KF metamodel formalization

Pablo Rubén Fillottrani[1,2] and C. Maria Keet[3]

[1]Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina
[2]Comisión de Investigaciones Científicas, Provincia de Buenos Aires, Argentina
[3]Department of Computer Science, University of Cape Town, South Africa

Abstract

The KF metamodel [Keet and Fillottrani, 2013, Keet and Fillottrani, 2014] is a comprehensive unifying metamodel covering the static structural entities and constraints of UML Class Diagrams (v2.4.1), ER, EER, ORM, and ORM2, and intended to boost interoperability of common conceptual data modelling languages. It was originally designed in UML with textual constraints, and in this report we present its formalisations in FOL and OWL, which accompanies the paper that describes, discusses, and analyses the KF metamodel in detail. These new formalizations contribute to give a precise meaning to the metamodel, to understand its complexity properties and to provide a basis for future implementations.

1 Introduction

Interoperability of conceptual data modelling languages has become a necessity with ever complex software systems, be this through integration of 'legacy' systems or *de novo* development. In such cases, typically, one would use various conceptual models, such as ORM for usability with domain experts and interaction between requirements and data analysis, perhaps an EER model for the back-end database, and UML for any application layer software, which requires the system analyst to link entities across models represented in different conceptual modelling languages. To aid this endeavour, we have developed an ontology-driven unifying metamodel of UML v2.4.1, ER, EER, ORM and ORM2, whose static, structural entities have been presented in [Keet and Fillottrani, 2013], and an extended version also covering constraint in [Keet and Fillottrani, 2014]. More specifically, this concerned UML Class diagrams as specified in the UML Superstructure specification v2.4.1 [Object Management Group, 2012], typical ORM as described in [Committee Members, 2012, Halpin and Morgan, 2008], and both the original ash the encyclopaedic entries for ER and EER [Chen, 1976, Song and Chen, 2009, Thalheim, 2009].

This was modelled in UML Class Diagram notation with textual constraints for purpose of facilitating communication. Here we present the FOL formalisation of that metamodel to ensure precision of meaning and as a first step toward its computational use, including its necessity for designing efficient algorithms to computationally verify an inter-model assertion is correct (see also [Fillottrani and Keet, 2014]). Secondly, we have formalised it also in OWL 2 [Motik et al., 2009], for there are many tools that can process OWL files, hence opening up other avenues of the use of the metamodel, such as for categorising entities in extant models.

The remainder of this technical report presents first the FOL formalisation in Section 2 and subsequently discusses the modelling decisions for the OWL 2 DL version in Section 3. We close in Section 4.

2 FOL Formalization

The FOL formalization is organised along the subfigures of the metamodel, where we first present a brief overview (Section 2.1), and subsequently the relationships and attributes and the constraints among those entities. From Section 2.4 onwards, the model constraints are presented, starting with simple mandatory up to ORM's join constraints. Each subfigure of the metamodel is introduced first, which is followed by a brief description of the element in the figure and its formalisation. We use function-free FOL with equality, with a model-theoretic semantics; for a good overview of the language, see, e.g., [Hedman, 2004]. For notational convenience, we also use counting quantifiers $\exists^{\leq c}$, $\exists^{\geq c}$, and $\exists^{=c}$ for all $c > 0$ which do not add expressive power to FOL but facilitates the interpretation of formula. See [Baader et al., 2008] for their translation into traditional FOL.

2.1 Overview of the Static Entities

Figures 1 and 2 present the static entities of the metamodel, noting that it has been extended cf. [Keet and Fillottrani, 2013] with, notably, UML's qualifier, qualified association, and qualified identification. As before, we also use these two figures to communicate the overlap among the selected conceptual modelling languages, where a dark colour indicates all three families have that entity, classes filled with crossed lines are entities that appear in two of the three, a single line in only one, and an non-filled (white) class icon denotes an entity that appears in neither, but is used only in the metamodel to unify the other entities.
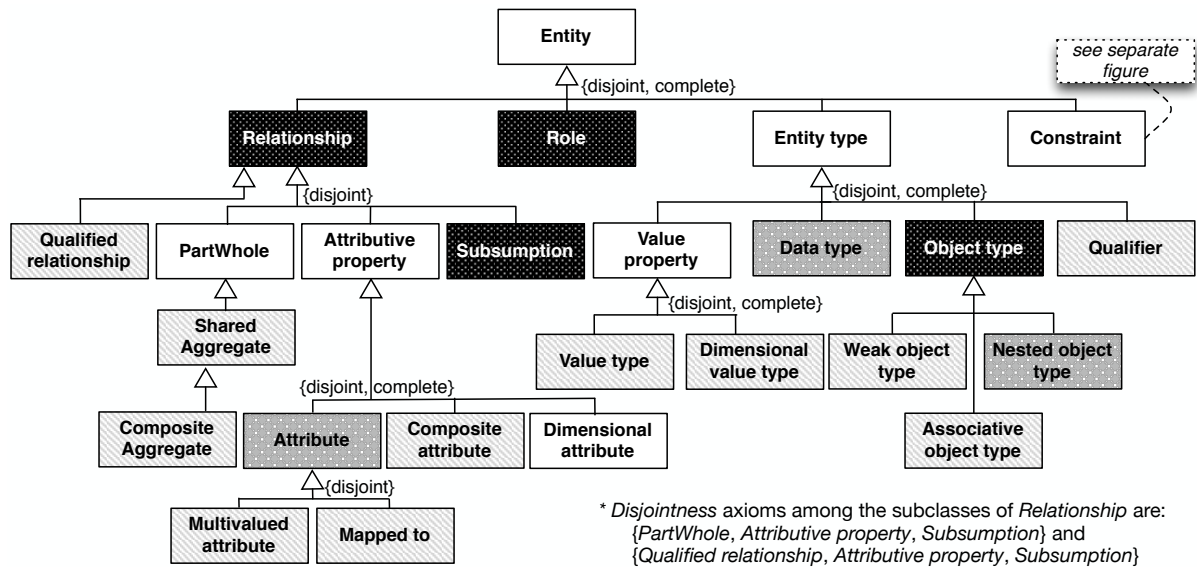

Figure 1: Principal static entities of the metamodel.

Formalization of Static Entities The formalization is described based on each UML class diagram element in the corresponding figure, from top to bottom and left to right, and after that the textual constraints. For example, in figure 1 we have 10 isa relations and one textual constraint.

- isa (disjoint and complete)

$\forall(x)(\texttt{Relationship}(x) \rightarrow \texttt{Entity}(x))$

$\forall(x)(\texttt{Role}(x) \rightarrow \texttt{Entity}(x))$

$\forall(x)(\texttt{EntityType}(x) \rightarrow \texttt{Entity}(x))$

$\forall(x)(\texttt{Constraint}(x) \rightarrow \texttt{Entity}(x))$

$\forall(x)(\neg(\texttt{Relationship}(x) \wedge \texttt{Role}(x)))$

$\forall(x)(\neg(\texttt{Relationship}(x) \wedge \texttt{EntityType}(x)))$

$\forall(x)(\neg(\texttt{Relationship}(x) \wedge \texttt{Constraint}(x)))$

$\forall(x)(\neg(\texttt{Role}(x) \wedge \texttt{EntityType}(x)))$

$\forall(x)(\neg(\texttt{Role}(x) \wedge \texttt{Constraint}(x)))$

$\forall(x)(\neg(\texttt{EntityType}(x) \wedge \texttt{Constraint}(x)))$

$\forall(x)(\texttt{Entity}(x) \rightarrow (\texttt{Relationship}(x) \vee \texttt{Role}(x) \vee \texttt{EntityType}(x) \vee \texttt{Constraint}(x)))$

- isa

  $\forall(x)(\texttt{QualifiedRelationship}(x) \rightarrow \texttt{Relationship}(x))$

- isa (disjoint)

  $\forall(x)(\texttt{PartWhole}(x) \rightarrow \texttt{Relationship}(x))$

  $\forall(x)(\texttt{AttributiveProperty}(x) \rightarrow \texttt{Relationship}(x))$

  $\forall(x)(\texttt{Subsumption}(x) \rightarrow \texttt{Relationship}(x))$

  $\forall(x)(\neg(\texttt{PartWhole}(x) \wedge \texttt{AttributiveProperty}(x)))$

  $\forall(x)(\neg(\texttt{PartWhole}(x) \wedge \texttt{Subsumption}(x)))$

  $\forall(x)(\neg(\texttt{AttributiveProperty}(x) \wedge \texttt{Subsumption}(x)))$

- isa

  $\forall(x)(\texttt{SharedAggregate}(x) \rightarrow \texttt{PartWhole}(x))$

- isa

  $\forall(x)(\texttt{CompositeAggregate}(x) \rightarrow \texttt{SharedAggregate}(x))$

- isa (disjoint, complete)

  $\forall(x)(\texttt{Attribute}(x) \rightarrow \texttt{AttributiveProperty}(x))$

  $\forall(x)(\texttt{CompositeAttribute}(x) \rightarrow \texttt{AttributiveProperty}(x))$

  $\forall(x)(\texttt{DimensionalAttribute}(x) \rightarrow \texttt{AttributiveProperty}(x))$

  $\forall(x)(\neg(\texttt{Attribute}(x) \wedge \texttt{CompositeAttribute}(x)))$

  $\forall(x)(\neg(\texttt{Attribute}(x) \wedge \texttt{DimensionalAttribute}(x)))$

  $\forall(x)(\neg(\texttt{CompositeAttribute}(x) \wedge \texttt{DimensionalAttribute}(x)))$

  $\forall(x)(\texttt{AttributiveProperty}(x) \rightarrow (\texttt{Attribute}(x) \vee \texttt{CompositeAttribute}(x) \vee$
  $\texttt{DimensionalAttribute}(x)))$

- isa (disjoint)

  $\forall(x)(\texttt{MultivaluedAttribute}(x) \rightarrow \texttt{Attribute}(x))$

  $\forall(x)(\texttt{MappedTo}(x) \rightarrow \texttt{Attribute}(x))$

  $\forall(x)(\neg(\texttt{MultivaluedAttribute}(x) \wedge \texttt{MappedTo}(x)))$

- isa (disjoint, complete)

$\forall(x)(\texttt{ValueProperty}(x) \rightarrow \texttt{EntityType}(x))$

$\forall(x)(\texttt{DataType}(x) \rightarrow \texttt{EntityType}(x))$

$\forall(x)(\texttt{ObjectType}(x) \rightarrow \texttt{EntityType}(x))$

$\forall(x)(\texttt{Qualifier}(x) \rightarrow \texttt{EntityType}(x))$

$\forall(x)(\neg(\texttt{ValueProperty}(x) \wedge \texttt{DataType}(x)))$

$\forall(x)(\neg(\texttt{ValueProperty}(x) \wedge \texttt{ObjectType}(x)))$

$\forall(x)(\neg(\texttt{ValueProperty}(x) \wedge \texttt{Qualifier}(x)))$

$\forall(x)(\neg(\texttt{DataType}(x) \wedge \texttt{ObjectType}(x)))$

$\forall(x)(\neg(\texttt{DataType}(x) \wedge \texttt{Qualifier}(x)))$

$\forall(x)(\neg(\texttt{ObjectType}(x) \wedge \texttt{Qualifier}(x)))$

$\forall(x)(\texttt{EntityType}(x) \rightarrow (\texttt{ValueProperty}(x) \vee \texttt{DataType}(x) \vee \texttt{ObjectType}(x) \vee \texttt{Qualifier}(x)))$

- isa (disjoint,complete)

  $\forall(x)(\texttt{ValueType}(x) \rightarrow \texttt{ValueProperty}(x))$

  $\forall(x)(\texttt{DimensionalValueType}(x) \rightarrow \texttt{ValueProperty}(x))$

  $\forall(x)(\neg(\texttt{ValueType}(x) \wedge \texttt{DimensionalValueType}(x)))$

  $\forall(x)(\texttt{ValueProperty}(x) \rightarrow (\texttt{ValueType}(x) \vee \texttt{DimensionalValueType}(x)))$

- isa

  $\forall(x)(\texttt{WeakObjectType}(x) \rightarrow \texttt{ObjectType}(x))$

  $\forall(x)(\texttt{NestedObjectType}(x) \rightarrow \texttt{ObjectType}(x))$

  $\forall(x)(\texttt{AssociativeObjectType}(x) \rightarrow \texttt{ObjectType}(x))$

- disjointness (first textual constraint)

  $\forall(x)(\neg(\texttt{QualifiedRelationship}(x) \wedge \texttt{AttributiveProperty}(x)))$

  $\forall(x)(\neg(\texttt{QualifiedRelationship}(x) \wedge \texttt{Subsumption}(x)))$

  $\forall(x)(\neg(\texttt{AttributiveProperty}(x) \wedge \texttt{Subsumption}(x)))$

2.2 Overview of the Constraints

There are several types of constraints in modelling languages, and this is reflected in the number of nodes in the graph in figure 2. The main is relation in the figue does not show all the siblings at the same level due to space limitations. In this section we only show the different types of constraints. In later sections, each constraint type is analyzed in detail together with related static entities.

