in the inclusion axioms in \mathcal{T} .

4 Rational Closure in \mathcal{ALCQI}

It's time now to briefly present the specialization of the RC [?] procedure for the DL \mathcal{ALCQI} . Among the various proposals in non-monotonic reasoning, we have chosen RC because of its interesting characteristics: (i) the related consequence relation satisfies important logical properties [?, ?]; (ii) the conclusions one can draw are intuitive; (iii) the procedure can be reduced to a series of decisions w.r.t. the classical \mathcal{ALCQI} consequence relation \models .

A TBox \mathcal{T} for \mathcal{ALCQI} consists of a finite set of general inclusion axioms (GCI) of form $C \sqsubseteq D$ (C and D being concepts). Now we introduce also the *defeasible inclusion axiom* $C \sqsubset D$, that is read as ‘Typically, an individual falling under the concept C falls also under the concept D ’. We indicate with \mathcal{D} (DBox) the finite set of such inclusion axioms.

Example 1. Consider a modification of the classical ‘penguin example’, with the concepts P, B, F, I, Fi , respectively read as ‘penguin’, ‘bird’, ‘flying’, ‘insect’, and ‘fish’, and a role $Prey$, where a role instantiation $(a, b):Prey$ read as ‘ a preys for b ’. We can define a defeasible knowledge base (KB) $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ with $\mathcal{T} = \{P \sqsubseteq B, I \sqsubseteq \neg Fi\}$ and $\mathcal{D} = \{P \sqsubset \neg F, B \sqsubset F, P \sqsubset \forall Prey.Fi \sqcap \exists Prey.\top, B \sqsubset \forall Prey.I \sqcap \exists Prey.\top\}$.

RC is a logical procedure that, given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$, decides if a defeasible axiom $C \sqsubseteq D$ is derivable or not from the KB. The aim is to resolve potential conflicts in our KB (*e.g.* penguins do not fly, but, being a subtype of birds, we could also derive that they fly). The basic idea is to order the defeasible information w.r.t. its specificity: an *exceptionality ranking* of the axioms in \mathcal{D} starting from the most general ones (*e.g.* the ones that give us information about birds) up to more specific ones (*e.g.* the ones that give us information about penguins). In case of potential conflicts (*e.g.*, the conclusion that penguins fly and do not fly), the procedure retains the more specific defeasible information (penguins typically do not fly), eliminating the more general one (birds typically fly). Due to the limits in space, we present here a minimal technical explanation of the procedure, referring the reader to [?] for a more extensive presentation of the technicalities and the intuitions behind the procedure.

Given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ and a query $C \sqsubseteq D$, the procedure can be divided into two phases: (i) the definition of the *exceptionality ranking* of the axioms in \mathcal{D} ; (ii) the decision whether $C \sqsubseteq D$ is or is not in the RC of \mathcal{K} .

(i) A concept is considered *exceptional* in a KB $\langle \mathcal{T}, \mathcal{D} \rangle$ if it is classically negated, that is, C is exceptional in $\langle \mathcal{T}, \mathcal{D} \rangle$ if

$$\mathcal{T} \models \bigwedge \overline{\mathcal{D}} \sqcap C \sqsubseteq \perp$$

where $\overline{\mathcal{D}} = \{\neg C \sqcup D \mid C \sqsubseteq D \in \mathcal{D}\}$, and \models is the classical consequence relation associated to \mathcal{ALCQI} . In turn, a defeasible axiom is considered exceptional if its antecedent is exceptional. Let E be a function that, given $\langle \mathcal{T}, \mathcal{D} \rangle$ gives back the exceptional axioms in \mathcal{D} ($E(\mathcal{D}) = \{C \sqsubseteq D \in \mathcal{D} \mid C \text{ exceptional in } \langle \mathcal{T}, \mathcal{D} \rangle\}$). We can define a sequence $\mathcal{E}_0, \mathcal{E}_1, \dots$ of subsets of \mathcal{D} s.t. $\mathcal{E}_0 = \mathcal{D}$ and $\mathcal{E}_{i+1} = E(\mathcal{E}_i)$. Since \mathcal{D} is a finite set, the procedure terminates with a (possibly empty) fixed point of E , that we call \mathcal{E}_∞ .

