

# Polynomial encoding of ORM conceptual models in $\mathcal{CFDI}_{nc}^{\forall-}$

Pablo Rubén Fillottrani<sup>1,2</sup>, C. Maria Keet<sup>3</sup>, David Toman<sup>4</sup>

<sup>1</sup> Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, [prf@cs.uns.edu.ar](mailto:prf@cs.uns.edu.ar)

<sup>2</sup> Comisión de Investigaciones Científicas, Provincia de Buenos Aires, Argentina

<sup>3</sup> Department of Computer Science, University of Cape Town, South Africa  
[mkeet@cs.uct.ac.za](mailto:mkeet@cs.uct.ac.za)

<sup>4</sup> Cheriton School of Computer Science, University of Waterloo, Canada  
[david@cs.uwaterloo.ca](mailto:david@cs.uwaterloo.ca)

**Abstract.** The use of conceptual models has long been confined to the data analysis stage of software development. In recent years, this has been extended to use them at run-time as well, for, among others, querying large amounts of data. This brings afore the need to have tractable logic-based reconstructions of the conceptual models, i.e., in at most PTIME. We provide such a logic-based reconstruction for most of ORM using the Description Logic language  $\mathcal{CFDI}_{nc}^{\forall-}$ , which has several features important for conceptual models, notably  $n$ -ary relationships, complex identification constraints, and role subsumption. The encoding captures over 96% of the constructs used in practice in the set of 33 ORM diagrams analysed. The results are easily transferable to EER and UML Class diagrams, with an even greater coverage.

## 1 Introduction

While for many years conceptual models were developed and then shelved upon implementation or used ‘offline’ for documentation purposes, recent years has seen an increase in using such precious resources computationally. One strand of investigation focuses on expressiveness and a logic foundation to compute satisfiability and detect inconsistencies over the TBox only, among others [2, 5, 8, 16, 24], and several implementations exists using various technologies that are more or less scalable; e.g., using OCL [16], Alloy [8, 23], or a DL reasoner [19]. Another looks at usage during runtime for a range of purposes, such as using a conceptual model for scalable test data generation [35] and for designing [6] and executing [13] queries. Approximations of conceptual models are used also deeper into the machinery of querying databases, in particular during various stages of query compilation, e.g., when reasoning about duplicates [39]. The former purpose requires a logic foundation using a (very) expressive logic, whereas the latter requires computationally better behaved logics to keep the whole process feasible. What such a tractable logic language looks like that captures most, or all, important conceptual data modelling language features, has received little

attention, however. To the best of our knowledge, there are only four efforts to capture a fragment of which concept or full satisfiability checking is (or is claimed to be) in polynomial time (or less): one for ER [2], two for UML [1, 25], and one for ORM [35], with the first one logic-driven and the last one implementation-driven, and they all differ substantially in coverage of features. The main general question, thus, is: what is a useful fragment for tractable runtime usage of a conceptual data model? Then, how to formalise it.

To answer this question, we choose to focus on ORM first, for it is strictly more expressive than ER and UML class diagrams, and then facilitating transferability of the results. Given that there is a lot of insight in computational complexity of Description Logic (DL) languages and which ones are in P, we zoom in on those for a logic-based reconstruction. It appears that the DL  $\mathcal{CFDI}_{nc}^{\forall-}$  [40] is a good fit and can capture 96.5% of the ORM models in a dataset of 33 ORM models that are part of the dataset of 101 conceptual models (dataset available from [28]). This is chiefly thanks to ‘trading’ costly but lesser used covering constraints for the more often used arbitrary identifiers and  $n$ -ary relationships.

In the remainder of the paper, we first provide background definitions in Section 2. The main results are presented in Section 3, and are discussed and compared with related research in Section 4. We summarize our results and briefly outline future work in Section 5.

## 2 Preliminaries

The two preliminaries are the DL  $\mathcal{CFDI}_{nc}^{\forall-}$  and the ORM language, so as to keep the paper self-contained.

### 2.1 The Description Logic $\mathcal{CFDI}_{nc}^{\forall-}$

All members of the  $\mathcal{CFD}$  family of DLs are fragments of FOL with underlying signatures based on disjoint sets of unary predicate symbols called *primitive concepts*, constant symbols called *individuals* and unary function symbols called *features*.

**Definition 1** ( $\mathcal{CFDI}_{nc}^{\forall-}$  Knowledge Bases) Let  $F$ ,  $PC$  and  $IN$  be disjoint sets of (names of) functional features, primitive concepts and individuals, respectively. A *path function*  $Pf$  is a word in  $F^*$ , and we denote the empty word by  $id$  and concatenation by “.”. Metadata and data in a  $\mathcal{CFDI}_{nc}^{\forall-}$  knowledge base  $\mathcal{K}$  are respectively defined by a *TBox*  $\mathcal{T}$  and an *ABox*  $\mathcal{A}$ . A TBox  $\mathcal{T}$  consists of a finite set of *inclusion dependencies* of the form

$$\begin{aligned} A \sqsubseteq B, \quad A \sqsubseteq \neg B, \quad A \sqsubseteq \forall f.B, \quad \forall f.A \sqsubseteq B, \quad A \sqsubseteq \exists f^{-1}, \\ \text{or } A \sqsubseteq B : Pf_1, \dots, Pf_k \rightarrow Pf \end{aligned}$$

where  $A, B \in PC$ ,  $f \in F$ , and  $Pf_i \in F^*$ . A concept “ $B : Pf_1, \dots, Pf_k \rightarrow Pf$ ” that participates in the last dependency is called a *path functional dependency* (PFD). An ABox  $\mathcal{A}$  consists of a finite set of facts in the form of *concept assertions*  $A(a)$ , and *function assertions*  $f(a) = b$  where  $A \in PC$ ,  $f \in F$ , and  $a, b \in IN$ . Any PFD

occurring in  $\mathcal{T}$  must also satisfy a *regularity* condition by adhering to one of the following two forms:

$$C : \text{Pf} . \text{Pf}_1, \text{Pf}_2, \dots, \text{Pf}_k \rightarrow \text{Pf} \quad \text{or} \quad C : \text{Pf} . \text{Pf}_1, \text{Pf}_2, \dots, \text{Pf}_k \rightarrow \text{Pf} . g. \quad (1)$$

A PFD is a *key* if it adheres to the first of these forms.