Formalization of Constraint types

- isa

  $\forall(x)(\texttt{CardinalityConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{SubsetConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{JoinConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{UniquenessConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{RelationshipConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{DisjointnessConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{EqualityConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{ValueConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{CompletenessConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{ValueComparisonConstraint}(x) \rightarrow \texttt{Constraint}(x))$

  $\forall(x)(\texttt{IdentificationConstraint}(x) \rightarrow \texttt{Constraint}(x))$

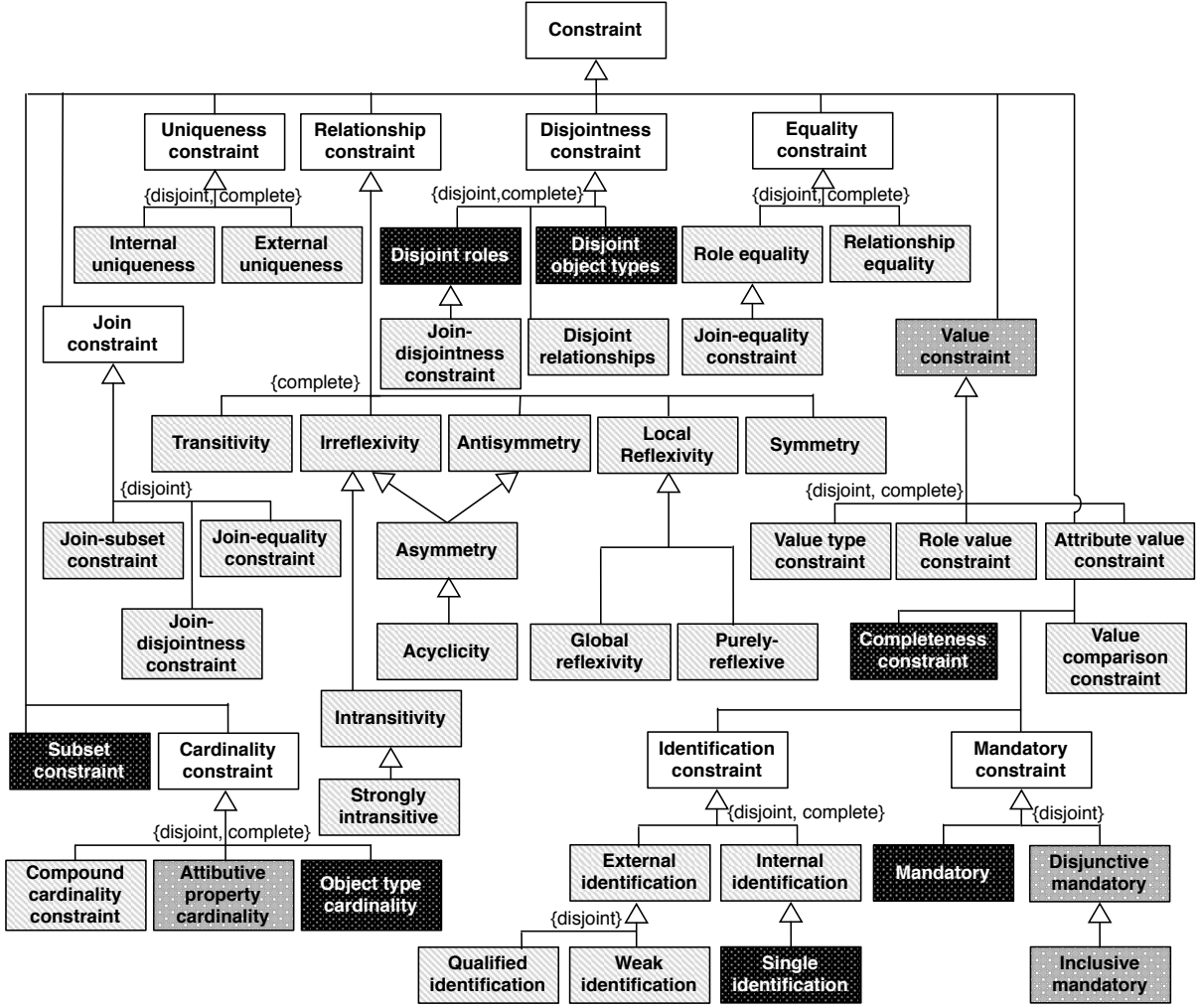 $\forall(x)(\texttt{MandatoryConstraint}(x) \rightarrow \texttt{Constraint}(x))$

Figure 2: Unified hierarchy of constraints in the metamodel.

- isa (disjoint, complete)

  $\forall(x)(\text{CompoundCardinalityConstraint}(x) \rightarrow \text{CardinalityConstraint}(x))$

  $\forall(x)(\text{AttributivePropertyCardinality}(x) \rightarrow \text{CardinalityConstraint}(x))$

  $\forall(x)(\text{ObjectTypeCardinality}(x) \rightarrow \text{CardinalityConstraint}(x))$

  $\forall(x)(\neg(\text{CompoundCardinalityConstraint}(x) \wedge \text{AttributivePropertyCardinality}(x)))$

  $\forall(x)(\neg(\text{CompoundCardinalityConstraint}(x) \wedge \text{ObjectTypeCardinality}(x)))$

  $\forall(x)(\neg(\text{AttributivePropertyCardinality}(x) \wedge \text{ObjectTypeCardinality}(x)))$

  $\forall(x)(\text{CardinalityConstraint}(x) \rightarrow (\text{CompoundCardinalityConstraint}(x) \vee$
  $\qquad \text{AttributivePropertyCardinality}(x) \vee \text{ObjectTypeCardinality}(x)))$

- isa (disjoint)

  $\forall(x)(\text{JoinSubsetConstraint}(x) \rightarrow \text{JoinConstraint}(x))$

  $\forall(x)(\text{JoinEqualityConstraint}(x) \rightarrow \text{JoinConstraint}(x))$

  $\forall(x)(\text{JoinDisjointnessConstraint}(x) \rightarrow \text{JoinConstraint}(x))$

  $\forall(x)(\neg(\text{JoinSubsetConstraint}(x) \wedge \text{JoinEqualityConstraint}(x)))$

  $\forall(x)(\neg(\text{JoinSubsetConstraint}(x) \wedge \text{JoinDisjointnessConstraint}(x)))$

  $\forall(x)(\neg(\text{JoinEqualityConstraint}(x) \wedge \text{JoinDisjointnessConstraint}(x)))$

- isa (disjoint, complete)

$\forall(x)(\text{InternalUniquenessConstraint}(x) \rightarrow \text{UniquenessConstraint}(x))$

$\forall(x)(\text{ExternalUniquenessConstraint}(x) \rightarrow \text{UniquenessConstraint}(x))$

$\forall(x)(\neg(\text{InternalUniquenessConstraint}(x) \wedge \text{ExternalUniquenessConstraint}(x)))$

$\forall(x)(\text{UniquenessConstraint}(x) \rightarrow (\text{InternalUniquenessConstraint}(x) \vee \\ \text{ExternalUniquenessConstraint}(x)))$

- isa (complete)

  $\forall(x)(\text{Transitivity}(x) \rightarrow \text{RelationshipConstraint}(x))$

  $\forall(x)(\text{Antisymmetry}(x) \rightarrow \text{RelationshipConstraint}(x))$

  $\forall(x)(\text{Irreflexivity}(x) \rightarrow \text{RelationshipConstraint}(x))$

  $\forall(x)(\text{LocalReflexivity}(x) \rightarrow \text{RelationshipConstraint}(x))$

  $\forall(x)(\text{Symmetry}(x) \rightarrow \text{RelationshipConstraint}(x))$

  $\forall(x)(\text{RelationshipConstraint}(x) \rightarrow (\text{Transitivity}(x) \vee \text{Antisymmetry}(x) \vee \\ \text{Irreflexivity}(x) \vee \text{LocalReflexivity} \vee \text{Symmetry}(x)))$

- isa

  $\forall(x)(\text{Asymmetry}(x) \rightarrow \text{Antisymmetry}(x))$

- isa

  $\forall(x)(\text{Asymmetry}(x) \rightarrow \text{Irreflexivity}(x))$

- isa

  $\forall(x)(\text{Acyclicity}(x) \rightarrow \text{Asymmetry}(x))$

- isa

  $\forall(x)(\text{Intransitivity}(x) \rightarrow \text{Irreflexivity}(x))$

- isa

  $\forall(x)(\text{StronglyIntransitivity}(x) \rightarrow \text{Intransitivity}(x))$

- isa

  $\forall(x)(\text{GlobalReflexivity}(x) \rightarrow \text{LocalReflexivity}(x))$

- isa

  $\forall(x)(\text{PurelyReflexivity}(x) \rightarrow \text{LocalReflexivity}(x))$

- isa (disjoint, complete)

  $\forall(x)(\text{DisjointRoles}(x) \rightarrow \text{DisjointnessConstraint}(x))$

  $\forall(x)(\text{DisjointRelationships}(x) \rightarrow \text{DisjointnessConstraint}(x))$

  $\forall(x)(\text{DisjointObjectTypes}(x) \rightarrow \text{DisjointnessConstraint}(x))$

  $\forall(x)(\neg(\text{DisjointRoles}(x) \wedge \text{DisjointObjectTypes}(x)))$

  $\forall(x)(\neg(\text{DisjointRoles}(x) \wedge \text{DisjointRelationships}(x)))$

  $\forall(x)(\neg(\text{DisjointRelationships}(x) \wedge \text{DisjointObjectTypes}(x)))$

  $\forall(x)(\text{DisjointnessConstraint}(x) \rightarrow (\text{DisjointRoles}(x) \vee \\ \text{DisjointObjectTypes}(x) \vee \text{DisjointRelationships}(x)))$

- isa

  $\forall(x)(\text{JoinDisjointnessConstraint}(x) \rightarrow \text{DisjointRoles}(x))$

- isa (disjoint, complete)

  $\forall(x)(\text{RoleEquality}(x) \rightarrow \text{EqualityConstraint}(x))$

  $\forall(x)(\text{RelationshipEquality}(x) \rightarrow \text{EqualityConstraint}(x))$

  $\forall(x)(\neg(\text{RoleEquality}(x) \wedge \text{RelationshipEquality}(x)))$

  $\forall(x)(\text{EqualityConstraint}(x) \rightarrow (\text{RoleEquality}(x) \vee \text{RelationshipEquality}(x)))$

- isa

  $\forall(x)(\text{JoinEqualityConstraint}(x) \rightarrow \text{RoleEquality}(x))$

- isa (disjoint, complete)

  $\forall(x)(\text{ValueTypeConstraint}(x) \rightarrow \text{ValueConstraint}(x))$
  $\forall(x)(\text{RoleValueConstraint}(x) \rightarrow \text{ValueConstraint}(x))$
  $\forall(x)(\text{AttributeValueConstraint}(x) \rightarrow \text{ValueConstraint}(x))$
  $\forall(x)(\neg(\text{ValueTypeConstraint}(x) \wedge \text{RoleValueConstraint}(x)))$
  $\forall(x)(\neg(\text{ValueTypeConstraint}(x) \wedge \text{AttributeValueConstraint}(x)))$
  $\forall(x)(\neg(\text{RoleValueConstraint}(x) \wedge \text{AttributeValueConstraint}(x)))$
  $\forall(x)(\text{ValueConstraint}(x) \rightarrow (\text{ValueTypeConstraint}(x) \vee$
  $\qquad \text{RoleValueConstraint}(x) \vee \text{AttributeValueConstraint}(x)))$

- isa (disjoint, complete)

  $\forall(x)(\text{ExternalIdentification}(x) \rightarrow \text{IdentificationConstraint}(x))$
  $\forall(x)(\text{InternalIdentification}(x) \rightarrow \text{IdentificationConstraint}(x))$
  $\forall(x)(\neg(\text{ExternalIdentification}(x) \wedge \text{InternalIdentification}(x)))$
  $\forall(x)(\text{IdentificationConstraint}(x) \rightarrow (\text{ExternalIdentification}(x) \vee$
  $\qquad \text{InternalIdentification}(x)))$

- isa (disjoint)

  $\forall(x)(\text{QualifiedIdentification}(x) \rightarrow \text{ExternalIdentification}(x))$
  $\forall(x)(\text{WeakIdentification}(x) \rightarrow \text{ExternalIdentification}(x))$
  $\forall(x)(\neg(\text{QualifiedIdentification}(x) \wedge \text{WeakIdentification}(x)))$

- isa

  $\forall(x)(\text{SingleIdentification}(x) \rightarrow \text{InternalIdentification}(x))$

- isa (disjoint)

  $\forall(x)(\text{Mandatory}(x) \rightarrow \text{MandatoryConstraint}(x))$
  $\forall(x)(\text{DisjunctiveMandatory}(x) \rightarrow \text{MandatoryConstraint}(x))$
  $\forall(x)(\neg(\text{MandatoryConstraint}(x) \wedge \text{DisjunctiveMandatory}(x)))$

- isa

  $\forall(x)(\text{InclusiveMandatory}(x) \rightarrow \text{DisjunctiveMandatory}(x))$

2.3 Relationships between Relationship, Role and Entity type

This section goes into detail on relationships and roles in general, and subsequently subsumption and aggregation, and attributes and value types. The main entities and constraints pertaining to relationships are depicted in Figure 3.