(ii) Given a query $C \sqsubseteq D$, we check at which level of exceptionality we have to position the antecedent C , that is, we associate C with the defeasible information contained in the lowest \mathcal{E}_i s.t.:

$$\mathcal{T} \not\models \bigwedge \overline{\mathcal{E}_i} \sqcap C \sqsubseteq \perp$$

Hence, we can decide if the defeasible inclusion axiom $C \sqsubseteq D$ is in the lexicographic closure of $\langle \mathcal{T}, \mathcal{D} \rangle$:

$$\langle \mathcal{T}, \mathcal{D} \rangle \Vdash_r C \sqsubseteq D \text{ iff } \mathcal{T} \models \bigwedge \overline{\mathcal{E}_i} \sqcap C \sqsubseteq D$$

where \Vdash_r indicates the inference relation characterizing the Rational Closure.

Example 2. Consider the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ in Example 1. $\overline{\mathcal{D}} = \{\neg P \sqcup \neg F, \neg B \sqcup F, \neg P \sqcup (\forall \text{Prey}. Fi \sqcap \exists \text{Prey}. \top), \neg B \sqcup (\forall \text{Prey}. I \sqcap \exists \text{Prey}. \top)\}$. We obtain the exceptionality ranking of the sequents: $\mathcal{E}_0 = \{B \sqsubseteq F, B \sqsubseteq \forall \text{Prey}. I \sqcap \exists \text{Prey}. \top, P \sqsubseteq \neg F, P \sqsubseteq \forall \text{Prey}. Fi \sqcap \exists \text{Prey}. \top\}$; $\mathcal{E}_1 = \{P \sqsubseteq \neg F, P \sqsubseteq \forall \text{Prey}. Fi \sqcap \exists \text{Prey}. \top\}$.

Assume we want to check the properties of penguins (concept P). Hence, we have to find the exceptionality level of P , that is 1 since $\mathcal{T} \models \bigwedge \overline{\mathcal{E}_0} \sqcap P \sqsubseteq \perp$, but $\mathcal{T} \not\models \bigwedge \overline{\mathcal{E}_1} \sqcap P \sqsubseteq \perp$. Associating to P the defeasible information in \mathcal{E}_1 we are able to derive that penguins do not fly and eat fishes, but not that penguins fly and eat insects.

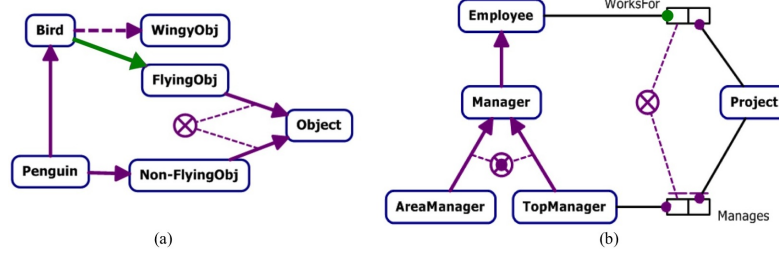


Fig. 2. (a) Defeasible subtyping (b) Defeasible mandatory.

Table 2. \mathcal{ALCQI} encoding of the ORM2^{zero} ‘Non-Flying Birds’ example.

Subtyping: $\text{FlyingObject} \sqcup^{\text{NonFlyingObject}} \sqsubseteq \text{Object}$
 $\text{FlyingObject} \sqsubseteq \neg^{\text{NonFlyingObject}}$
 $\text{Penguin} \sqsubseteq \text{Bird}, \text{Penguin} \sqsubseteq^{\text{NonFlyingObject}}$
 $\text{Bird} \sqsubseteq \text{WingyObject}, \text{Bird} \sqsubseteq \text{FlyingObject}$

5 Defeasible constraints for ORM2

This section introduces a new set of ORM2 defeasible constraints. Such constraints, that are green in our examples, have exactly the same meaning of the defeasible inclusion axioms defined above: the constraint *typically* holds, but there could be exceptional cases that do not respect it.

Defeasible subtyping relation. The green arrow indicates that each element of the class C is also an element of the class D , if not informed of the contrary. While the classical subtype relation is encoded as $C \sqsubseteq D$, the new defeasible connection is encoded by $C \sqsubseteq^{\text{D}} D$.