The semantics is defined in the standard way with respect to an interpretation  $\mathcal{I} = (\Delta, (\cdot)^{\mathcal{I}})$ , where  $\Delta$  is a domain of “objects” and  $(\cdot)^{\mathcal{I}}$  an interpretation function that fixes the interpretation of primitive concepts  $A$  to be subsets of  $\Delta$ , features  $f$  to be total functions on  $\Delta$ , and individuals  $a$  to be elements of  $\Delta$ . The interpretation function is extended to path expressions by interpreting *id*, the empty word, as the identity function  $\lambda x.x$ , concatenation as function composition, and to derived concept descriptions as follows:

$$\begin{aligned} (\neg A)^{\mathcal{I}} &= \Delta \setminus A^{\mathcal{I}} \\ (\forall f.A)^{\mathcal{I}} &= \{x \mid f^{\mathcal{I}}(x) \in A^{\mathcal{I}}\} \\ (\exists f^{-1})^{\mathcal{I}} &= \{x \mid \exists y \in \Delta : f^{\mathcal{I}}(y) = x\} \\ (C : \text{Pf}_1, \dots, \text{Pf}_k \rightarrow \text{Pf})^{\mathcal{I}} &= \{x \mid \forall y \in C^{\mathcal{I}} : (\bigwedge_{i=1}^k \text{Pf}_i^{\mathcal{I}}(x) = \text{Pf}_i^{\mathcal{I}}(y)) \\ &\quad \Rightarrow \text{Pf}^{\mathcal{I}}(x) = \text{Pf}^{\mathcal{I}}(y)\} \end{aligned}$$

An interpretation  $\mathcal{I}$  satisfies an inclusion dependency  $C \sqsubseteq D$  if  $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ , a concept assertion  $A(a)$  if  $a^{\mathcal{I}} \in A^{\mathcal{I}}$ , and a function assertion  $f(a) = b$  if  $f^{\mathcal{I}}(a^{\mathcal{I}}) = b^{\mathcal{I}}$ .  $\mathcal{I}$  satisfies a knowledge base  $\mathcal{K}$  if it satisfies each inclusion dependency and assertion in  $\mathcal{K}$ . In addition, every  $\mathcal{CFDI}_{nc}^{\forall-}$  knowledge base must satisfy the following two conditions.

1. (*inverse feature and value restriction interaction*) If  $\{A \sqsubseteq \exists f^{-1}, \forall f.A' \sqsubseteq B\} \subseteq \mathcal{T}$  then (a)  $A \sqsubseteq A' \in \mathcal{T}$ , (b)  $A' \sqsubseteq A \in \mathcal{T}$  or (c)  $A \sqsubseteq \neg A' \in \mathcal{T}$ .
2. (*inverse feature and PFD interaction*) Any non-key PFD occurring in  $\mathcal{T}$  that involves features used in the  $\exists f^{-1}$  concept in  $\mathcal{T}$  must also satisfy a *stronger* regularity condition by adhering to the following form:

$$C : \text{Pf} . f, \text{Pf}_2, \dots, \text{Pf}_k \rightarrow \text{Pf} . g. \quad (2)$$

**Proposition 1** ([40]). *Consistency of  $\mathcal{CFDI}_{nc}^{\forall-}$  knowledge bases is complete for PTIME.*

Other reasoning tasks, such as logical implication and concept consistency reduce (linearly) to knowledge base consistency. Relaxing either of the aforementioned conditions leads to EXPTIME and PSPACE completeness, respectively [40]. Note also, that condition (2) imposed on PFDs applies only to non-key PFDs. Overall, however, the restrictions do not seem to impact the modeling utility of  $\mathcal{CFDI}_{nc}^{\forall-}$  in relation to keys and functional constraints. Indeed, arbitrary functional dependencies in relational schema are easily captured. Finally and for convenience  $\mathcal{CFDI}_{nc}^{\forall-}$  supports additional syntax, e.g., subsumptions of the form  $\exists f^{-1} \sqsubseteq A$ . This additional syntax is mere syntactic sugar and can be equivalently expressed in  $\mathcal{CFDI}_{nc}^{\forall-}$  as defined above [40].

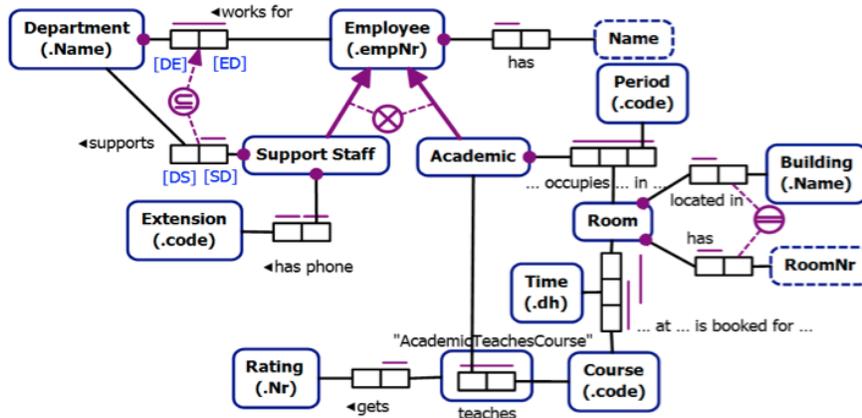


Fig. 1. Sample ORM2 diagram for a particular university where, among others, each Support Staff must have exactly one Extension that is identified by a code, Support Staff and Academics are subtypes of Employee and are disjoint, each Room is identified by the Building and its RoomNr, each Employee has exactly one Name, and a support staff that supports a department also works for that department.

## 2.2 Brief overview of ORM2

Object-Role Modelling as a general term is also known as Fact-Based Modelling and comprises several notational variants with varying language features, ranging from its predecessor NIAM and its recent notation in the proprietary CogNIAM tool of PNA, to the FCO-IM flavour [4], to ORM as in [21] popularised with VisioModeler 3.1 and ORM2 [22] popularised with the NORMA tool [15] as plugin for MS Visio, among others. We use Halpin’s notation [22] in the remainder of the paper for its compactness. As we are interested in the static, structural components, we ignore deontic constraints<sup>5</sup>—they did not occur in any of the evaluated ORM models anyway [28]—and derived constraints (ORM’s version of methods). ORM2 has four different kinds of named entities:

- *entity type*, which is like an EER entity type and UML class and may be objectified, and is denoted with a blue rectangle with round corners;
- *value type*, which is an entity type that has a binary fact type (relationship) to a data type, denoted with a blue dashed rectangle with round corners;
- *role* that each entity/value type plays in a fact type, denoted with a small rectangle and connected to the object type or value type;
- *n-ary fact type* ( $n \geq 1$ ) that relates entity types to each other or entity types to value types and they have to be elementary (uniqueness constraint spanning  $n$  or  $n - 1$  roles), denoted with a rectangle composed of the roles.

Typically, roles and fact types are named automatically, but this can be added, as indicated with the user-named role [DE] in Fig. 1; note that the text next to the fact types are ‘readings’ for model verbalisations, not fact type names.

<sup>5</sup> Refer to [36] for a treatment of deontic constraints in the context of DLs and SBVR

ORM has many constraint types (some of which are used rarely [28]); a non-exhaustive example is shown in Fig. 1. The ones used in the figure are mandatory (small blob), uniqueness (line over rectangle), reference schemes (simple identifiers, e.g., .empNr), and external identifiers (circle with double horizontal line), subtype (arrow) and subset (over roles, fact types, paths), disjointness (encircled cross); others include coverage, constraints on values, and 11 relationship constraints (a.o., transitive, irreflexive).

As transformations exist from one ORM model to UML and to ER [22], ORM is ‘more conceptual’ than the other models for one does not have to commit to an implementation paradigm upfront. Moreover, the notation is also used for business rules specifications [32].