Formalization of Relationship, Role, EntityType and related constraints

- composition $(1, 2..*)$

  $\forall(x, y)(\text{Contains}(x, y) \rightarrow \text{Relationship}(x) \wedge \text{Role}(y))$
  $\forall(x)\exists^{\geq 2}y(\text{Contains}(x, y))$
  $\forall(x)(\text{Role}(x) \rightarrow \exists^{=1}(y)(\text{Contains}(y, x)))$

  We do not need to add asymmetric and irreflexive axioms to the formalization because they are both implicit after the disjointness between `Relationship` and `Role`. Also, transitivity is not included since we are dealing with a "direct parthood" relationship.
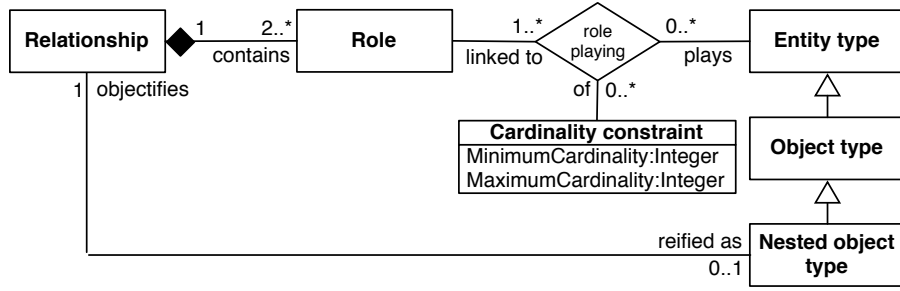
Figure 3: Relationships between Relationship, Role, and Entity type; see text for details.

- ternary relation $(1..*, 0..*, 0..*)$

  $\forall(x, y, z)(\text{RolePlaying}(x, y, z) \rightarrow (\text{Role}(x) \wedge \text{CardinalityConstraint}(y) \wedge \text{EntityType}(z)))$

  $\forall(x)(\text{Role}(x) \rightarrow \exists^{\geq 1}(y, z)(\text{RolePlaying}(x, y, z)))$

  There are several possible interpretations of cardinality constraints over non-binary relations on UML class diagrams. In this formalization we interpret all such constraints as in [Artale et al., 2007], constraining the quantity of tuples in which each entity may participate. This choice is reflected in the formula.

- attributes

  $\forall(x, y)((\text{CardinalityConstraint}(x) \wedge \text{MinimumCardinality}(x, y)) \rightarrow \text{Integer}(y))$

  $\forall(x, y)((\text{CardinalityConstraint}(x) \wedge \text{MaximumCardinality}(x, y)) \rightarrow \text{Integer}(y))$

  $\forall(x)(\text{CardinalityConstraint}(x) \rightarrow \exists^{\leq 1}(y)(\text{MinimumCardinality}(x, y)))$

  $\forall(x)(\text{CardinalityConstraint}(x) \rightarrow \exists^{\leq 1}(y)(\text{MaximumCardinality}(x, y)))$

  Attributes in UML classes may also have cardinality constraints. In this figure, we understand each attribute to have [0..1] cardinality. The type Integer is assumed to be a range of integer numbers from 0 to a maximum number noted n. This last number is interpreted to mean "any integer value".

- relation $(1, 0..1)$

  $\forall(x, y)(\text{ReifiedAs}(x, y) \rightarrow \text{Relationship}(x) \wedge \text{NestedObjectType}(y))$

  $\forall(x)(\text{NestedObjectType}(x) \rightarrow \exists^{=1}(y)(\text{ReifiedAs}(x, y)))$

  $\forall(x(\text{Relationship}(x) \rightarrow \exists^{\leq 1}(y)(\text{ReifiedAs}(x, y)))$

- relation between ReifiedAs, Contains and RolePlaying (not in figure)

  $\forall(x, y)(\text{ReifiedAs}(x, y) \rightarrow \forall(z, w)(\text{Contains}(x, z) \leftrightarrow \text{RolePlaying}(z, w, y)))^*$

  This constraint has more than two variables and cannot be replaced with equivalent formula in two variables. All the formula with this property are shown with a * at then end. This star means the formula has to be taken out the formalization in order to show its decidability. See section 2.11 for more on the properties of the formalization.

2.3.1 Subsumption and Aggregation

The metamodel fragment dealing with subsumption and aggregation is depicted in Figure 4 and has the following logic-based reconstruction.

Formalization of Subsumption and Aggregation

- relation $(0..*, 2)$, subset and union constraints
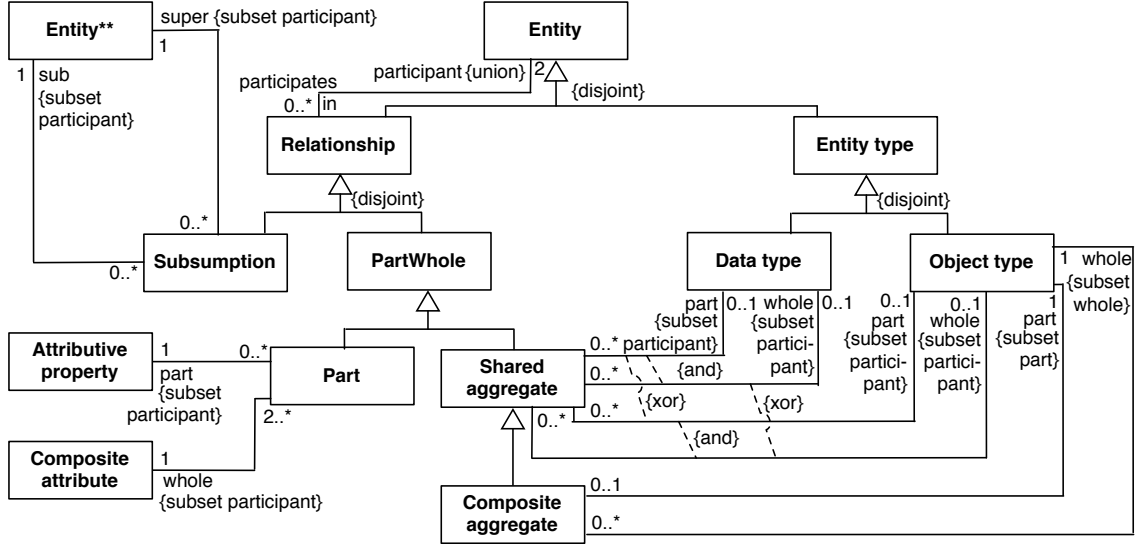
* The two *Entities* participating in a *sub* and *super* in a *Subsumption* must be compatible (of the same category).
* *Shared aggregate* must participate at least once in a *part* and at least once in a *whole*.
** *Entity* excluding its subtypes *Qualified relationship*, *Attributive property*, *Subsumption*, *Qualifier*, and *Constraint*.

Figure 4: Subsumption and aggregation.

$$\forall(x,y)(\texttt{Participant}(x,y) \rightarrow ((\texttt{Relationship}(x) \land \texttt{Entity}(y)) \lor$$
$$(\texttt{ValueComparisonConstraint}(x) \land \texttt{Role}(y)) \lor$$
$$(\texttt{RelationshipConstraint}(x) \land \texttt{Role}(y))))$$
$$\forall(x)(\texttt{Relationship}(x) \rightarrow \exists^{=2}(y)(\texttt{Participant}(x,y)))$$
$$\forall(x,y)((\texttt{Participant}(x,y) \land \texttt{Relationship}(x)) \leftrightarrow (\texttt{Sub}(x,y) \lor$$
$$\texttt{Super}(x,y) \lor \texttt{Part}(x,y) \lor \texttt{Whole}(x,y)))$$

- relation $(1, 0..*)$, third textual constraint

$$\forall(x,y)(\texttt{Sub}(x,y) \rightarrow (\texttt{Subsumption}(x) \land \texttt{Entity}(y) \land$$
$$\neg\texttt{QualifiedRelationship}(y) \land \neg\texttt{AttributiveProperty}(y) \land$$
$$\neg\texttt{Subsumption}(y) \land \neg\texttt{Qualifier}(y) \land \neg\neg\texttt{Constraint}(y)))$$
$$\forall(x)(\texttt{Subsumption}(x) \rightarrow \exists^{=1}(y)(\texttt{Sub}(x,y)))$$

- relation $(1, 0..*)$, third textual constraint

$$\forall(x,y)(\texttt{Super}(x,y) \rightarrow (\texttt{Subsumption}(x) \land \texttt{Entity}(y) \land$$
$$\neg\texttt{QualifiedRelationship}(y) \land \neg\texttt{AttributiveProperty}(y) \land$$
$$\neg\texttt{Subsumption}(y) \land \neg\texttt{Qualifier}(y) \land \neg\texttt{Constraint}(y)))$$
$$\forall(x)(\texttt{Subsumption}(x) \rightarrow \exists^{=1}(y)(\texttt{Super}(x,y)))$$

- disjunctive exclusion (not in figure)

$$\forall(x,y)((\texttt{Subsumption}(x) \land \texttt{Sub}(x,y) \land \Phi(y)) \rightarrow \exists(z)(\Phi(z) \land \texttt{Super}(x,z)))^*$$
$$\forall(x,y)((\texttt{Subsumption}(x) \land \texttt{Super}(x,y) \land \Phi(y)) \rightarrow \exists(z)(\Phi(z) \land \texttt{Sub}(x,z)))^*$$

For all $\Phi \in \{\texttt{SharedAggregate}, \texttt{CompositeAggregate}, \texttt{Role}, \texttt{DataType},$
$\texttt{Relationship} \land \neg\texttt{PartWhole}, \texttt{ObjectType}\}$.

- isa

$$\forall(x,y)(\texttt{Part}(x) \rightarrow \texttt{PartWhole}(x))$$

- relation $(0..*, 1)$

9

$$\forall(x,y)(\texttt{Part}(x,y) \to ((\texttt{Part}(x) \wedge \texttt{AttributiveProperty}(y)) \vee$$
$$(\texttt{SharedAggregate}(x) \wedge \texttt{DataType}(y)) \vee$$
$$(\texttt{SharedAggregate}(x) \wedge \texttt{ObjectType}(y))))$$
$$\forall(x)(\texttt{Part}(x) \to \exists^{=1}(y)(\texttt{Part}(x,y) \wedge \texttt{AttributiveProperty}(y)))$$

- relation $(2..*, 1)$

$$\forall(x,y)(\texttt{Whole}(x,y) \to ((\texttt{Part}(x) \wedge \texttt{CompositeAttribute}(y)) \vee$$
$$(\texttt{SharedAggregate}(x) \wedge \texttt{DataType}(y)) \vee$$
$$(\texttt{SharedAggregate}(x) \wedge \texttt{ObjectType}(y))))$$
$$\forall(x)(\texttt{Part}(x) \to \exists^{=1}(y)(\texttt{Whole}(x,y) \wedge \texttt{CompositeAttribute}(y)))$$
$$\forall(x)(\texttt{CompositeAttribute}(x) \to \exists^{\geq 2}(y)(\texttt{Part}(y) \wedge \texttt{Whole}(y,x)))$$

- relation $(0..*, 0..1)$, xor, second textual constraint

$$\forall(x)(\texttt{SharedAggregate}(x) \to \exists^{\leq 1}(y)((\texttt{DataType}(y) \vee \texttt{ObjectType}(y)) \wedge \texttt{Part}(x,y)))$$
$$\forall(x)(\texttt{SharedAggregate}(x) \to (\exists^{\geq 1}(y)(\texttt{Part}(x,y)) \wedge \exists^{\geq 1}(y)(\texttt{Whole}(x,y))))$$

- relation $(0..*, 0..1)$, xor

$$\forall(x)(\texttt{SharedAggregate}(x) \to \exists^{\leq 1}(y)((\texttt{DataType}(y) \vee \texttt{ObjectType}(y)) \wedge \texttt{Whole}(x,y)))$$

The last two set of formula only show the disjunction of the xor. The exclusion between `DataType` and `ObjectType` is inherited from the disjoint isa in figure 1.