Example 3 (Defeasible subtype relation). Consider Fig. 2(a). The ORM2 schema represents the classic penguin example: penguins are birds and do not fly, while birds fly and have wings. The encoding procedure of the classical ORM2 schema into \mathcal{ALCQI} gives back the TBox \mathcal{T} in Table 2, that implies $\mathcal{T} \models \text{Penguin} \sqsubseteq \perp$, *i.e.* the concept **Penguin** must be empty. We can modify the KB introducing defeasible information, in particular stating that birds *typically* fly. In such a way we obtain a new KB with $\text{Bird} \sqsubseteq^{\text{D}} \text{FlyingObject}$, substituting the corresponding classical axiom. Now we can derive the same kind of conclusions as in Example 2 (*e.g.*, we cannot derive that penguins fly).

Defeasible mandatory participation. We introduce a defeasible version of the mandatory participation constraint ‘•’. If the connection between a class C and a relation R is accompanied by the green version of the mandatory constraint, it is read as ‘each element of the class C participates to the relation R , if we are not informed of the contrary’. The mandatory participation of the class **B** to the role A_N is encoded into the axiom $\text{B} \sqsubseteq \exists f1^- . A_N$, hence we make it defeasible using $\text{B} \sqsubseteq^{\text{D}} \exists f1^- . A_N$.

Table 3. *ALCQI* encoding of the ‘Non-Managing Employees’ example.

Typing:	$\text{WorksFor} \sqsubseteq \exists f1^-. \text{Employee}, \text{WorksFor} \sqsubseteq \exists f2^-. \text{Project}$ $\text{Manages} \sqsubseteq \exists f1^-. \text{TopManager}, \text{Manages} \sqsubseteq \exists f2^-. \text{Project}$
Frequency:	$\exists f1^-. \text{Manages} \sqsubseteq = 1 f1^-. \text{Manages}$
Mandatory:	$\text{Employee} \sqsubseteq \exists f1^-. \text{WorksFor}$ $\text{TopManager} \sqsubseteq \exists f1^-. \text{Manages}$ $\text{Project} \sqsubseteq \exists f2^-. \text{WorksFor}$ $\text{Project} \sqsubseteq \exists f2^-. \text{Manages}$
Exclusion:	$\exists f1^-. \text{WorksFor} \sqsubseteq A_{T2} \sqcap \neg \exists f1. \text{Manages}$
Subtyping:	$\text{Manager} \sqsubseteq \text{Employee} \sqcap (\text{AreaManager} \sqcup \text{TopManager})$ $\text{AreaManager} \sqsubseteq \neg \text{TopManager}$

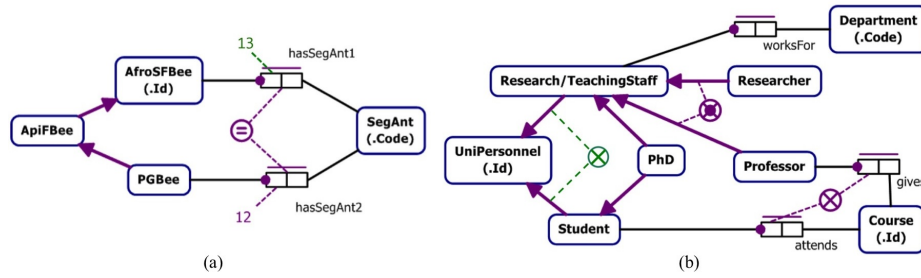


Fig. 3. (a) Defeasible frequency occurrence (b) Defeasible exclusive subtyping.

Example 4 (Defeasible mandatory participation). Consider Fig. 2(b). The schema represents the organization of a firm: the class **Manager** is a subtype of the class **Employee**, and every employee *must* work for a project. On the other hand, every top manager mandatorily manages a project, and **TopManager** is a subtype of **Manager**. The correspondent *ALCQI* TBox \mathcal{T} is in table 3. Since managing and working for a project are not compatible roles, $\mathcal{T} \models \text{TopManager} \sqsubseteq \perp$. Instead, if we declare that *typically* an employee works for a project, we end up considering the top managers as exceptional employees, and we obtain a KB as the one in Table 3, but with $\text{Employee} \sqsubseteq \exists f1^-. \text{WorksFor}$ instead of $\text{Employee} \sqsubseteq \exists f1^-. \text{WorksFor}$. Since **TopManager** is not consistent with the TBox plus the defeasible axiom, we cannot associate the latter to **TopManager** and, despite we know that normally an employee works for a project, we are not forced to such a conclusion about the top managers.