### 3 ORM2<sub>cf</sub>d fragment into $\mathcal{CFDI}_{nc}^{\forall-}$

While we primarily focus on ORM to  $\mathcal{CFDI}_{nc}^{\forall-}$  mapping here, we also aim for an easy extension to EER and UML Class diagrams, the ability to use it for inter-model assertions across models represented in different languages [17], and for unification of the conceptual modelling language families for a widely used subset of features. To this end, we first define the syntax and semantics of a ‘generic’ conceptual data modelling language, which we name  $\mathcal{CM}_{com}^-$ , as it is, essentially, a proper fragment of  $\mathcal{CM}_{com}$ —used as common conceptual data modelling language for EER, UML, and ORM [30]—without covering and disjointive mandatory constraints, and with limited cardinalities and a more precise definition of relationship subsumption and disjointness.  $\mathcal{CM}_{com}^-$  contains those features mappable into  $\mathcal{CFDI}_{nc}^{\forall-}$  (as described in Section 3.3) and captures a subset of features of ORM, named ORM2<sub>cf</sub>d. Thus, within the scope of this paper, one can equate  $\mathcal{CM}_{com}^-$  and ORM2<sub>cf</sub>d.

#### 3.1 Syntax

The syntax is introduced first, and subsequently illustrated with an example. We assume a transformation where an ORM value type is encoded in  $\mathcal{CM}_{com}^-$  as an attribute, and note that recursive relationships are allowed such that a class can participate more than once in a relationship.

**Definition 2 (Conceptual Data Model  $\mathcal{CM}_{com}^-$  syntax)** A  $\mathcal{CM}_{com}^-$  conceptual data model is a tuple  $\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}_R, \text{CARD}_A, \text{ISAC}, \text{ISAR}, \text{ISAU}, \text{DISJ}_C, \text{DISJ}_R, \text{ID}, \text{EXTID}, \text{FD}, \text{OBJ}, \text{REX})$  such that:

1.  $\mathcal{L}$  is a finite alphabet partitioned into the sets:  $\mathcal{C}$  (*object type* symbols),  $\mathcal{A}$  (*attribute* symbols),  $\mathcal{R}$  (*relationship* symbols),  $\mathcal{U}$  (*role* symbols), and  $\mathcal{D}$  (*domain* symbols); the tuple  $(\mathcal{C}, \mathcal{A}, \mathcal{R}, \mathcal{U}, \mathcal{D})$  is the *signature* of  $\Sigma$ .
2.  $\text{ATT}$  is a function that maps a class symbol in  $\mathcal{C}$  to an  $\mathcal{A}$ -labeled tuple over  $\mathcal{D}$ , so that  $\text{ATT}(C) = \{A_1 : D_1, \dots, A_h : D_h\}$  where  $h$  is a non-negative integer.
3.  $\text{REL}$  is a function that maps a relationship symbol in  $\mathcal{R}$  to an  $\mathcal{U}$ -labeled tuple over  $\mathcal{C}$ ,  $\text{REL}(R) = \{U_i : C_i\}_{i=1}^h$ ,  $h \geq 1$ ,  $U_i \neq U_j$  if  $i \neq j$ ,  $h$  is called the *arity* of  $R$ , and if  $(U_i, C_i) \in \text{REL}(R)$  then  $\text{PLAYER}(R, U_i) = C_i$  and  $\text{ROLE}(R, C_i) = U_i$ . The signature of the relation is  $\sigma_R = \{U_i\}_{i=1}^n$ .

4.  $\text{CARD}_R$  is a partial function  $\text{CARD}_R : \mathcal{R} \times \mathcal{U} \rightarrow \{0, 1\} \times \{1, \infty\}$  denoting cardinality constraints. We denote with  $\text{CMIN}(R, U)$  and  $\text{CMAX}(R, U)$  the first and second component of  $\text{CARD}_R$ .
5.  $\text{CARD}_A$  is a partial function  $\text{CARD}_A : \mathcal{C} \times \mathcal{A} \rightarrow \{0, 1\} \times \{1, \infty\}$  denoting multiplicity constraints for attributes, such that  $\text{CARD}_A(C, A)$  is defined iff  $(A, D) \in \text{ATT}(C)$  for some  $D \in \mathcal{D}$ . We denote with  $\text{CMIN}(C, A)$  and  $\text{CMAX}(C, A)$  the first and second component of  $\text{CARD}_A$ .
6.  $\text{ISA}_C$  is a binary relationship  $\text{ISA}_C \subseteq \mathcal{C} \times \mathcal{C}$ .
7.  $\text{ISA}_R$  is a binary relationship  $\text{ISA}_R \subseteq \mathcal{R} \times \mathcal{R}$ .  $\text{ISA}_R$  between relationships is restricted to relationships with compatible signatures.
8.  $\text{ISA}_U$  is a binary relationship  $\text{ISA}_U \subseteq \mathcal{U} \times \mathcal{U}$ .
9.  $\text{DISJ}_C$  is a relationship over  $2^{\mathcal{C}} \times \mathcal{C}$ , describing disjointness partitions over groups of ISA that share the same superclass.
10.  $\text{DISJ}_R$  is a subset of  $2^{\mathcal{R}}$ , describing disjointness over a group of relations (with compatible signatures).
11.  $\text{ID}$  is a partial function,  $\text{ID} : \mathcal{C} \rightarrow 2^{\mathcal{A}}$ , that maps a class symbol in  $\mathcal{C}$  to its identifier (key) attributes;  $\text{CARD}_A(C, A) = (1, 1)$  holds for each  $A \in \text{ID}(C)$ .
12.  $\text{EXTID}$  is called an external uniqueness, which is a partial function that maps a class to a set of sets of relation-role pairs or attributes,  $\text{EXTID} : \mathcal{C} \rightarrow 2^{2^{(\mathcal{R} \times \mathcal{U}) \cup \mathcal{A}}}$ , where for any  $S \in \text{EXTID}(C)$  it holds that at least one pair  $(R, U) \in S$ , and whenever  $(R, U) \in S$  then  $\text{PLAYER}(R, U) = C$ ,  $\text{CARD}_R(R, U) = (1, 1)$ , and when  $A \in S$  then  $\text{CARD}_A(C, A) = (1, 1)$ .
13.  $\text{FD}$  is a functional dependency assertion, a partial function  $\text{FD} : \mathcal{R} \rightarrow 2^{2^{\mathcal{U}}} \times \mathcal{U}$  such that  $\text{FD}(R)$  may be defined only if its arity is  $\geq 2$  and for all  $U_i, U \in \mathcal{U}$  such that  $(\{U_i\}, U) \in \text{FD}(R)$  then exist  $C_i, C$  such that  $U_i : C_i \in \text{REL}(R)$  and  $U : C \in \text{REL}(R)$ .
14.  $\text{OBJ}$  is an objectification partial function that maps an  $n$ -ary relationship symbol  $R \in \mathcal{R}$  to a  $C \in \mathcal{C}$ , i.e.,  $\text{OBJ} : \mathcal{R} \rightarrow \mathcal{C}$ . Whenever  $\text{OBJ}(R) = C$  and  $U_i : C_i \in \text{REL}(R)$ ,  $1 \leq i \leq n$  then there exist  $n$  new binary relationships  $R_i \in \mathcal{R}$  such that  $\text{REL}(R_i) = \{U_i^1 : C, U_i^2 : C_i\}$ , with  $U_i^1, U_i^2 \in \mathcal{U}$ ;  $\text{EXTID}(C) = \{(U_i, C_i) \mid 0 < i \leq n\}$ ;  $\text{CARD}_R(R_i, U_i^1) = (1, 1)$  and  $\text{CARD}_R(R_i, U_i^2) = (0, 1)$ .
15.  $\text{REX}$  is a subset of  $2^{\mathcal{U}}$  describing disjointness partitions over a group of roles, i.e, if  $\{U_i\}_{i=1}^h \in \text{REX}$  then exist  $R_i \in \mathcal{R}, C_i \in \mathcal{C}$  such  $U_i : C_i \in \text{REL}(R_i)$ , and all  $R_i$  have the same arity.