- and relationships

$$\forall(x,y)((\texttt{SharedAggregate}(x) \wedge \texttt{DataType}(y) \wedge \texttt{Part}(x,y)) \to$$
$$\exists(z)(\texttt{DataType}(z) \wedge \texttt{Whole}(x,z)))^*$$
$$\forall(x,y)((\texttt{SharedAggregate}(x) \wedge \texttt{DataType}(y) \wedge \texttt{Whole}(x,y)) \to$$
$$\exists(z)(\texttt{DataType}(z) \wedge \texttt{Part}(x,z)))^*$$
$$\forall(x,y)((\texttt{SharedAggregate}(x) \wedge \texttt{ObjectType}(y) \wedge \texttt{Part}(x,y)) \to$$
$$\exists(z)(\texttt{ObjectType}(z) \wedge \texttt{Whole}(x,z)))^*$$
$$\forall(x,y)((\texttt{SharedAggregate}(x) \wedge \texttt{ObjectType}(y) \wedge \texttt{Whole}(x,y)) \to$$
$$\exists(z)(\texttt{ObjectType}(z) \wedge \texttt{Part}(x,z)))^*$$

- irreflexivity, asymmetry (not in figure)

$$\forall(x,y,z)((\texttt{PartWhole}(x) \wedge \texttt{Part}(x,y) \wedge \texttt{Whole}(x,z)) \to \neg(y=z))^*$$
$$\forall(x,y,z)((\texttt{PartWhole}(x) \wedge \texttt{Part}(x,y) \wedge \texttt{Whole}(x,z)) \to$$
$$\neg\exists(v)(\texttt{PartWhole}(v) \wedge \texttt{Part}(v,z) \wedge \texttt{Whole}(v,y)))^*$$

- relation $(0..1, 1)$

$$\forall(x,y)((\texttt{Part}(x,y) \wedge \texttt{CompositeAggregate}(x)) \to \texttt{ObjectType}(y))$$
$$\forall(x)(\texttt{CompositeAggregate}(x) \to \exists^{=1}(y)(\texttt{Part}(x,y)))$$
$$\forall(x)(\texttt{CompositeAggregate}(x) \to \exists^{\leq 1}(y)(\texttt{Part}(y,x)))$$

- relation $(0..*, 1)$

$$\forall(x,y)((\texttt{CompositeAggregate}(x) \wedge \texttt{Whole}(x,y)) \to \texttt{ObjectType}(y))$$
$$\forall(x)(\texttt{CompositeAggregate}(x) \to \exists^{=1}(y)(\texttt{Whole}(x,y)))$$

- first textual constraint

$$\forall(x, y)(\mathrm{Compatible}(x, y) \rightarrow$$
$$((\mathrm{ValueProperty}(x) \wedge \mathrm{ValueProperty}(y)) \vee$$
$$(\mathrm{DataType}(x) \wedge \mathrm{DataType}(y)) \vee$$
$$(\mathrm{ObjectType}(x) \wedge \mathrm{ObjectType}(y)) \vee$$
$$(\mathrm{Role}(x) \wedge \mathrm{Role}(y)) \vee$$
$$(\mathrm{Relationship}(x) \wedge \mathrm{Relationship}(y))))$$
$$\forall(x, y)((\mathrm{Compatible}(x, y) \wedge \mathrm{Role}(x)) \rightarrow \exists(v, w, s, t)(\mathrm{RolePlaying}(x, v, w) \wedge$$
$$\mathrm{RolePlaying}(y, s, t) \wedge \mathrm{Compatible}(w, t)))^{*}$$
$$\forall(x, y)((\mathrm{Compatible}(x, y) \wedge \mathrm{Relationship}(x)) \rightarrow$$
$$((\exists^{=n}(z)(\mathrm{Contains}(x, z)) \leftrightarrow \exists^{=n}(z)(\mathrm{Contains}(y, z))) \wedge$$
$$(\exists(z, v)(\mathrm{Contains}(x, z) \wedge \mathrm{Contains}(y, w) \wedge \mathrm{Compatible}(z, w)))))^{*}$$
$$\forall(x, y, z)((\mathrm{Subsumption}(x) \wedge \mathrm{Sub}(x, y) \wedge \mathrm{Super}(x, z)) \rightarrow \mathrm{Compatible}(y, z))^{*}$$

Here n is any natural number from 2 up to the maximum arity of a relationship in the model. Therefore the formula is not second order.

2.3.2 Attributes and Value Types

The metamodel fragment concerning attributes and ORM's counterpart, value types, is shown Figure 5 and has the following logic-based reconstruction.
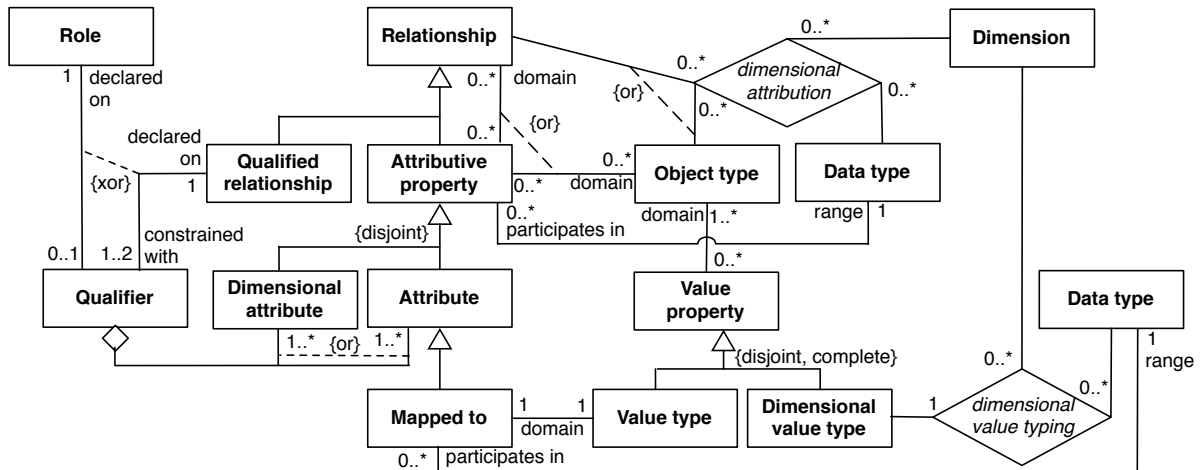

Figure 5: Metamodel fragment for value properties and simple attributes; Dimensional attribute is reified version of the ternary relation dimensional attribution, and likewise for Dimensional value type and dimensional value typing.

Formalization of Attributes and Value Types

- relation $(0..1, 1)$, relation $(1..2, 1)$, xor

$\forall(x,y)(\texttt{DeclaredOn}(x,y) \rightarrow ((\texttt{Qualifier}(x) \land \texttt{Role}(y)) \lor$
$\qquad (\texttt{Qualifier}(x) \land \texttt{QualifiedRelationship}(y)) \lor$
$\qquad (\texttt{MandatoryConstraint}(x) \land \texttt{Role}(y)) \lor$
$\qquad (\texttt{InternalUniquenessConstraint}(x) \land \texttt{Role}(y)) \lor$
$\qquad (\texttt{ExternalIdentification}(x) \land \texttt{Relationship}(y)) \lor$
$\qquad (\texttt{IdentificationConstraint}(x) \land \texttt{ValueProperty}(y)) \lor$
$\qquad (\texttt{IdentificationConstraint}(x) \land \texttt{AttributiveProperty}(y)) \lor$
$\qquad (\texttt{RelationshipEquality}(x) \land \texttt{Relationship}(y)) \lor$
$\qquad (\texttt{DisjointRelationships}(x) \land \texttt{Relationship}(y)) \lor$
$\qquad (\texttt{RoleEquality}(x) \land \texttt{Role}(y)) \lor$
$\qquad (\texttt{DisjointRoles}(x) \land \texttt{Role}(y)) \lor$
$\qquad (\texttt{DisjointEntities}(x) \land \texttt{Subsumption}(y)) \lor$
$\qquad (\texttt{ObjectTypeEquality}(x) \land \texttt{ObjectType}(y)) \lor$
$\qquad (\texttt{CompletenessConstraint}(x) \land \texttt{Subsumption}(y))))$

$\forall(x)(\texttt{Qualifier}(x) \rightarrow \exists^{=1}(y)(\texttt{DeclaredOn}(x,y) \land (\texttt{Role}(y) \lor \texttt{QualifiedRelationship}(y))))$

$\forall(x)(\texttt{Role}(x) \rightarrow \exists^{\leq 1}(y)(\texttt{Qualifier}(y) \land \texttt{DeclaredOn}(y,x)))$

$\forall(x)(\texttt{QualifiedRelationship}(x) \rightarrow (\exists^{\geq 1}(y)(\texttt{Qualifier}(y) \land \texttt{DeclaredOn}(y,x)) \land$
$\qquad \exists^{\leq 2}(y)(\texttt{Qualifier}(y) \land \texttt{DeclaredOn}(y,x))))$

$\forall(x,y,z)((\texttt{DeclaredOn}(x,y) \land \texttt{DeclaredOn}(x,z) \land \texttt{Qualifier}(x) \land \texttt{Role}(y)) \rightarrow \texttt{Role}(z)))^*$

$\forall(x,y,z)((\texttt{DeclaredOn}(x,y) \land \texttt{DeclaredOn}(x,z) \land \texttt{Qualifier}(x) \land$
$\qquad \texttt{QualifiedRelationship}(y)) \rightarrow \texttt{QualifiedRelationship}(z)))^*$

- composition $(0..*, 1..*)$, or

  $\forall(x,y)(\texttt{HasComponent}(x,y) \rightarrow (\texttt{Qualifier}(x) \land (\texttt{DimensionalAttribute}(y) \lor \texttt{Attribute}(y))))$

  $\forall(x)(\texttt{Qualifier}(x) \rightarrow \exists^{\geq 1}(y)(\texttt{HasComponent}(x,y)))$

- relation $(0..*, 0..*)$

  $\forall(x,y)(\texttt{Domain}(x,y) \rightarrow ((\texttt{AttributiveProperty}(x) \land \texttt{Relationship}(y)) \lor$
  $\qquad (\texttt{AttributiveProperty}(x) \land \texttt{ObjectType}(y)) \lor (\texttt{MappedTo}(x) \land \texttt{ValueType}(y)) \lor$
  $\qquad (\texttt{ValueProperty}(x) \land \texttt{ObjectType}(y))))$

- relation $(0..*, 1)$

  $\forall(x,y)(\texttt{Range}(x,y) \rightarrow ((\texttt{AttributiveProperty}(x) \land \texttt{DataType}(y)) \lor$
  $\qquad (\texttt{MappedTo}(x) \land \texttt{DataType}(y))))$

  $\forall(x)(\texttt{AttributiveProperty}(x) \rightarrow \exists^{=1}(y)(\texttt{DataType}(y) \land \texttt{Range}(x,y)))$

- relation $(0..*, 1..*)$

  $\forall(x)(\texttt{ValueProperty}(x) \rightarrow \exists^{\geq 1}(y)(\texttt{Domain}(x,y) \land \texttt{ObjectType}(y)))$

- relation $(0..*, 1)$

  $\forall(x)(\texttt{MappedTo}(x) \rightarrow \exists^{=1}(y)(\texttt{DataType}(y) \land \texttt{Range}(x,y)))$

- ternary relation $(0..*, 0..*, 0..*)$, or

  $\forall(x,y,z)(\texttt{DimensionalAttribution}(x,y,z) \rightarrow (\texttt{Dimension}(x) \land \texttt{DataType}(y) \land$
  $\qquad (\texttt{ObjectType}(z) \lor \texttt{Relationship}(z))))$

  Although this formula has three variables, it can be replaced by an equivalent set of several formula in two variables, thus it is not starred. This translation is similar to the process of reification described in Section 3, and it is applied to all ternary relations.

- ternary relation $(0..*, 0..*, 1)$

  $\forall(x,y,z)(\texttt{DimensionalValueTyping}(x,y,z) \rightarrow (\texttt{Dimension}(x) \land$
  $\qquad \texttt{DataType}(y) \land \texttt{DimensionalValueType}(z)))$

  $\forall(x)(\texttt{DimensionalValueType}(x) \rightarrow \exists^{=1}(y,z)(\texttt{DimensionalValueTyping}(y,z,x)))$

2.4 Mandatory Constraints

Form this section onwards, we turn our focus to constraints that can be represented on the selected conceptual modelling languages. The first one is about the different mandatory constraints, as shown in Figure 6.
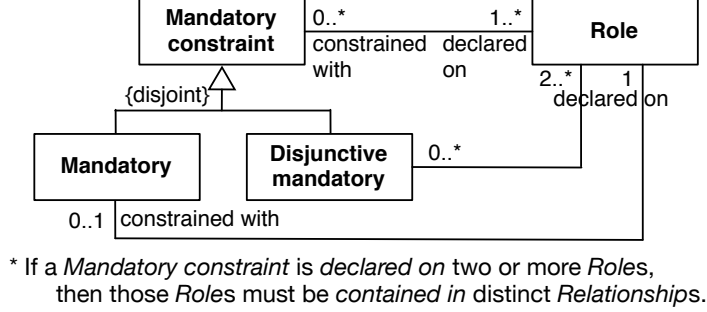

* If a *Mandatory constraint* is *declared on* two or more *Role*s,
  then those *Role*s must be *contained in* distinct *Relationship*s.

Figure 6: Mandatory constraints.

Formalization of Mandatory Constraints

- relation $(0..*, 1..*)$

  $\forall(x)(\mathtt{MandatoryConstraint}(x) \rightarrow \exists^{\geq 1}(y)(\mathtt{Role}(y) \wedge \mathtt{DeclaredOn}(x, y)))$

- relation $(0..*, 2..*)$

  $\forall(x)(\mathtt{DisjunctiveMandatory}(x) \rightarrow \exists^{\geq 2}(y)(\mathtt{Role}(y) \wedge \mathtt{DeclaredOn}(x, y)))$

- relation $(0..1, 1)$

  $\forall(x)(\mathtt{Mandatory}(x) \rightarrow \exists^{=1}(y)(\mathtt{Role}(y) \wedge \mathtt{DeclaredOn}(x, y)))$

  $\forall(x)(\mathtt{Role}(x) \rightarrow \exists^{\leq 1}(y)(\mathtt{DeclaredOn}(y, x) \wedge \mathtt{Mandatory}(y)))$

- constraint on relationships containing roles with the same mandatory constraint, not in diagram

  $\forall(x, y, z, v, w)((\mathtt{DeclaredOn}(x, y) \wedge \mathtt{DeclaredOn}(x, z) \wedge \mathtt{MandatoryConstraint}(x) \wedge$
  $\mathtt{Role}(y) \wedge \mathtt{Role}(z) \wedge \mathtt{Contains}(w, y) \wedge \mathtt{Contains}(v, z) \wedge$
  $\mathtt{Relationship}(v) \wedge \mathtt{Relationship}(w)) \rightarrow \neg(w = v))^*$

2.5 Uniqueness Constraints

The second core constraints in the languages is uniqueness, as shown in Figure 7, which is formalised as follows.