Defeasible internal frequency occurrence. We can introduce a form of defeasible frequency constraint, that binds the number of times an instance of a particular type can play a certain role, if we have not more specific information imposing different constraints. The frequency constraint stating how many times each instance of a type B can play a role A_N is encoded by $\exists f1^-.A_N \sqsubseteq i f1^-.A_N$ ($i \in \mathbb{N}$), and hence made defeasible using $\exists f1^-.A_N \sqsubset = i f1^-.A_N$.

Example 5 (Defeasible internal frequency occurrence). A taxonomization problem about bee species in sub-Saharan Africa: most male afrotropical bees have thirteen

Table 4. *ALCQI* encoding of the ‘Bees classification’ example.

Typing:	$\text{HasSegAnt1} \sqsubseteq \exists f1^-. \text{AfroSFBee}, \text{HasSegAnt1} \sqsubseteq \exists f2^-. \text{SegAnt}$ $\text{HasSegAnt2} \sqsubseteq \exists f1^-. \text{PGBee}, \text{HasSegAnt2} \sqsubseteq \exists f2^-. \text{SegAnt}$
Frequency:	$\exists f1^-. \text{HasSegAnt1} \sqsubseteq = 13 f1^-. \text{HasSegAnt1}$ $\exists f1^-. \text{HasSegAnt2} \sqsubseteq = 12 f1^-. \text{HasSegAnt2}$
Mandatory:	$\text{AfroSFBee} \sqsubseteq \exists f1^-. \text{HasSegAnt1}$ $\text{PGBee} \sqsubseteq \exists f1^-. \text{HasSegAnt2}$
Equality:	$\text{HasSegAnt1} \equiv \text{HasSegAnt2}$
Subtyping:	$\text{PGBee} \sqsubseteq \text{ApiFBee}, \text{ApiFBee} \sqsubseteq \text{AfroSFBee}$

Table 5. *ALCQI* encoding of the ‘University Personnel’ example.

Typing:	$\text{WorksFor} \sqsubseteq \exists f1^-. \text{R\&TStaff}, \text{WorksFor} \sqsubseteq \exists f2^-. \text{Department}$ $\text{Attends} \sqsubseteq \exists f1^-. \text{Student}, \text{Attends} \sqsubseteq \exists f2^-. \text{Course}$ $\text{Gives} \sqsubseteq \exists f1^-. \text{Professor}, \text{Gives} \sqsubseteq \exists f2^-. \text{Course}$
Mandatory:	$\text{R\&TStaff} \sqsubseteq \exists f1^-. \text{WorksFor}$ $\text{Professor} \sqsubseteq \exists f1^-. \text{Gives}$ $\text{Student} \sqsubseteq \exists f1^-. \text{Attends}$
Exclusion:	$\exists f1^-. \text{Attends} \sqsubseteq A_{T2} \sqcap \neg \exists f1^-. \text{Professor}$
Subtyping:	$\text{Researcher} \sqsubseteq \text{R\&TStaff} \sqcap \neg \text{Professor}, \text{Professor} \sqsubseteq \text{R\&TStaff}$ $\text{R\&TStaff} \sqsubseteq \text{UniPersonnel} \sqcap (\text{Researcher} \sqcup \text{Professor})$ $\text{Student} \sqsubseteq \text{UniPersonnel}, \text{Student} \sqsubseteq \neg \text{R\&TStaff}$

segment antennae (*SegAnt*), but male belonging to the parasite genus (*PGBee*), that belongs to the apidae family (*ApiFBee*) of the afrotropical super family, has only twelve segment antennae. We can formalize such information in the ORM2 schema in Fig. 3(a), but such a schema is not strongly consistent, since its translation in *ALCQI* (Table 4) implies $\text{PGBee} \sqsubseteq \perp$. Introducing a form of defeasible frequency occurrence for the more general type *AfroSFBee*, that is encoded into the defeasible inclusion axiom $\exists f1^-. \text{HasSegAnt1} \sqsubseteq = 13 f1^-. \text{HasSegAnt1}$, we preserve strong consistency, since the type *PGBee* turns out to be exceptional, and we don’t have to consider the defeasible axiom when reasoning about *PGBee*. Hence we can derive that, if we are not informed of the contrary, male afrotropical bees have thirteen segment antennae, while the males of the parasite genus have twelve segment antennae, but still being a subtype of *AfroSFBee*.