To link this syntax to ORM's icons, value types are transformed into attributes, any unary relationship is translated into a class and a binary relationship with a Boolean datatype, and a suitable naming scheme for the roles and fact types is in place.

**Example 1** Let us consider a mapping between this  $\mathcal{CM}_{com}^-$  syntax and some of the mappable ORM icons of the  $\text{ORM2}_{cta}$  fragment, using the diagram in Fig. 1. The fact type verbalised with **works for** can be represented with the relationship  $\text{REL}(\text{works}) = \{\text{DE} : \text{Department}, \text{ED} : \text{Employee}\}$  Identification (single attribute, resp. external) of the **Employee** and **Room** with  $\text{ID}(\text{Employee}) = \{\text{empNr}\}$  and  $\text{EXTID}(\text{Room}) = \{\{\text{locBuildingName}, \text{roomNr}\}\}$ . Subsumption of ORM entity types and fact types is straightforward as:

Academic  $\text{ISA}_C$  Employee and supports  $\text{ISA}_R$  works  
and mandatory participation of Department in works as  
 $\text{CARD}_R(\text{works}, \text{DE}) = (1, \infty)$ . ◇

### 3.2 Semantics

The model-theoretic semantics of  $\mathcal{CM}_{com}^-$  in the light of  $\text{ORM2}_{ctd}$  is as follows.

**Definition 3 ( $\mathcal{CM}_{com}^-$  Semantics)** Let  $\Sigma$  be a  $\mathcal{CM}_{com}^-$  conceptual data model. An *interpretation* for the conceptual model  $\Sigma$  is a tuple  $\mathcal{I} = (\Delta^{\mathcal{I}} \cup \Delta_{\mathcal{D}}^{\mathcal{I}}, \cdot^{\mathcal{I}})$ , such that:

- $\Delta^{\mathcal{I}}$  is a nonempty set of abstract objects disjoint from  $\Delta_{\mathcal{D}}^{\mathcal{I}}$ ;
- $\Delta_{\mathcal{D}}^{\mathcal{I}} = \bigcup_{D_i \in \mathcal{D}} \Delta_{D_i}$  is the set of basic domain values used in  $\Sigma$ ; and
- $\cdot^{\mathcal{I}}$  is a function that maps:
  - Every basic domain symbol  $D \in \mathcal{D}$  into a set  $D^{\mathcal{I}} = \Delta_D$ .
  - Every class  $C \in \mathcal{C}$  to a set  $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ .
  - Every attribute  $A \in \mathcal{A}$  to a set  $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathcal{D}}^{\mathcal{I}}$ , such that for each  $A \in \mathcal{A}$  and  $C \in \mathcal{C}$  with  $A : D \in \text{ATT}(C)$  for some  $D \in \mathcal{D}$ , then for all  $o \in C^{\mathcal{I}}$  there exists  $d \in D^{\mathcal{I}}$  such that  $(o, d) \in A^{\mathcal{I}}$ .
  - Every relationship  $R \in \mathcal{R}$  to a set  $R^{\mathcal{I}}$  of tuples of  $\mathcal{U}$ -labeled elements of  $\Delta^{\mathcal{I}}$ : for  $R$  an  $n$ -ary relationship connecting the classes  $C_1, \dots, C_n$ ,  $\text{REL}(R) = \{U_i : C_i\}_{i=1}^n$ , i.e.,  $\{U_i : o_i\}_{i=1}^n \in R^{\mathcal{I}}$  implies  $o_i \in C_i^{\mathcal{I}}$ .

$\mathcal{I}$  is called a *legal database state* or *legal application software state* if it satisfies all of the constraints expressed in the conceptual data model:

- for each  $C_1, C_2 \in \mathcal{C}$ : if  $C_1 \text{ ISA}_C C_2$ , then  $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ .
- for each  $R_1, R_2 \in \mathcal{R}$  (with the same signature): if  $R_1 \text{ ISA}_R R_2$ , then  $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$ .  
In addition, ORM stipulates that the participating role sequences (which in this case are all roles of  $R_i$ ) of every relationship participating in the  $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$  to be *role-sequence compatible*.<sup>6</sup>
- for each  $U_1, U_2 \in \mathcal{U}$ : if  $U_1 \text{ ISA}_U U_2$ , then there exist  $R_1, R_2 \in \mathcal{R}$  with  $\text{PLAYER}(R_1, U_1) = C_1$  and  $\text{PLAYER}(R_2, U_2) = C_1$ , and  $C_1, C_2 \in \mathcal{C}$ , and  $\{o \in C_1^{\mathcal{I}} \text{ such that } U_1 : o \in R_1^{\mathcal{I}}\} \subseteq \{o \in C_2^{\mathcal{I}} \text{ such that } U_2 : o \in R_2^{\mathcal{I}}\}$ .
- for each  $C \in \mathcal{C}, R \in \mathcal{R}, U \in \mathcal{U}$ : if  $\text{CARD}_R(R, U) = (x, y)$  and  $\text{PLAYER}(R, U) = C$ , then  $x \leq \#\{o \in C^{\mathcal{I}} \text{ such that } U : o \in R^{\mathcal{I}}\} \leq y$ .
- for each  $C \in \mathcal{C}, A \in \mathcal{A}$ : if  $\text{CARD}_A(C, A) = (x, y)$ , then  $A : D \in \text{ATT}(C)$  for some  $D \in \mathcal{D}$ , and  $x \leq \#\{d \in D^{\mathcal{I}} \text{ such that } (o, d) \in A^{\mathcal{I}}, o \in C^{\mathcal{I}}\} \leq y$ .
- if  $\{C_i\}_{i=1}^n \in \text{DISJ}_C$  then  $C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}} = \emptyset$  for all  $i, j, j \neq i, 1 \leq i, j \leq n$ .
- if  $\{R_i\}_{i=1}^n \in \text{DISJ}_R$  then  $R_i$  are role-sequence compatible and  $R_j^{\mathcal{I}} \cap R_k^{\mathcal{I}} = \emptyset$ .
- for each  $C \in \mathcal{C}, A_i \in \mathcal{A}, 1 \leq i \leq n$ : if  $\text{ID}(C) = \{A_i\}$ , then exists  $D_i \in \mathcal{D}$  such that  $A_i : D_i \in \text{ATT}(C)$ , and  $\#\{o \in C^{\mathcal{I}} \text{ such that } \bigwedge_{i=1}^n (o, d_i) \in A_i^{\mathcal{I}}\} \leq 1$  for any  $d_i \in D_i^{\mathcal{I}}, 1 \leq i \leq n$ .
- for each  $C \in \mathcal{C}, R_i \in \mathcal{R}, U_i \in \mathcal{U}, 1 \leq i \leq h, A_j, 1 \leq j \leq l$ : if  $\{(R_i, U_i)\}_{i=1}^h \cup \{A_j\}_{j=1}^l \in \text{EXTID}(C)$ , then exist  $D_j \in \mathcal{D}$  with  $A_j : D_j \in \text{ATT}(C)$ , and  $\#\{o \in C^{\mathcal{I}} \text{ such that } (U_i : o) \in R_i^{\mathcal{I}} \text{ and } (o, d_j) \in A_j^{\mathcal{I}}\} \leq 1$  for any  $d_j \in D_j^{\mathcal{I}}, 1 \leq j \leq l$ .