Formalization of Uniqueness Constraints

- relation $(0..*, 1..*)$

  $\forall(x)(\mathtt{InternalUniquenessConstraint}(x) \rightarrow \exists^{\geq 1}(y)(\mathtt{Role}(y) \wedge \mathtt{DeclaredOn}(x, y)))$

- ternary relation $(1, 0; 2..*, 0; 2..*)$, third textual constraint

  $\forall(x, y, z)(\mathtt{ExtUnique}(x, y, z) \rightarrow (\mathtt{ExternalUniquenessConstraint}(x) \wedge$
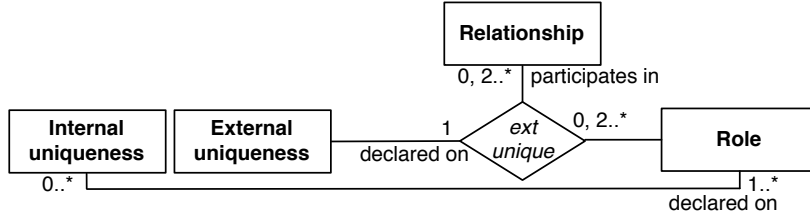 $\mathtt{Role}(y) \wedge \mathtt{Relationship}(z) \wedge \mathtt{Contains}(z, y)))^*$

  $\forall(x)(\mathtt{ExternalUniquenessConstraint}(x) \rightarrow \exists^{=1}(y, z)(\mathtt{ExtUnique}(x, y, z)))$

  $\forall(x)(\mathtt{Role}(x) \rightarrow (\exists^{=0}(y, z)(\mathtt{ExternalUnique}(y, x, z)) \vee$
  $\exists^{\geq 2}(y, z)(\mathtt{ExternalUnique}(y, x, z))))$

  $\forall(x)(\mathtt{Relationship}(x) \rightarrow (\exists^{=0}(y, z)(\mathtt{ExternalUnique}(y, z, x)) \vee$
  $\exists^{\geq 2}(y, z)(\mathtt{ExternalUnique}(y, z, x))))$

* The cardinality of *Internal uniqueness* being *declared on* at least one *Role*
has a maximum cardinality less than or equal the number of *Role*s in
the *Relationship* that *contains* those *Role*s.
* The cardinality of *Relationship* and *Role* in *ext unique* is equal.
* The *Role*s on which an *External uniqueness* is *declared on* are *contained in* the
Relationships that *participate in* the *ext unique*.

Figure 7: Uniqueness constraints.

- first textual constraint

$$\forall(x,y,z)((\texttt{InternalUniquenessConstraint}(x) \wedge \texttt{Role}(y) \wedge \texttt{DeclaredOn}(x,y) \wedge$$
$$\texttt{Relationship}(z) \wedge \texttt{Contains}(z,y)) \rightarrow$$
$$(\exists^{=n}(w)(\texttt{Contains}(z,w)) \leftrightarrow \exists^{\leq n}(v)(\texttt{DeclaredOn}(x,v))))^{*}$$

This is necessary first order since n is any natural number up to the maximum arity of
any relationships in the model.

- second textual constraint

$$\forall(x)(\texttt{ExternalUniquenessConstraint}(x) \rightarrow (\exists^{=n}(y)\exists(z)(\texttt{ExternalUnique}(x,y,z)$$
$$\leftrightarrow \exists^{=n}(z)\exists(y)\texttt{ExternalUnique}(x,y,z))))^{*}$$

This is necessary first order since n is any natural number up to the maximum arity of a
relationships in the model.

2.6 Identification Constraints

The metamodel fragment concerning identifiers (Figure 8), while comprehensive in its treatment,
does not deal with definitions of identity (see [Keet, 2011] for a brief discussion on this topic).
How identification is dealt with in the languages is formalised as follows.

Formalization of Identification Constraints

- relation $(0..*, 1)$

$$\forall(x,y)(\texttt{PartiallyIdentifies}(x,y) \rightarrow$$
$$(\texttt{QualifiedIdentification}(x) \wedge \texttt{QualifiedRelationship}(y)))$$
$$\forall(x)(\texttt{QualifiedIdentification}(x) \rightarrow \exists^{=1}(y)(\texttt{PartiallyIdentifies}(x,y)))$$

- relation $(0..*, 1..*)$

$$\forall(x)(\texttt{ExternalIdentification}(x) \rightarrow \exists^{\geq 1}(y)(\texttt{DeclaredOn}(x,y))$$
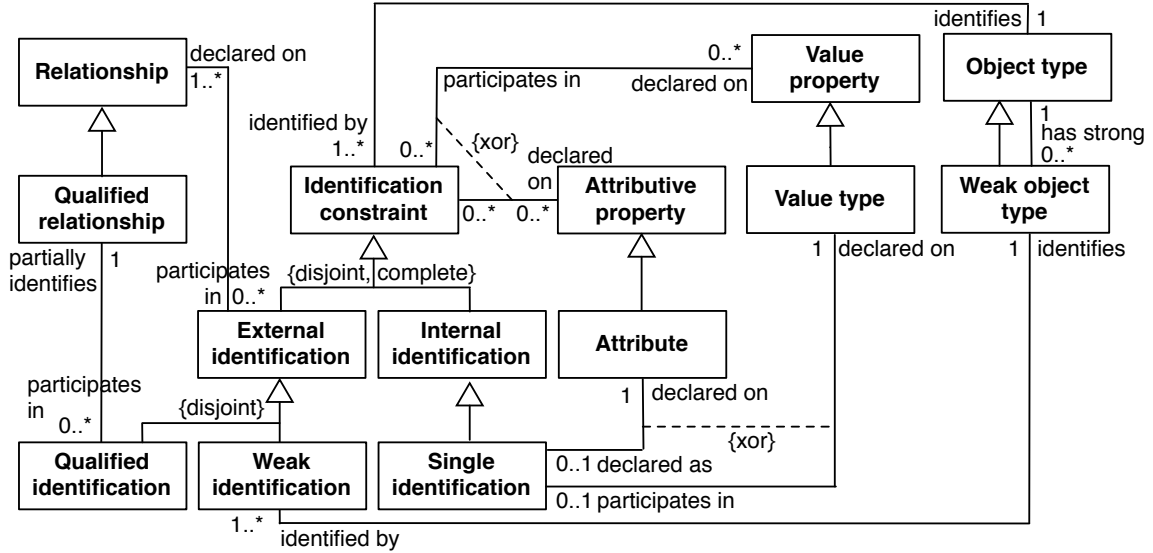
- relation $(1..*, 1)$

$$\forall(x,y)(\texttt{Identifies}(x,y) \rightarrow (\texttt{IdentificationConstraint}(x) \wedge \texttt{ObjectType}(y)))$$
$$\forall(x)(\texttt{IdentificationConstraint}(x) \rightarrow \exists^{=1}(y)(\texttt{Identifies}(x,y)))$$
$$\forall(x)(\texttt{ObjectType}(x) \rightarrow \exists^{\geq 1}(y)(\texttt{Identifies}(y,x)))$$

- relation $(0..*, 0..*)$, xor, mandatory (not in diagram)

Figure 8: Partial representation of identification constraints.

$\forall(x, y, z)((\texttt{DeclaredOn}(x, y) \wedge \texttt{DeclaredOn}(x, z) \wedge$
$\qquad \texttt{IdentificationConstraint}(x) \wedge \texttt{ValueProperty}(y)) \rightarrow$
$\qquad \texttt{ValueProperty}(z))^*$

$\forall(x, y, z)((\texttt{DeclaredOn}(x, y) \wedge \texttt{DeclaredOn}(x, z) \wedge$
$\qquad \texttt{IdentificationConstraint}(x) \wedge \texttt{AttributiveProperty}(y)) \rightarrow$
$\qquad \texttt{AttributiveProperty}(z))^*$

$\forall(x)(\texttt{IdentificationConstraint}(x) \rightarrow \exists(y)(\texttt{DeclaredOn}(x, y)))$

- relation $(0..1, 1)$

$\forall(x, y)((\texttt{DeclaredOn}(x, y) \wedge \texttt{SingleIdentification}(x)) \rightarrow (\texttt{Attribute}(y) \vee \texttt{ValueType}(y)))$
$\forall(x)(\texttt{SingleIdentification}(x) \rightarrow \exists^{=1}(y)(\texttt{DeclaredOn}(x, y))$
$\forall(x)((\texttt{Attribute}(x) \vee \texttt{ValueType}(x)) \rightarrow \exists^{\leq 1}(y)(\texttt{DeclaredOn}(y, x) \wedge$
$\qquad \texttt{SingleIdentification}(y)))$

Xor is inherited from previous set of formula.

- relation $(1..*, 1)$

$\forall(x, y)((\texttt{Identifies}(x, y) \wedge \texttt{WeakIdentification}(x)) \rightarrow \texttt{WeakObjectType}(y))$
$\forall(x)(\texttt{WeakIdentification}(x) \rightarrow \exists^{=1}(y)(\texttt{Identifies}(x, y)))$
$\forall(x)(\texttt{WeakObjectType}(x) \rightarrow \exists^{\geq 1}(y)(\texttt{WeakIdentification}(y) \wedge$
$\qquad \texttt{Identifies}(y, x)))$

- relation $(0..*, 1)$

$\forall(x, y)(\texttt{HasStrong}(x, y) \rightarrow (\texttt{WeakObjectType}(x) \wedge \texttt{ObjectType}(y)))$
$\forall(x)(\texttt{WeakObjectType}(x) \rightarrow \exists^{=1}(y)(\texttt{HasStrong}(x, y)))$

- first textual constraint

$$\forall(x, y, z)((\texttt{WeakIdentification}(x) \wedge \texttt{Identifies}(x, y) \wedge \texttt{DeclaredOn}(x, z)) \rightarrow$$
$$((\texttt{AttributiveProperty}(z) \wedge \texttt{Domain}(z, y)) \vee$$
$$\exists^{=1}(v, w, s)(\texttt{IdentificationConstraint}(v) \wedge \texttt{Identifies}(v, w) \wedge$$
$$\texttt{Relationship}(s) \wedge \texttt{DeclaredOn}(v, z) \wedge \texttt{Participant}(s, w) \wedge \texttt{Participant}(s, y))))^*$$
$$\forall(x, y, z, v, w, s, t)((\texttt{WeakIdentification}(x) \wedge \texttt{Identifies}(x, y) \wedge \texttt{DeclaredOn}(x, s) \wedge$$
$$\texttt{IdentificationConstraint}(v, w) \wedge \texttt{Identifies}(v, t) \wedge$$
$$\texttt{DeclaredOn}(v, s)) \rightarrow \neg(y = t))^*$$

- second textual constraint

$$\forall(x, y, z)((\texttt{SingleIdentification}(x) \wedge \texttt{Identifies}(x, y) \wedge \texttt{DeclaredOn}(x, z)) \rightarrow$$
$$\exists(v)(\texttt{CardO}(z, y, v) \wedge \texttt{MinimumCardinality}(v, 1) \wedge \texttt{MaximumCardinality}(v, 1)))^*$$

- third textual constraint

$$\forall(x, y)((\texttt{QualifiedIdentification}(x) \wedge \texttt{DeclaredOn}(x, y)) \rightarrow \texttt{AttributiveProperty}(y))$$
$$\forall(x, y)((\texttt{ExternalIdentification}(x) \wedge \texttt{DeclaredOn}(x, y)) \rightarrow \texttt{AttributiveProperty}(y))$$

- fourth textual constraint

$$\forall(\texttt{x}, \texttt{y})(\texttt{PartiallyIdentifies}(\texttt{x}, \texttt{y}) \rightarrow \exists(z, v, w)(\texttt{RolePlaying}(z, v, w) \wedge$$
$$\texttt{Contains}(y, z) \wedge \texttt{MinimumCardinality}(v, 1) \wedge \texttt{MaximumCardinality}(v, 1)))^*$$

2.7 Cardinality Constraints

The fourth important constraint of the languages are the cardinality constraints; see Figure 9, which is formalised below.
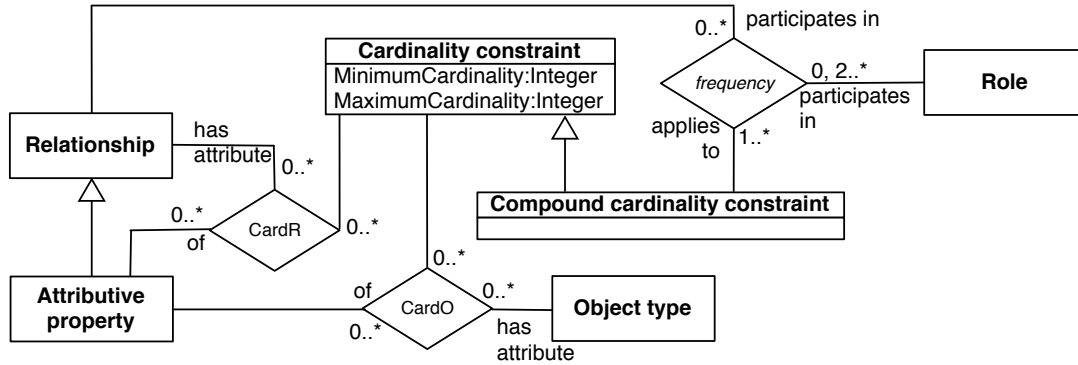
* If there is a *Cardinality constraint*,
    then it must include a *MinimumCardinality* or *MaximumCardinality* or both.