Defeasible disjointness. It’s a defeasible exclusive constraint, stating that two types or the participation to two distinct roles are mutually disjoint, if not informed of the contrary. In the *ALCQI* encoding, an axiom $A \sqsubseteq \neg B$ is changed into an axiom $A \sqsubseteq \neg B$

Example 6 (Defeasible disjointness). In the university we distinguish between the two disjoint classes of students and research/teaching staff, in turn partitioned into professors and researchers. Assume we add to the schema the class of the PhD students, that are students, but are also part of the research/teaching staff. In the classical formulation we would derive $\text{PhD} \sqsubseteq \perp$, since students and research staff must be disjoint class, but if we make defeasible the disjointness axiom ($\text{Student} \sqsubseteq \neg \text{R\&TStaff}$), we

conclude that normally the individuals in one class cannot be in the other, but the PhD students are an exceptional class that shares the properties of both the students and the research/teaching staff. Moreover, since the research/teaching staff is partitioned into professors and researches, and since giving courses, a mandatory property of the professors, is not compatible with attending to them, we can derive that PhD students are a subtype of the researchers.

Consistency. As seen at the end of Sect. 3, the notion of consistency we have to deal with is the *strong consistency*, that is, a schema is consistent if we are not forced to conclude about any type that it is empty. The examples above show that the introduction of defeasible constraints into $\text{ORM2}^{\text{zero}}$ allows to build schemas that in the standard notation would be considered inconsistent (w.r.t. strong consistency), but that, once introduced the defeasible constraints, allow for an instantiation such that all the classes can be non-empty. Hence it is necessary to redefine the notion of consistency check in order to deal with such situations. If one decide to rely on the ranking procedure presented above, it is sufficient to check the exceptionality ranking of the KB: if a concept C is exceptional, then it represents an atypical situation, but that is compatible with the information conveyed by the defeasible inclusion axioms (consider the exceptional types in our examples). In RC the only case in which a concept appearing in the axioms is necessarily empty is when $\mathcal{E}_\infty \neq \emptyset$, *i.e.*, despite we eliminate all the defeasible axioms we are allowed to, the antecedents of the axioms in \mathcal{E}_∞ still are negated, *i.e.*, they are empty even in the most exceptional situations. Note that schema that it's inconsistent w.r.t. its non-defeasible part, is inconsistent also in the following new definition.

Definition 2 (Strong consistency). *A KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ is strongly consistent if none of the atomic concepts present in its axioms are forced to be empty, that is, if $\mathcal{E}_\infty \neq \emptyset$.*

6 Conclusions and further work

We have presented a way to implement a form of defeasible reasoning into the ORM2 formalism. Exploiting the possibility of encoding $\text{ORM2}^{\text{zero}}$ into the description logic \mathcal{ALCQI} on one hand, and a procedure appropriate for modeling RC into DLs on the other, we have defined a set of new ORM2 constraints that are appropriate for modeling defeasible information. Once translated into \mathcal{ALCQI} , these constraints allow to use the procedures characterizing RC to reason on $\text{ORM2}^{\text{zero}}$ schemas.

The present proposal deals only with reasoning on the information contained in the TBox obtained from an ORM2 schema, but, once we have done the RC of the TBox, we can think also of introducing an ABox, that is, the information about a particular domain of individuals; a first proposal in such direction is in [?]. At the moment we still also lack a semantic characterization of RC in DLs (a first step in such a direction is in [?]).

An immediate extension of the present work will be to determine if there are others ORM2 constraints that could be introduced also in their defeasible versions:

we have to consider such constraints both from the modeling and the technical point of view (*i.e.*, we have to check if such defeasible constraints make sense from the perspective of the modeler and if they are implementable in our logical framework). Since all the constraints in an $\text{ORM2}^{\text{zero}}$ schema are encoded into inclusion axioms $C \sqsubseteq D$, in principle every $\text{ORM2}^{\text{zero}}$ constraint can be modified into a defeasible version using \sqsubseteq , but we need feedback from the ORM2 community in order to understand which defeasible constraints would be desirable. The introduction of defeasible constraints in other conceptual modeling languages as ER and UML is another possible development. Eventually, it would be interesting to investigate the use of defeasible constraints as a revision tool: dealing with strong consistency, in many cases we have the possibility to transform an inconsistent schema into a consistent one just substituting some classical constraints with their defeasible versions, instead of simply eliminating them.