<sup>6</sup> We assume that the compatibility is enforced *explicitly* by additional  $\text{ISA}_C$  pairs of the classes linked to the matching roles in the relationships, e.g., for  $U : C \in \text{REL}(R_1)$  and  $U : D \in \text{REL}(R_2)$  we have  $C \text{ ISA}_C D$ .

- for each  $U, U_i \in \mathcal{U}, 1 \leq i \leq h, R \in \mathcal{R}$ : if  $\text{FD}(R) = (\{U_i\}_{i=1}^h, U)$ , then exists  $C, C_i \in \mathcal{C}$  such that  $C : U \in \text{REL}(R), C_i : U_i \in \text{REL}(R)$  and  $\#\{o \in C^{\mathcal{I}} \text{ such that } \{U : o\} \cup \{U_i : o_i, \} \in R^{\mathcal{I}}\} \leq 1$  for any  $o_i \in C_i^{\mathcal{I}}, 1 \leq i \leq h$ .
  - for each  $R \in \mathcal{R}, C \in \mathcal{C}$ : if  $\text{OBJ}(R) = C$ , then exist  $R_i \in \mathcal{R}$  with  $\text{REL}(R_i) = \{U_i^1 : C, U_i^2 : C_i\}$ , for all  $1 \leq i \leq n$ , and the following statements are also satisfied:  $\text{EXTID}(C) = \{(R_i, U_i^2) \mid 0 < i \leq n\}$ ,  $\text{CARD}_R(R_i, U_i^1) = (1, 1)$ , and  $\text{CARD}_R(R_i, U_i^2) = (0, 1)$ .
  - for each  $U_i \in \mathcal{U}, 1 \leq i \leq h$ : if  $\{U_i\}_{i=1}^h \in \text{REX}$ , then exists  $R_i \in \mathcal{R}$  with arity  $m$ , such that  $U_i : C_i \in \text{REL}(R_i)$ . Then for each  $o \in C_i^{\mathcal{I}}$  the following condition is true  $\#\{k \text{ such that } \{U_k : o\} \in R_k^{\mathcal{I}}, 1 \leq k \leq h\} \leq 1$ .
- $\Sigma$  is said to be *globally consistent* if it admits at least one legal state.

### 3.3 Mapping of ORM2<sub>ctd</sub> to $\mathcal{CFDT}_{nc}^{\forall-}$

We formalize how  $\mathcal{CFDT}_{nc}^{\forall-}$  can capture ORM2<sub>ctd</sub> conceptual models. Let  $\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}_R, \text{CARD}_A, \text{ISAC}, \text{ISAR}, \text{ISAU}, \text{DISJ}_C, \text{DISJ}_R, \text{ID}, \text{EXTID}, \text{FD}, \text{OBJ}, \text{REX})$  be a  $\mathcal{CM}_{com}^-$  conceptual data model corresponding to ORM2<sub>ctd</sub>. We map  $\Sigma$  to a  $\mathcal{CFDT}_{nc}^{\forall-}$  TBox  $\mathcal{T}_\Sigma$  in the vocabulary PC, F using the following rules:

- Include in the vocabulary one concept name for each ORM2<sub>ctd</sub> object type and datatype, i.e., for each  $C \in \mathcal{C}, D \in \mathcal{D}$  we have  $C \in \text{PC}, D \in \text{PC}$ .
- To map attributes, there are two cases to consider: if the attribute is functional then it is mapped as a function symbol and two concepts that reify each of the roles; otherwise it is reified as a new concept, with the corresponding cardinality constraints. For each  $C \in \mathcal{C}$  such that  $\text{ATT}(C) = \{A_i : D_i\}_{i=1}^n$ , and for each  $i$ :
  - if  $\text{CARD}_A(C, A_i) = (1, 1)$ , then we introduce two new concepts  $U_1^{A_i}, U_2^{A_i} \in \text{PC}$ , a new function symbol  $a_i \in \text{F}$ , and  $\{C \sqsubseteq \forall a_i.D_i, C \sqsubseteq U_1^{A_i}, U_1^{A_i} \sqsubseteq C, U_2^{A_i} \sqsubseteq \exists a_i^{-1}, \exists a_i^{-1} \sqsubseteq U_2^{A_i}\} \in \mathcal{T}_\Sigma$ .
  - otherwise we introduce new concept symbols  $A_i, U_{i,1}^A, U_{i,2}^A \in \text{PC}$ , and two new function symbols  $a_{i,1}, a_{i,2} \in \text{F}$ , with  $\{A_i \sqsubseteq \forall a_{i,1}.U_{i,1}^A, A_i \sqsubseteq \forall a_{i,2}.U_{i,2}^A, U_{i,1}^A \sqsubseteq C, U_{i,2}^A \sqsubseteq D_i, U_{i,1}^A \sqsubseteq \exists a_{i,1}^{-1}, U_{i,2}^A \sqsubseteq \exists a_{i,2}^{-1}, \forall a_{i,1}.U_{i,1}^A \sqsubseteq A_i, \forall a_{i,2}.U_{i,2}^A \sqsubseteq A_i, A_i \sqsubseteq A_i : a_{i,1}, a_{i,2} \rightarrow id\} \in \mathcal{T}_\Sigma$ . If  $\text{CMIN}(C, A_i) = 1$ , then also  $C \sqsubseteq U_{i,1}^A \in \mathcal{T}_\Sigma$ ; if  $\text{CMAX}(C, A_i) = 1$ , then  $A_i \sqsubseteq A_i : a_{i,1} \rightarrow id \in \mathcal{T}_\Sigma$ .
- The mapping of relationships (ORM2<sub>ctd</sub> fact types) is similar as the mapping of attributes. If the relationship is binary and one of its roles has a (1, 1) constraint, then it is mapped as an attribute; in the other cases the relationship and its roles are reified as a new concepts, with new attributes from the reified relationship to the reified roles. The reified roles are then subconcepts of the concepts originally participating in the relationship. For each  $R \in \mathcal{R}$  such that  $\text{REL}(R) = \{U_i : C_i\}_{i=1}^n$  then
  - if  $n = 2$  and  $\text{CARD}_R(R, U_1) = (1, 1)$ , then we introduce a new function symbol  $u_1 \in \text{F}$ , and  $\{C_1 \sqsubseteq \forall u_1.C_2, C_1 \sqsubseteq C_1 : u_1 \rightarrow id\} \in \mathcal{T}_\Sigma$ . Similarly if  $\text{CARD}_R(R, U_2) = (1, 1)$  then we introduce function symbol  $u_2 \in \text{F}$  and  $\{C_2 \sqsubseteq \forall u_2.C_1, C_2 \sqsubseteq C_2 : u_2 \rightarrow id\} \in \mathcal{T}_\Sigma$ .