* Each *Compound cardinality constraint* must include a *MinimumCardinality* or *MaximumCardinality* or both.
* For each *Compound cardinality constraint* over 2 or more *Roles*,
    those *Roles* must be *contained in* the same *Relationship* and are distinct.

Figure 9: Cardinality constraints on attributes of object types and relationships, and compound cardinality over more than one role in a relationship.

Formalization of Cardinality Constraints

- ternary relationship $(0..*, 0..*, 0..*)$

$$\forall(x, y, z)(\texttt{CardR}(x, y, z) \rightarrow (\texttt{AttributiveProperty}(x) \wedge \texttt{Relationship}(y) \wedge$$
$$\texttt{CardinalityConstraint}(z)))$$

- ternay relation $(0..*, 0..*, 0..*)$

$$\forall(x, y, z)(\texttt{CardO}(x, y, z) \rightarrow (\texttt{AttributiveProperty}(x) \wedge \texttt{ObjectType}(y) \wedge$$
$$\texttt{CardinalityConstraint}(z)))$$

- ternary relation $(0..*, 0; 2..*, 1)$

$\forall(x,y,z)(\texttt{Frequency}(x,y,z) \to (\texttt{Relationship}(x) \wedge \texttt{Role}(y) \wedge$
$\qquad \texttt{CompoundCardinalityConstraint}(z)))$
$\forall(x)(\texttt{CompoundCardinalityConstraint}(x) \to \exists^{=1}(y,z)(\texttt{Frequency}(y,z,x))$
$\forall(x)(\texttt{Role}(x) \to (\exists^{=0}(y,z)(\texttt{Frequency}(y,x,z)) \vee$
$\qquad \exists^{\geq 2}(y,z)(\texttt{Frequency}(y,x,z))))$

- first and second textual constraint

$\forall(x)(\texttt{CardinalityConstraint}(x) \to \exists(y)(\texttt{MaximumCardinality}(x,y) \vee$
$\qquad \texttt{MinimumCardinality}(x,y)))$

The other part of these constraints are implicit from the formalization of attributes. The second constraint is implicit from the inheritance.

- isa, third textual constraint

$\forall(x)(\texttt{CompoundCardinalityConstraint}(x) \to \texttt{CardinalityConstraint}(x))$
$\forall(x,y,z,v,w)((\texttt{Frequency}(x,y,z) \wedge \texttt{Frequency}(v,w,z)) \to$
$\qquad ((x=v) \wedge \texttt{Contains}(x,y) \wedge \texttt{Contains}(v,w) \wedge \neg(y=w)))^*$

2.8 Value type, Role and Attribute Value constraints

The formalisation of value types and role and attribute constraints is rather elegant compared to the difficulty of drawing that metamodel fragment (Figure 10), for there are repeating components to it that can be more easily captured in FOL.
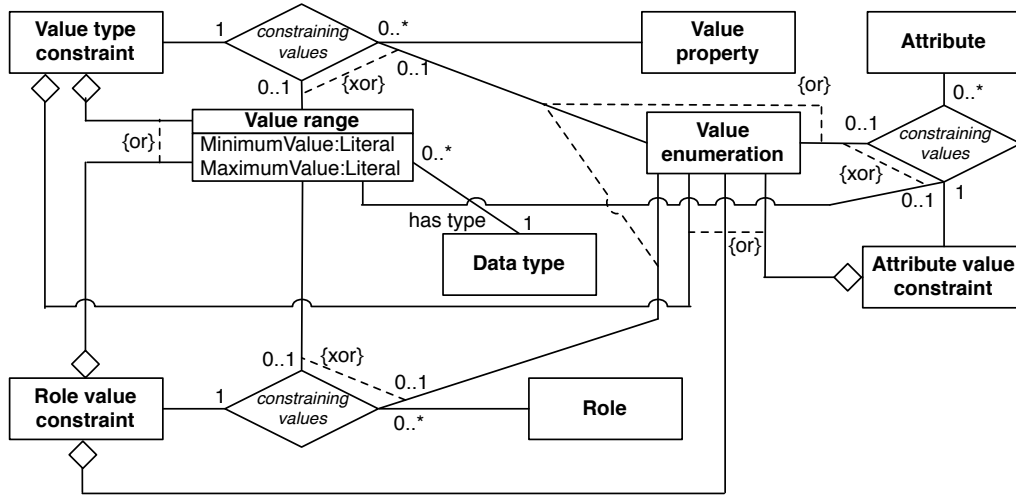

Figure 10: Value type, role, and attribute value constraints.

Formalization of Value type, Role and Attribute Value Constraints

- ternary relations $(1, 0..*, 0..*)$, xor

$\forall(x, w, z)(\texttt{ConstrainingValues}(x, y, z) \rightarrow (((\texttt{ValueTypeConstraint}(x) \wedge$
$\qquad \texttt{ValueProperty}(y)) \vee (\texttt{RoleValueConstraint}(x) \wedge \texttt{Role}(y)) \vee$
$\qquad (\texttt{AttributeValueConstraint}(x) \wedge \texttt{Attribute}(y))) \wedge$
$\qquad (\texttt{ValueEnumeration}(z) \vee \texttt{ValueRange}(z))))$

$\forall(x)(\texttt{ValueTypeConstraint}(x) \rightarrow \exists^{=1}(y, z)(\texttt{ConstrainingValues}(x, y, z)))$

$\forall(x)(\texttt{RoleValueConstraint}(x) \rightarrow \exists^{=1}(y, z)(\texttt{ConstrainingValues}(x, y, z)))$

$\forall(x)(\texttt{AttributeValueConstraint}(x) \rightarrow \exists^{=1}(y, z)(\texttt{ConstrainingValues}(x, y, z)))$

$\forall(x)(\texttt{ValueEnumeration}(x) \rightarrow \neg\texttt{ValueRange}(x))$

$\forall(x)(\texttt{ValueRange}(x) \rightarrow \neg\texttt{ValueEnumeration}(x))$

- relation $(0..*, 1)$

  $\forall(x, y)(\texttt{HasType}(x, y) \rightarrow (\texttt{ValueRange}(x) \wedge \texttt{DataType}(y)))$

  $\forall(x)(\texttt{ValueRange}(x) \rightarrow \exists^{=1}(y)(\texttt{HasType}(x, y)))$

- attributes

  $\forall(x, y)((\texttt{ValueRange}(x) \wedge \texttt{MinimumValue}(x, y)) \rightarrow \texttt{Literal}(y))$

  $\forall(x, y)((\texttt{ValueRange}(x) \wedge \texttt{MaximumValue}(x, y)) \rightarrow \texttt{Literal}(y))$

  $\forall(x)(\texttt{ValueRange}(x) \rightarrow \exists^{=1}(y)(\texttt{MinimumValue}(x, y)))$

  $\forall(x)(\texttt{ValueRange}(x) \rightarrow \exists^{=1}(y)(\texttt{MaximumValue}(x, y)))$

  In this figure we assign cardinalities [1..1] to both attributes. The type `Literal` is a data type covering all textual representations of values from simple data types in the model.

2.9 Value comparison constraints

The metamodel fragment for value comparison constraints is depicted in Figure 11. Note that in the formalisation below, a textual representation is chosen for the comparison operators.
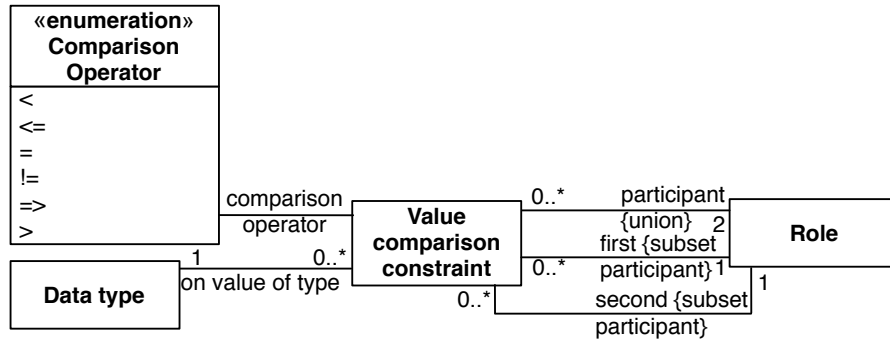
* A *Value comparison constraint* cannot have as *participant* the same *Role*.
* The values subjected to a *comparison operator* in a *Value comparison constraint* must involve values of the same *Data type*.

Figure 11: Value comparison constraint; the roles have to be compatible (note: the enumeration already has a 1..1 constraint).

- relation $(0..*, 2)$ role subset, first textual constraint

$$\forall(x)(\texttt{ValueComparisonConstraint}(x) \rightarrow \exists^{=2}(y)(\texttt{Participant}(x,y)))$$
$$\forall(x,y)(\texttt{First}(x,y) \rightarrow ((\texttt{ValueComparisonConstraint}(x) \wedge \texttt{Role}(y))\vee$$
$$(\texttt{RelationshipConstraint}(x) \wedge \texttt{Role}(y))))$$
$$\forall(x,y)(\texttt{Second}(x,y) \rightarrow ((\texttt{ValueComparisonConstraint}(x) \wedge \texttt{Role}(y))\vee$$
$$(\texttt{RelationshipConstraint}(x) \wedge \texttt{Role}(y))))$$
$$\forall(x,y)(\texttt{First}(x,y) \rightarrow \texttt{Participant}(x,y))$$
$$\forall(x,y)(\texttt{Second}(x,y) \rightarrow \texttt{Participant}(x,y))$$
$$\forall(x,y)((\texttt{Participant}(x,y) \wedge \texttt{ValueComparisonConstraint}(x)) \rightarrow$$
$$(\texttt{First}(x,y) \vee \texttt{Second}(x,y)))$$
$$\forall(x,y)(\neg(\texttt{First}(x,y) \wedge \texttt{Second}(x,y) \wedge \texttt{ValueComparisonConstraint}(x)))$$
$$\forall(x)(\texttt{ValueComparisonConstraint}(x) \rightarrow \exists^{=1}(y)(\texttt{First}(x,y)))$$
$$\forall(x)(\texttt{ValueComparisonConstraint}(x) \rightarrow \exists^{=1}(y)(\texttt{Second}(x,y)))$$

- nominals

$$\forall(x,y)(\texttt{ComparisonOperatorOf}(x,y) \rightarrow (\texttt{ValueComparisonConstraint}(x) \wedge \texttt{Operator}(y)))$$
$$\forall(x)(\texttt{ValueComparisonConstraint}(x) \rightarrow \exists^{=1}(y)(\texttt{ComparisonOperatorOf}(x,y)))$$
$$\forall(x)(\texttt{Operator}(x) \rightarrow ((x = \texttt{Less}) \vee (x = \texttt{Leq}) \vee (x = \texttt{Eq})\vee$$
$$(x = \texttt{Neq}) \vee (x = \texttt{Geq}) \vee (x = \texttt{Greater})))$$
$$\forall(x)((x = \texttt{Less}) \rightarrow (\texttt{Operator}(x) \wedge \neg(x = \texttt{Leq}) \wedge \neg(x = \texttt{Eq})\wedge$$
$$\neg(x = \texttt{Neq}) \wedge \neg(x = \texttt{Geq}) \wedge \neg(x = \texttt{Greater})))$$
$$\forall(x)((x = \texttt{Leq}) \rightarrow (\texttt{Operator}(x) \wedge \neg(x = \texttt{Less}) \wedge \neg(x = \texttt{Eq})\wedge$$
$$\neg(x = \texttt{Neq}) \wedge \neg(x = \texttt{Geq}) \wedge \neg(x = \texttt{Greater})))$$
$$\forall(x)((x = \texttt{Eq}) \rightarrow (\texttt{Operator}(x) \wedge \neg(x = \texttt{Less}) \wedge \neg(x = \texttt{Leq})\wedge$$
$$\neg(x = \texttt{Neq}) \wedge \neg(x = \texttt{Geq}) \wedge \neg(x = \texttt{Greater})))$$
$$\forall(x)((x = \texttt{Neq}) \rightarrow (\texttt{Operator}(x) \wedge \neg(x = \texttt{Less}) \wedge \neg(x = \texttt{Leq})\wedge$$
$$\neg(x = \texttt{Eq}) \wedge \neg(x = \texttt{Geq}) \wedge \neg(x = \texttt{Greater})))$$
$$\forall(x)((x = \texttt{Geq}) \rightarrow (\texttt{Operator}(x) \wedge \neg(x = \texttt{Less}) \wedge \neg(x = \texttt{Leq})\wedge$$
$$\neg(x = \texttt{Eq}) \wedge \neg(x = \texttt{Neq}) \wedge \neg(x = \texttt{Greater})))$$
$$\forall(x)((x = \texttt{Greater}) \rightarrow (\texttt{Operator}(x) \wedge \neg(x = \texttt{Less}) \wedge \neg(x = \texttt{Leq})\wedge$$
$$\neg(x = \texttt{Eq}) \wedge \neg(x = \texttt{Neq}) \wedge \neg(x = \texttt{Geq})))$$

`Operator` is the name assigned to the concept associated with the nominals in Figure 11. These nominals are the only operators allowed, and they are all different.