- otherwise we add new concept symbols  $R, U_i^R \in \text{PC}, 1 \leq i \leq n$ , new function symbols  $u_i^R \in \text{F}, 1 \leq i \leq n$ , and then  $\{R \sqsubseteq \forall u_i^R. U_i^R, U_i^R \sqsubseteq C_i, U_i^R \sqsubseteq \exists u_i^{R^{-1}}, \forall u_i^R. U_i^R \sqsubseteq R\}_{i=1}^n \cup \{R : u_1^R, \dots, u_n^R \rightarrow id\} \subseteq \mathcal{T}_\Sigma$ . Additionally for each  $i, 1 \leq i \leq n$  if  $\text{CMIN}(R, U_i) = 1$  then also  $C_i \sqsubseteq U_i^R \in \mathcal{T}_\Sigma$ ; and if  $\text{CMAX}(R, U_i) = 1$ , then  $R \sqsubseteq R : u_i^R \rightarrow id \in \mathcal{T}_\Sigma$ .
- for each  $C_1, C_2 \in \mathcal{C}$  such that  $C_1 \text{ ISA}_C C_2$ , then  $C_1 \sqsubseteq C_2 \in \mathcal{T}_\Sigma$ .
- for each  $R_1, R_2 \in \mathcal{R}$  such that  $R_1 \text{ ISA}_R R_2$ , then  $R_1 \sqsubseteq R_2 \in \mathcal{T}_\Sigma$ . In order to define relationship inheritance in  $\text{ORM2}_{\text{cfd}}$ , the types of the participating concepts must be compatible, therewith adhering to the syntax restriction of ORM (as aside: without this condition, the reconstruction in a PTIME language is not possible).
- for each  $U_1, U_2 \in \mathcal{U}$  such that  $U_1 \text{ ISA}_U U_2$  then  $U_1 \sqsubseteq U_2 \in \mathcal{T}_\Sigma$ .
- for each  $C_i, C \in \mathcal{C}$  such that  $\{C_i\}_{i=1}^n \text{ DISJ}_C C$ , then  $\{C_i \sqsubseteq C\}_{i=1}^n \cup \{C_i \sqsubseteq \neg C_j\}_{i \neq j, i, j=1}^n \subseteq \mathcal{T}_\Sigma$ , stating the concepts are pairwise disjoint.
- for each  $R_i \in \mathcal{R}$  such that  $\{R_i\}_{i=1}^n \in \text{DISJ}_R$ , then  $\{R_i \sqsubseteq \neg R_j, R_i \sqsubseteq R_j : \sigma_{R_j} \rightarrow id\}_{i \neq j, i, j=1}^n \subseteq \mathcal{T}_\Sigma$ . Again, ORM has the condition that relationship exclusion must be defined over compatible types for the participating concepts (which also happens to be a necessary condition for the efficiency of the translation).
- for each  $C \in \mathcal{C}, A_i \in \mathcal{A}$  such that  $\text{ID}(C) = \{A_i\}_{i=1}^n$ , then  $C \sqsubseteq C : a_1, \dots, a_n \rightarrow id \in \mathcal{T}_\Sigma$ . In this case since the attributes are keys then they must be in (1, 1) constraint with  $C$ , so they are mapped as features in  $\mathcal{T}_\Sigma$  by the first point of the respective rule, described above.
- for each  $C \in \mathcal{C}, R_i \in \mathcal{R}, U_i \in \mathcal{U}, A_j \in \mathcal{A}$  such that  $\{(R_i, U_i)\}_{i=1}^h \cup \{A_j\}_{j=1}^k \in \text{EXTID}(C)$ , then  $C \sqsubseteq C : u_1^R, \dots, u_h^R, a_1, \dots, a_k \rightarrow id \in \mathcal{T}_\Sigma$ . Here the only possibility is that  $U_i$  and  $A_j$  belonging to the external identifier have a (1, 1) constraint, so they are mapped with features.
- for each  $U, U_i \in \mathcal{U}, R \in \mathcal{R}$ : if  $\text{FD}(R) = (\{U_i\}_{i=1}^h, U)$ , then  $C_R \sqsubseteq C_R : u_1^R, \dots, u_h^R \rightarrow u^R$ . This case is similar as the previous one: we ensure the roles and attributes belonging to the external identifier are mapped as features in  $\mathcal{T}_\Sigma$  because they are (1, 1) to  $C$ .
- for each  $R \in \mathcal{R}, C \in \mathcal{C}$ : if  $\text{OBJ}(R) = C$ , then we have the mappings for  $\text{EXTID}(C)$ ,  $\text{CARD}_R(R_i, U_i^1) = (1, 1)$ , and  $\text{CARD}_R(R_i, U_i^2) = (0, 1)$ .
- for each  $U_i \in \mathcal{U}$ : if  $\{U_i\}_{i=1}^h \in \text{REX}$ , then exists  $R_i \in \mathcal{R}, C_i \in \mathcal{C}$  with arity  $m$ , such that  $(U_i : C_i) \in \text{REL}(R_i), 1 \leq i \leq h$ . Then  $\{U_i^R \sqsubseteq \neg U_j^R\}_{i \neq j, i, j=1}^h \subseteq \mathcal{T}_\Sigma$ . Note that REX requires optional participation in the role, and therewith uses the second case of the relationship mapping above.

Case analysis of the translation combined with Proposition 1 yields the following:

**Theorem 1.** *Let  $\Sigma$  be an  $\text{ORM2}_{\text{cfd}}$  conceptual data model. A class  $C$  is consistent in  $\Sigma$  if and only if the knowledge base  $(\mathcal{T}_\Sigma, \{C(a_C)\})$  is consistent.  $\Sigma$  is globally consistent if and only if  $\mathcal{T}_\Sigma$  is consistent with  $\mathcal{A} = \{C(a_C) \mid C \in \mathcal{C}\}$ .*

**Example 2** Consider again the running example with Fig. 1 and the illustration of the syntax (Example 1), the corresponding  $\mathcal{CFDI}_{nc}^{\forall}$  knowledge base contains, among others: the translation where  $\text{empNr}$  is an attribute of  $\text{Employee}$  with

$\text{CARD}_A(\text{empNr}, \text{Employee}) = (1, 1)$ , the following  $\mathcal{CFDL}_{nc}^{\forall-}$  axioms:

$$\{\text{Employee} \sqsubseteq \forall \text{empNr}.\text{Integer}, \text{Employee} \sqsubseteq U_1^{\text{empNr}}, U_1^{\text{empNr}} \sqsubseteq \text{Employee}, \\ U_2^{\text{empNr}} \sqsubseteq \exists \text{empNr}^{-1}, \exists \text{empNr}^{-1} \sqsubseteq U_2^{\text{empNr}}\}.$$

Then, to represent the identifier, we add  $\text{Employee} \sqsubseteq \text{Employee} : \text{empNr} \rightarrow id$ . The mapping of  $\text{REL}(\text{works})$  with cardinalities  $\text{CARD}_R(\text{works}, \text{DE}) = (1, \infty)$  and  $\text{CARD}_R(\text{works}, \text{ED}) = (0, \infty)$  is the set of axioms

$$\left\{ \begin{array}{l} \text{Works} \sqsubseteq \forall \text{de}^{\text{works}}.\text{DE}^{\text{works}}, \text{DE}^{\text{works}} \sqsubseteq \text{Department}, \text{DE}^{\text{works}} \sqsubseteq \exists \text{de}^{\text{works}^{-1}}, \\ \forall \text{de}^{\text{works}}.\text{DE}^{\text{works}} \sqsubseteq \text{Works}, \text{Works} \sqsubseteq \text{Works} : \text{de}^{\text{works}}, \text{ed}^{\text{works}} \rightarrow id, \\ \text{Department} \sqsubseteq \text{DE}^{\text{works}}, \text{Works} \sqsubseteq \forall \text{ed}^{\text{works}}.\text{ED}^{\text{works}}, \text{ED}^{\text{works}} \sqsubseteq \text{Employee}, \\ \text{ED}^{\text{works}} \sqsubseteq \exists \text{ed}^{\text{works}^{-1}}, \forall \text{ed}^{\text{works}}.\text{ED}^{\text{works}} \sqsubseteq \text{Works} \end{array} \right\},$$

and likewise for the remainder of the diagram in Figure 1.  $\diamond$

## 4 Discussion

There are many papers with logic-based reconstructions of ORM, EER, and UML; we discuss a subset relevant to the scope of this paper.