- relation $(1, 0..*)$

$$\forall(x,y)(\texttt{OnValueOfType}(x,y) \rightarrow (\texttt{ValueComparisonConstraint}(x) \wedge \texttt{DataType}(y)))$$
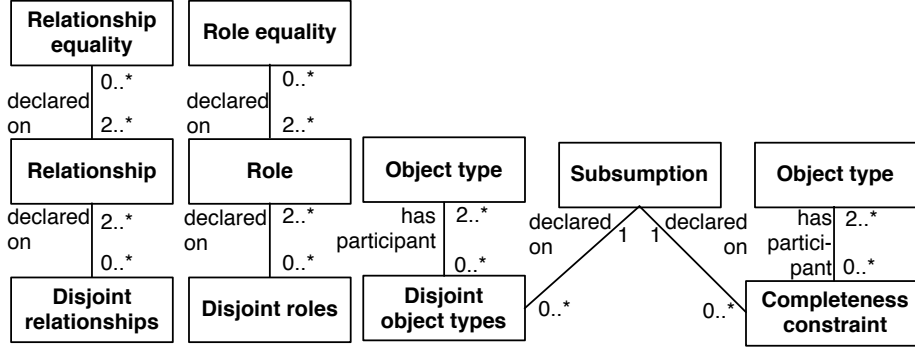$$\forall(x)(\texttt{ValueComparisonConstraint}(x) \rightarrow \exists^{=1}(y)(\texttt{OnValueOfType}(x,y)))$$

- second textual constraint

$$\forall(x,y,z)((\texttt{ValueComparisonConstraint}(x) \wedge \texttt{First}(x,y) \wedge \texttt{Second}(x,z)) \rightarrow$$
$$\exists(u,v,w)(\texttt{RolePlaying}(y,u,v) \wedge \texttt{RolePlaying}(z,w,v) \wedge \texttt{DataType}(v)))^*$$

2.10   Role and Relationship constraints

Constraints among roles and relationship are shown in Figure 12, and it, with its repetitive list of similar textual constraints are formalised afterward in this section. The Relationship constraints (reflexivity etc.) are presented afterward in Figure 13 and its formalisation. This section closes with the metamodel fragment for ORM's join constraints, depicted in Figure 14 and formalised at the end of this section, therewith completing the metamodel and its FOL formalisation.

Formalization of Equality and Disjointness Constraints

Relationship equality		Role equality							

The diagram shows boxes connected with the following labels:

- Relationship equality — declared on 0..* / 2..* — Relationship
- Role equality — declared on 0..* / 2..* — Role
- Relationship — declared on 2..* / 0..* — Disjoint relationships
- Role — declared on 2..* / 0..* — Disjoint roles
- Object type — has participant 2..* / 0..* — Disjoint object types
- Subsumption — declared on 1 / declared on 1 — 0..* / 0..*
- Object type — has participant 2..* / 0..* — Completeness constraint

* For each *Disjoint object types* that is *declared on* two or more *Object type*s,
    these *Object type*s share a direct common subsumer.
* For each *Completeness constraint* that is *declared on* two or more *Object type*s,
    these *Object type*s share a direct common subsumer.
* Compatible *Object type*s are *Object type*s that are of the same category.
* Compatible *Role*s are *Role*s that are *linked to* compatible *Entity type*s
    and those *Entity type*s are either the same or one is a *sub* of the other.
* Compatible *Relationship*s are *Relationship*s that have the same number of *Role*s
    participating in the *sub* and *super* position and those *Role*s are compatible.
* Each *Disjoint object types* is *declared on* compatible *Object type*s.
* Each *Disjoint relationships* is *declared on* compatible *Relationship*s.
* Each *Disjoint roles* is *declared on* compatible *Role*s
    and those *Role*s are *contained in* distinct *Relationship*s.
* Each *Role equality* is *declared on* compatible *Role*s
    and those *Role*s are *contained in* distinct *Relationship*s.
* Each *Relationship equality* is *declared on* compatible *Relationship*s.

Figure 12: Disjointness and completeness constraints, and role and relationship equality.

- four relations DeclaredOn $(0..*, 2..*)$

$\forall(x)(\text{RelationshipEquality}(x) \to \exists^{\geq 2}(y)(\text{DeclaredOn}(x,y)))$

$\forall(x)(\text{DisjointRelationships}(x) \to \exists^{\geq 2}(y)(\text{DeclaredOn}(x,y)))$

$\forall(x)(\text{RoleEquality}(x) \to \exists^{\geq 2}(y)(\text{DeclaredOn}(x,y)))$

$\forall(x)(\text{DisjointRoles}(x) \to \exists^{\geq 2}(y)(\text{DeclaredOn}(x,y)))$

- two relations DeclaredOn $(0..*, 1)$

$\forall(x)(\text{CompletenessConstraint}(x) \to \exists^{=1}(y)(\text{DeclaredOn}(x,y)))$

$\forall(x)(\text{DisjointEntities}(x) \to \exists^{=1}(y)(\text{DeclaredOn}(x,y)))$

- two relations hasParticipant $(0..*, 2..*)$

$\forall(x,y)(\text{HasParticipant}(x,y) \to ((\text{CompletenessConstraint}(x) \vee \text{DisjointEntities}(x)) \wedge$
$\text{Entity}(y) \wedge \neg\text{QualifiedRelationship}(y) \wedge \neg\text{Subsumption}(y) \wedge$
$\neg\text{AttributiveProperty}(y) \wedge \neg\text{Qualifier}(y) \wedge \neg\text{Constraint}(y)))$

$\forall(x)(\text{CompletenessConstraint}(x) \to \exists^{\geq 2}(y)(\text{HasParticipant}(x,y)))$

$\forall(x)(\text{DisjointEntities}(x) \to \exists^{\geq 2}(y)(\text{HasParticipant}(x,y)))$

- textual constraints

$$\forall(x,y)((\texttt{DisjointEntities}(x) \wedge \texttt{HasParticipant}(x,y)) \rightarrow$$
$$\exists(z)(\texttt{DeclaredOn}(x,z) \wedge \texttt{Subsumption}(z) \wedge \texttt{Sub}(z,y)))^*$$
$$\forall(x,y,z)((\texttt{DisjointEntities}(x) \wedge \texttt{DeclaredOn}(x,y) \wedge \texttt{Sub}(y,z)) \rightarrow \texttt{HasParticipant}(x,z))^*$$
$$\forall(x,y)((\texttt{CompletenessConstraint}(x) \wedge \texttt{HasParticipant}(x,y)) \rightarrow$$
$$\exists(z)(\texttt{DeclaredOn}(x,z) \wedge \texttt{Subsumption}(z) \wedge \texttt{Sub}(z,y)))^*$$
$$\forall(x,y,z)((\texttt{CompletenessConstraint}(x) \wedge \texttt{DeclaredOn}(x,y) \wedge \texttt{Sub}(y,z)) \rightarrow$$
$$\texttt{HasParticipant}(x,z))^*$$
$$\forall(x,y)(\texttt{Compatible}(x,y) \rightarrow$$
$$((\texttt{ValueProperty}(x) \wedge \texttt{ValueProperty}(y)) \vee$$
$$(\texttt{DataType}(x) \wedge \texttt{DataType}(y)) \vee$$
$$(\texttt{ObjectType}(x) \wedge \texttt{ObjectType}(y)) \vee$$
$$(\texttt{Role}(x) \wedge \texttt{Role}(y)) \vee$$
$$(\texttt{Relationship}(x) \wedge \texttt{Relationship}(y))))$$
$$\forall(x,y)((\texttt{Compatible}(x,y) \wedge \texttt{Role}(x)) \rightarrow$$
$$\exists(v,w,s,t)(\texttt{RolePlaying}(x,v,w) \wedge \texttt{RolePlaying}(y,s,t) \wedge \texttt{Compatible}(w,t)))^*$$
$$\forall(x,y)((\texttt{Compatible}(x,y) \wedge \texttt{Relationship}(x)) \rightarrow$$
$$((\exists^{=n}(z)(\texttt{Contains}(x,z)) \leftrightarrow \exists^{=n}(z)(\texttt{Contains}(y,z))) \wedge$$
$$(\exists(z,v)(\texttt{Contains}(x,z) \wedge \texttt{Contains}(y,w) \wedge \texttt{Compatible}(z,w)))))^*$$
$$\forall(x,y,z)((\texttt{DisjointRelationships}(x) \wedge \texttt{DeclaredOn}(x,y) \wedge \texttt{DeclaredOn}(x,z)) \rightarrow$$
$$\texttt{Compatible}(y,z))^*$$
$$\forall(x,y,z)((\texttt{DisjointRoles}(x) \wedge \texttt{DeclaredOn}(x,y) \wedge \texttt{DeclaredOn}(x,z)) \rightarrow (\texttt{Compatible}(y,z) \wedge$$
$$\forall(v,w)((\texttt{Contains}(v,y) \wedge \texttt{Contains}(w,z)) \rightarrow \neg(y=z))))^*$$
$$\forall(x,y,z)((\texttt{RoleEquality}(x) \wedge \texttt{DeclaredOn}(x,y) \wedge \texttt{DeclaredOn}(x,z)) \rightarrow (\texttt{Compatible}(y,z) \wedge$$
$$\forall(v,w)((\texttt{Contains}(v,y) \wedge \texttt{Contains}(w,z)) \rightarrow \neg(y=z))))^*$$
$$\forall(x,y,z)((\texttt{RelationshipEquality}(x) \wedge \texttt{DeclaredOn}(x,y) \wedge \texttt{DeclaredOn}(x,z)) \rightarrow$$
$$\texttt{Compatible}(y,z))^*$$

This is necessary first order since n is any natural number up to the maximum arity of a relationships in the model.
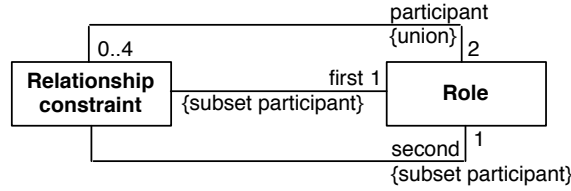
* Each *Relationship constraint* is *declared on* compatible *Role*s
and those *Role*s are *contained in* the same *Relationship*.

Figure 13: Relationship constraints (see Figure 2 for the hierarchy of relational properties).
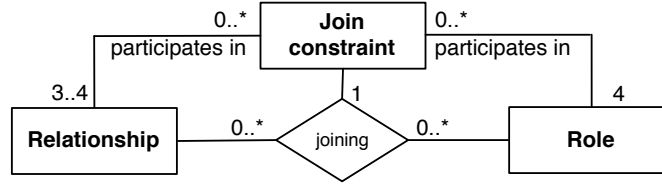
Formalization of Relationship Constraints

- relation $(0..4, 2)$

$$\forall(x)(\texttt{RelationshipConstraint}(x) \rightarrow \exists^{=2}(y)(\texttt{Participant}(x,y)))$$
$$\forall(x)(\texttt{Role}(x) \rightarrow \exists^{\leq 4}(y)(\texttt{Participant}(y,x)))$$

- two relations $(0..*, 1)$, subset, union

$$\forall(x,y)((\texttt{Participant}(x,y) \wedge \texttt{RelationshipConstraint}(x)) \rightarrow (\texttt{First}(x,y) \vee \texttt{Second}(x,y)))$$
$$\forall(x,y)(\neg(\texttt{First}(x,y) \wedge \texttt{Second}(x,y) \wedge \texttt{RelationshipConstraint}(x)))$$
$$\forall(x)(\texttt{RelationshipConstraint}(x) \rightarrow \exists^{=1}(y)(\texttt{First}(x,y)))$$
$$\forall(x)(\texttt{RelationshipConstraint}(x) \rightarrow \exists^{=1}(y)(\texttt{Second}(x,y)))$$

Figure 14: Join constraints.

* The *Relationship*s that *participate in* a *Join constraint* contain the *Role*s that *participate in* that *Join constraint*.
* *Role*s and *Relationship*s may *participate in* more than one *Join constraint*
  only if the combination of *participating Relationship*s and *Role*s is unique.