**Comparison with other ORM2 Encodings.** Fairly expressive logic-based reconstructions of ORM fragments exist, including  $\text{ORM2}^{\text{zero}}$  in the EXPTIME-complete  $\mathcal{ALCQI}$  [20],  $\text{ORM2}^-$  [30] in the EXPTIME-complete  $\mathcal{DLR}_{ifd}$  [11], an ORM2 fragment (e.g., [41]) in  $\mathcal{SROIQ}$  that is N2EXPTIME-Complete [27], and ORM in the undecidable FOL [21]. An Alloy encoding and a numeric model as encoding for ORM are proposed in [23], which are experimentally compared to unsatisfiability pattern checks, showing that the latter two far outperform the Alloy approach (seconds vs. hours and timeouts), but complexity results are not provided. Their ORM fragment of the number encoding does include ‘costly’ features, such as covering constraints, disjunctive mandatory, arbitrary frequency (with uniqueness check), external identifiers, and value constraints, but it is unclear what was used in the test ORM diagrams. The only ORM fragment claimed in PTIME is Smaragdakis et al.’s  $\text{ORM}^-$  [35] that also uses a number model. It includes non-overlapping uniqueness constraints over  $n$ -ary relationships, simple mandatory, non-overlapping frequency constraints (cardinalities  $> 1$ ), value constraints, and subtype constraints. Arbitrary frequency constraints (like arbitrary projections in a relational table) cause undecidability, but, though not specified in [35], one could assume it always occurs in conjunction with a suitable uniqueness constraint in order to regain decidability, as discussed in [23, 29]. Their value constraints are not constrained either, i.e., value ranges of integers, floats, and enumerations are allowed, and have no constraints, such as so-called “safe” data types [3]. Most problematic, however, is the integer bound propagation in step 2 of Algorithm 2 in [35], which has recently shown to be NP-Complete [7], hence, Smaragdakis et al.’s solution seems to be at least NP-Hard. To the best of our knowledge, the here provided logic-based reconstruction of the  $\text{ORM2}_{cfd}$  ORM fragment in the PTIME  $\mathcal{CFDL}_{nc}^{\forall-}$  is the first tractable encoding of ORM, yet still capturing most of the entities encountered in extant ORM models.

**Extensibility to EER and UML Class Diagrams.** As ORM is more expressive than UML Class Diagrams and EER, and an ORM diagram can be translated into UML and into ER [22], the results obtained should be at least as good for those. A quick matching thanks to availing of the unifying metamodel [31, 18] reveals that that is the case, where  $\mathcal{CFDI}_{nc}^{\vee-}$  can encode over 97% of the 34 UML models and over 99% of the 34 (E)ER models in our dataset. This can be of use here as well, as also most UML Class Diagram encodings focus on expressiveness, using, among others,  $\mathcal{DLR}_{ifd}$  [5] as well, or an OCL-lite encoding matching  $\mathcal{ALCI}$  [34], which is still EXPTIME-complete. Kaneiwa and Satoh claim to have some fragments of UML Class Diagrams in P and PSpace for full satisfiability checking (all classes must be satisfiable) [25, 26], but this has been proven otherwise by Artale et al. [1]. Focusing on coverage of features, the smallest restricted fragment is shown to be NLogSpace [1] when disallowing ISA on associations and completeness on subclasses, using approximations of reified binaries (i.e., missing EXTID, and thus also no qualified associations). An initial analysis shows that this might still capture almost 96% of the UML models in our dataset, and might thus also be a worthwhile fragment of UML. Such high coverage can be obtained partially due to the changes to UML v2.4.1 [33] where relational properties (asymmetry and transitivity) for aggregation have been dropped, with aggregation taking up about a quarter of all associations, and not all UML features in the standard are implemented in modelling tools.

The story is similar for (E)ER. Various encodings exist [14, 37, 38], (partially due to the absence of a standard), which either use a language in the EXPTIME-complete  $\mathcal{DLR}$  family [9–11] or *DL-Lite* family [12] with different computational complexities for different EER fragments [2]. Trading functionality for gaining a little in computational complexity [2], however, is certainly not an option if coverage is also an aim, especially due to all the identifiers in EER and ORM (which can be represented in  $\mathcal{CFDI}_{nc}^{\vee-}$ ).

## 5 Conclusions

A logic-based reconstruction for most of ORM using the PTIME Description Logic language  $\mathcal{CFDI}_{nc}^{\vee-}$  has been presented, covering 96.5% of a set of extant ORM models. This is the first tractable encoding of ORM, which includes features important for conceptual models, notably  $n$ -ary relationships and complex identification constraints. Future work includes working toward implementations of the scenarios alluded to in the introduction, and we also expect to apply this encoding to facilitate inter-model interoperability [17] as the results are easily transferable to EER and UML Class diagrams.

**Acknowledgments.** This work is based upon research supported by the National Research Foundation of South Africa (Project UID: 90041), the Argentinian Ministry of Science and Technology (PRF and CMK), and NSERC (DT).

## References

1. Artale, A., Calvanese, D., Ibáñez-García, Y.: Full satisfiability of UML class diagrams. In: Parsons, J., et al. (eds.) Proceedings of the 29th International Conference on Conceptual Modeling (ER'10). LNCS, vol. 6412, pp. 317–331. Springer (2010)
2. Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyashev, M.: Reasoning over extended ER models. In: Parent, C., Schewe, K.D., Storey, V.C., Thalheim, B. (eds.) Proceedings of the 26th International Conference on Conceptual Modeling (ER'07). LNCS, vol. 4801, pp. 277–292. Springer (2007), auckland, New Zealand, November 5-9, 2007
3. Artale, A., Ryzhikov, V., Kontchakov, R.: DL-Lite with attributes and datatypes. In: Proceeding of the 20th European Conference on Artificial Intelligence (ECAI'12). pp. 61–66. IOS Press (2012)
4. Bakema, G., Zwart, J.P., van der Lek, H.: Volledig Communicatiegeoriënteerde Informatiemodellering FCO-IM. Academic Service (2005)
5. Berardi, D., Calvanese, D., De Giacomo, G.: Reasoning on UML class diagrams. *Artificial Intelligence* 168(1-2), 70–118 (2005)
6. Bloesch, A.C., Halpin, T.A.: Conceptual Queries using ConQuer-II. In: Proceedings of ER'97: 16th International Conference on Conceptual Modeling. LNCS, vol. 1331, pp. 113–126. Springer (1997)
7. Bordeaux, L., Katsirelos, G., Narodytska, N., Vardi, M.Y.: The complexity of integer bound propagation. *Journal of Artificial Intelligence Research* 40, 657–676 (2011)
8. Braga, B.F.B., Almeida, J.P.A., Guizzardi, G., Benevides, A.B.: Transforming OntoUML into Alloy: towards conceptual model validation using a lightweight formal methods. *Innovations in Systems and Software Engineering* 6(1-2), 55–63 (2010)
9. Calvanese, D., De Giacomo, G., Lenzerini, M.: On the decidability of query containment under constraints. In: Proc. of the 17th ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems (PODS'98). pp. 149–158 (1998)
10. Calvanese, D., De Giacomo, G., Lenzerini, M.: Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In: Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99). pp. 84–89 (1999)
11. Calvanese, D., De Giacomo, G., Lenzerini, M.: Identification constraints and functional dependencies in description logics. In: Nebel, B. (ed.) Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001). pp. 155–160. Morgan Kaufmann (2001), seattle, Washington, USA, August 4-10, 2001
12. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning* 39(3), 385–429 (2007)
13. Calvanese, D., Keet, C.M., Nutt, W., Rodríguez-Muro, M., Stefanoni, G.: Web-based graphical querying of databases through an ontology: the WONDER system. In: Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C.C. (eds.) Proceedings of ACM Symposium on Applied Computing (ACM SAC'10). pp. 1389–1396. ACM (2010), 22-26 March, 2010, Sierre, Switzerland
14. Chen, P.P.: The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems* 1(1), 9–36 (1976)
15. Curland, M., Halpin, T.: Model driven development with NORMA. In: Proceedings of the 40th International Conference on System Sciences (HICSS-40). pp. 286a–286a. IEEE Computer Society (2007), los Alamitos, Hawaii
16. Farré, C., Queralt, A., Rull, G., Teniente, E., Urpí, T.: Automated reasoning on UML conceptual schemas with derived information and queries. *Information and Software Technology* 55(9), 1529 – 1550 (2013)