- textual constraint

$$\forall(x,y,z)((\texttt{RelationshipConstraint}(x,y) \wedge \texttt{RelationshipConstraint}(x,z) \wedge$$
$$\texttt{First}(x,y) \wedge \texttt{Second}(x,z)) \rightarrow (\texttt{Compatible}(y,z) \wedge$$
$$\exists(w)(\texttt{Contains}(w,y) \wedge \texttt{Contains}(w,z))))^*$$

Formalization of Join Constraints

- ternary relation $(0..*, 1, 0..*)$

$$\forall(x,y,z)(\texttt{Joining}(x,y,z) \rightarrow (\texttt{Relationship}(x) \wedge \texttt{Role}(z) \wedge \texttt{JoinConstraint}(y)))$$
$$\forall(x)(\texttt{JoinConstraint}(x) \rightarrow \exists^{=1}(y,z)(\texttt{Joining}(y,x,z)))$$

- relation $(3..4, 0..*)$ and relation $(4, 0..*)$

$$\forall(x,y)(\texttt{ParticipatesIn}(x,y) \rightarrow (\texttt{JoinConstraint}(y) \wedge (\texttt{Relationship}(x) \vee \texttt{Role}(x))))$$
$$\forall(x)(\texttt{JoinConstraint}(x) \rightarrow (\exists^{=3}(y)(\texttt{Relationship}(y) \wedge \texttt{ParticipatesIn}(y,x)) \vee$$
$$\exists^{=4}(y)(\texttt{Relationship}(y) \wedge \texttt{ParticipatesIn}(x,y))))$$
$$\forall(x)(\texttt{JoinConstraint}(x) \rightarrow \exists^{=4}(y)(\texttt{Role}(y) \wedge \texttt{ParticipatesIn}(y,x)))$$

- first textual constraint

$$\forall(x,y)((\texttt{JoinConstraint}(x) \wedge \texttt{Role}(y) \wedge \texttt{ParticipatesIn}(y,x)) \rightarrow$$
$$\exists(z)(\texttt{Relationship}(z) \wedge \texttt{Contains}(z,y) \wedge \texttt{ParticipatesIn}(z,x)))^*$$

2.11 On the complexity of the FOL formalization

The complete FOL formalization of the metamodel is probably undecidable. We have complex formula in five variables, counting quantifiers, and ternary predicates. Even if we reify ternary relations, the complexity of deciding if a given formula follows from some instantiated metamodel is similar to that of the general satisfiability problem. Problematic formula belong though to textual constraints of the figures; they are not originated from the graphical language of UML class diagrams. So if we remove from the metamodel all formula labeled with a "*", then we obtain a lighter formalization described in C^2, the fragment of function-free FOL with only two variables and counting quantifiers. This fragment is not only decidable, but also in NEXPTIME, as shown in [Pratt-Hartmann, 2005]. Most description logics are also subsets of C^2.

This lighter version of the metamodel is the basis for the OWL 2 version described in next section.

3 OWL 2 version of the metamodel

Considering that the complete formalisation of the metamodel requires full FOL, hence, using an undecidable language, we also look at representing a subset in a decidable language to facilitate the metamodel's use in various applications. A relevant and popular logic language in this

context is OWL, so that, thanks to the tools that can process it, we also can have some form of verification that our model is consistent (at least for what is representable in OWL 2 DL). This OWL version of the metamodel (v1.2) is available at `http://www.meteck.org/files/ontologies/metamodelCDML.owl`.

The OWL version is larger than the number of entities in the diagrams of the preceding section, and has 99 classes, 61 object properties, 2 data properties, and overall 657 axioms, and the language used is $\mathcal{SHIQ}(D)$ (data from Protégé v4.3's "ontology metrics" page), i.e., using features of OWL 2 DL [Motik et al., 2009] and in particular the qualified cardinality. Because the diagrams were made in Omnigraffle[1], no UML-in-Omnigraffle to OWL tool exists, there are textual constraints in our metamodel that can be captured formally, and other UML-to-OWL tools are partial, we formalised the metamodel manually. Further, due to the limitations of OWL on the one hand, and the greater expressiveness regarding other features, there is no exact match between the metamodel and the OWL file. We describe the main changes and design decisions in the remainder of this section.

Missing information.   The first thing to note is that the ternary and quaternary associations of the metamodel could not be represented in OWL, and thus had to be approximated. We used the following procedure for this, which is similar to procedures mentioned elsewhere (e.g., the ontology design pattern in [Presutti et al., 2008]):
  1. Add new class ReifiedNAry,
  2. If the n-ary was not reified already then add it as subclass of ReifiedNAry and append it with an R, if present already then add a subsumption to ReifiedNAry.
  3. Add n binaries between said class and the others, where the binaries have a naming scheme of the name of the n-ary appended with a number (1, 2 etc.) to distinguish them from real object properties, and add domain and range classes.
  4. These numbered object properties are not to be used as real object properties elsewhere in the OWL file.
  5. Add an exactly 1 (or other cardinality constraint, depending on the constraint in the model) from the ReifiedNAryR to the participating class.
Note that several ternary relations in the metamodel are already reified, since one of their participant has cardinality constraint 1..1. For these relations, the above procedure can be simplified.

Other information missing from the OWL file are the textual constraints whose formula are marked with "*" in the FOL formalisations, because these constraints cannot be expressed in OWL. Also, a few other complex constraints, such as the "The Single identification has a Mandatory constraint on the participating Role and the Relationship that Role is contained in has a 1:1 Cardinality constraint declared on it." (see second textual constraint in Figure 8), are not included due to OWL's expressiveness limitations.

Other modelling artefacts.   Some other representation choices had to be made. We note them here for information and documentation; their rationale is outside the scope of this technical report.

There are several entities that were not subsumed by anything in particular, such as MaximumCardinality, MinimumCardinality, Dimension, and ValueRange, which have now been made an OWL subclass of Abstract. This also meant that the four attributes in the metamodel have been 'flattened' on the OWL file, i.e., instead of one binary from class to data type with OWL's data property, it uses a separate part relation and then a hasCValue and hasValue, respectively, to the values.

The participation of IdentificationConstraint is twice 0..* in the figure, but that is translated into OWL as "some (ValueProperty or AttributiveProperty)", because the constraint has to be

---

[1] `https://www.omnigroup.com/omnigraffle`; last accessed 5-12-2014.

QUALIFIER (class)

Typology	A qualifier is an entity type.
Description	A qualifier is declared on exactly one thing that is a qualified relationship, or is a role. A qualifier has as part something that is an attribute, or is a dimensional attribute.
Others	disjointUnion(class('#EntityType'), class('#DataType'), class('#ObjectType'), class('#Qualifier'), class('#ValueProperty')).

REIFIED N ARY (class)

Examples	The following are reified n aries: role playing rs, joining rs and ext unique rs, and so on (11 items in total).

RELATIONSHIP (class)

Typology	A relationship is an entity.
Description	If any of the following relationships hold between X and Y then X is a relationship: "is relj1", "is rel4", "is rel3", "is rel2" and "is rel1". A relationship contains at least two roles. A relationship is reified as at most one nested object type. A relationship is participant exactly two entities. A relationship is rel1 at least two ext unique rs, or is rel1 exactly zero ext unique rs.
Others	disjointUnion(class('#Entity'), class('#Constraint'), class('#EntityType'), class('#Relationship'), class('#Role')).
Examples	The following are relationships: subsumptions, qualified relationships and part wholes, and so on (4 items in total).
Distinctions	The following are disjoint: entity types, relationships, and roles.

Figure 15: Section of the OWL file rendered in the SWAT NL controlled natural language (see text for details).

declared on something.

The comparison operators of Figure 11 are written out in text and represented as subclasses of Comparison and is made the disjoint union of its subclasses, and the ValueComparisonConstraint is given exactly 1 Comparison.

It has been shown that using OWL 2's new feature of inverse object properties results in better reasoner performance [Keet et al., 2014]. Therefore, we chose not to use object properties 'in both directions', but use that instead. For instance constrainedWith has not been used in the OWL file, but instead, in Protégé notation, inverse(declaredOn).

We have added partOf, properPartOf, and hasPart.

Satisfiability and consistency checking. We checked the OWL-ized metamodel with the automated reasoner HerMiT v1.3.8 that is shipped with Protégé v4.3. The sole useful inference was the detection of an inconsistency of Asymmetry in the hierarchy of constraints in an earlier version of the metamodel: the {disjoint, complete} on the object subtypes of Relationship constraint in [Keet and Fillottrani, 2013] had to be changed into {complete} in Figure 2 due to the multiple inheritance of Asymmetry that was overlooked. This has been corrected in this version of the metamodel.

Non-logician readability/rendering. One may expect a reader who made it this far to be familiar with at least logic or UML notation and thus have a general understanding of the contents of the metamodel and its formalisation. However, as can be observed from the previous paragraphs about the OWL version of it, there are some finer details of that logic reconstruction. Also, the FOL version of the metamodel is rather light on annotations. To facilitate readability, we have generated an online controlled natural language version (verbalization) of the OWL file, which was made with SWAT Natural Language tools [Third et al., 2011]. This easily navigable HTML document is available online at `www.meteck.org/files/ontologies/metamodelNL.html` and a section of it is shown in Figure 15.

4 Final remarks

We have presented the FOL formalisation of the KF metamodel that serves as a unifying metamodel for the static structural entities of UML Class Diagrams (v2.4.1), ER, EER, ORM, and ORM2. It has to be noted that this indeed requires FOL. Notwithstanding, and with an eye on implementations, we also have created an OWL 2 DL version of it, noting some modelling decisions, such as a pattern for representing n-aries.

Initial use of the formalisation has been demonstrated in [Fillottrani and Keet, 2014], where the constraints represented among the entities induce validation checking of inter-model assertions and model transformations.

Acknowledgments

References

[Artale et al., 2007] Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., and Zakharyaschev, M. (2007). Reasoning over extended ER models. In Parent, C., Schewe, K.-D., Storey, V. C., and Thalheim, B., editors, *Proceedings of the 26th International Conference on Conceptual Modeling (ER'07)*, volume 4801 of *LNCS*, pages 277–292. Springer. Auckland, New Zealand, November 5-9, 2007.

[Baader et al., 2008] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2008). *The Description Logics Handbook – Theory and Applications.* Cambridge University Press, 2 edition.

[Chen, 1976] Chen, P. P. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36.

[Committee Members, 2012] Committee Members (2012). Information technology – metamodel framework for interoperability (MFI) – Part xx: Metamodel for Fact Based Information Model Registration. ISO/IEC WD 19763-xx.02.

[Fillottrani and Keet, 2014] Fillottrani, P. and Keet, C. (2014). Conceptual model interoperability: a metamodel-driven approach. In Bikakis, A. et al., editors, *Proceedings of the 8th International Web Rule Symposium (RuleML'14)*, volume 8620 of *LNCS*, pages 52–66. Springer. August 18-20, 2014, Prague, Czech Republic.

[Halpin and Morgan, 2008] Halpin, T. and Morgan, T. (2008). *Information modeling and relational databases.* Morgan Kaufmann, 2nd edition.

[Hedman, 2004] Hedman, S. (2004). *A first course in logic—an introduction to model theory, proof theory, computability, and complexity.* Oxford University Press, Oxford.

[Keet et al., 2014] Keet, C., d'Amato, C., Khan, Z., and Lawrynowicz, A. (2014). Exploring reasoning with the DMOP ontology. In Bail, S., Glimm, B., Jiménez-Ruiz, E., Matentzoglu, N., Parsia, B., and Steigmiller, A., editors, *3rd Workshop on Ontology Reasoner Evaluation (ORE'14)*, volume 1207 of *CEUR-WS*, pages 64–70. CEUR-WS. July 13, 2014, Vienna, Austria.

[Keet, 2011] Keet, C. M. (2011). Enhancing identification mechanisms in UML class diagrams with meaningful keys. In *Proceeding of the SAICSIT Annual Research Conference 2011 (SAICSIT'11)*, pages 283–286. ACM Conference Proceedings. Cape Town, South Africa, October 3-5, 2011.

[Keet and Fillottrani, 2013] Keet, C. M. and Fillottrani, P. R. (2013). Toward an ontology-driven unifying metamodel for UML class diagrams, EER, and ORM2. In Ng, W., Storey, V. C., and Trujillo, J., editors, *32nd International Conference on Conceptual Modeling (ER'13)*, volume 8217 of *LNCS*, pages 313–326. Springer. 11-13 November, 2013, Hong Kong.

[Keet and Fillottrani, 2014] Keet, C. M. and Fillottrani, P. R. (2014). An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2. *Data & Knowledge Engineering*, (submitted).

[Motik et al., 2009] Motik, B., Patel-Schneider, P. F., and Parsia, B. (2009). OWL 2 web ontology language structural specification and functional-style syntax. W3c recommendation, W3C. http://www.w3.org/TR/owl2-syntax/.

[Object Management Group, 2012] Object Management Group (2012). Superstructure specification. Standard 2.4.1, Object Management Group. http://www.omg.org/spec/UML/2.4.1/.

[Pratt-Hartmann, 2005] Pratt-Hartmann, I. (2005). Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language and Information*, 14(3):369–395.

[Presutti et al., 2008] Presutti, V., Gangemi, A., David, S., de Cea, G. A., Surez-Figueroa, M. C., Montiel-Ponsoda, E., and Poveda, M. (2008). A library of ontology design patterns: reusable solutions for collaborative design of networked ontologies. NeOn deliverable D2.5.1, NeOn Project, Institute of Cognitive Sciences and Technologies (CNR).

[Song and Chen, 2009] Song, I.-Y. and Chen, P. P. (2009). Entity relationship model. In Liu, L. and Özsu, M. T., editors, *Encyclopedia of Database Systems*, volume 1, pages 1003–1009. Springer.

[Thalheim, 2009] Thalheim, B. (2009). Extended entity relationship model. In Liu, L. and Özsu, M. T., editors, *Encyclopedia of Database Systems*, volume 1, pages 1083–1091. Springer.

[Third et al., 2011] Third, A., Williams, S., and Power, R. (2011). OWL to English: a tool for generating organised easily-navigated hypertexts from ontologies. poster/demo paper, Open Unversity UK. 10th International Semantic Web Conference (ISWC'11), 23-27 Oct 2011, Bonn, Germany.