17. Fillottrani, P.R., Keet, C.M.: Conceptual model interoperability: a metamodel-driven approach. In: Bikakis, A., et al. (eds.) Proceedings of the 8th International Web Rule Symposium (RuleML'14). LNCS, vol. 8620, pp. 52–66. Springer (2014), august 18-20, 2014, Prague, Czech Republic
18. Fillottrani, P.R., Keet, C.M.: KF metamodel formalisation. Technical report 1412.6545v1 (September 2014), arxiv.org
19. Fillottrani, P.R., Franconi, E., Tessaris, S.: The ICOM 3.0 intelligent conceptual modelling tool and methodology. Semantic Web Journal 3(3), 293–306 (2012)
20. Franconi, E., Mosca, A.: The formalisation of ORM2 and its encoding in OWL2. Technical Report KRDB12-2, KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy (March 2012), <http://www.inf.unibz.it/krdp/pub/TR/KRDB12-2.pdf>
21. Halpin, T.: A logical analysis of information systems: static aspects of the data-oriented perspective. Ph.D. thesis, University of Queensland, Australia (1989)
22. Halpin, T., Morgan, T.: Information modeling and relational databases. Morgan Kaufmann, 2nd edn. (2008)
23. Jahangard-Rafsanjani, A., Mirian-Hosseiniabadi, S.H.: Lightweight formalization and validation of ORM models. Journal of Logical and Algebraic Methods in Programming p. <http://dx.doi.org/10.1016/j.jlamp.2015.03.001> (2015)
24. Jahangard Rafsanjani, A., Mirian-Hosseiniabadi, S.H.: A Z approach to formalization and validation of ORM models. In: Ariwa, E., El-Qawasmeh, E. (eds.) Digital Enterprise and Information Systems, Communications in Computer and Information Science, vol. 194, pp. 513–526. Springer (2011)
25. Kaneiwa, K., Satoh, K.: Consistency checking algorithms for restricted UML class diagrams. In: Proceedings of the 4th International Symposium on Foundations of Information and Knowledge Systems (FoIKS'06). Springer Verlag (2006)
26. Kaneiwa, K., Satoh, K.: On the complexities of consistency checking for restricted UML class diagrams. Theoretical Computer Science 411, 301–323 (2010)
27. Kazakov, Y.: RIQ and SROIQ Are Harder than SHOIQ. In: Brewka, G., Lang, J. (eds.) 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08). pp. 274–284. AAAI Press (2008), 16-19 September, 2008, Sydney, Australia
28. Keet, C.M., Fillottrani, P.R.: Dataset of publicly available conceptual models and its analysis (2015), <http://www.meteck.org/SAAR.html>
29. Keet, C.M.: Prospects for and issues with mapping the Object-Role Modeling language into  $\mathcal{DLR}_{ifd}$ . In: 20th International Workshop on Description Logics (DL'07). CEUR-WS, vol. 250, pp. 331–338 (2007), 8-10 June 2007, Bressanone, Italy
30. Keet, C.M.: Ontology-driven formal conceptual data modeling for biological data analysis. In: Elloumi, M., Zomaya, A.Y. (eds.) Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data, chap. 6, pp. 129–154. Wiley (2013)
31. Keet, C.M., Fillottrani, P.R.: Toward an ontology-driven unifying metamodel for UML class diagrams, EER, and ORM2. In: Ng, W., Storey, V.C., Trujillo, J. (eds.) 32nd International Conference on Conceptual Modeling (ER'13). LNCS, vol. 8217, pp. 313–326. Springer (2013), 11-13 November, 2013, Hong Kong
32. Object Management Group: Semantics of Business Vocabulary and Rules (SBVR) – OMG released versions of SBVR, formal/2008-01-02 (January 2008), <http://www.omg.org/spec/SBVR/1.0>
33. Object Management Group: Superstructure specification. Standard 2.4.1, Object Management Group (2012), <http://www.omg.org/spec/UML/2.4.1/>

34. Queralt, A., Artale, A., Calvanese, D., Teniente, E.: OCL-Lite: Finite reasoning on UML/OCL conceptual schemas. *Data & Knowledge Engineering* 73, 1–22 (2012)
35. Smaragdakis, Y., Csallner, C., Subramanian, R.: Scalable satisfiability checking and test data generation from modeling diagrams. *Automation in Software Engineering* 16, 73–99 (2009)
36. Solomakhin, D., Franconi, E., Mosca, A.: Logic-based reasoning support for SBVR. *Fundamenta Informaticae* 124(4), 543–560 (2013)
37. Song, I.Y., Chen, P.P.: Entity relationship model. In: Liu, L., Özsu, M.T. (eds.) *Encyclopedia of Database Systems*, vol. 1, pp. 1003–1009. Springer (2009)
38. Thalheim, B.: Extended entity relationship model. In: Liu, L., Özsu, M.T. (eds.) *Encyclopedia of Database Systems*, vol. 1, pp. 1083–1091. Springer (2009)
39. Toman, D., Weddell, G.E.: *Fundamentals of Physical Design and Query Compilation*. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
40. Toman, D., Weddell, G.E.: On adding inverse features to the description logic  $CFD_{nc}^{\forall}$ . In: *PRICAI 2014: Trends in Artificial Intelligence - 13th Pacific Rim International Conference on Artificial Intelligence*, Gold Coast, QLD, Australia, December 1-5, 2014. pp. 587–599 (2014)
41. Wagih, H.M., Zanfaly, D.S.E., Kouta, M.M.: Mapping Object Role Modeling 2 schemes into  $SR\mathcal{OIQ}(D)$  Description Logic. *International Journal of Computer Theory and Engineering* 5(2), 232–237 (2013)