Multi-Level Conceptual Modeling: Theory and Applications

João Paulo A. Almeida[1], Victorio A. Carvalho[2], Freddy Brasileiro[1], Claudenir M. Fonseca[4], Giancarlo Guizzardi[1,4]

[1]Ontology & Conceptual Modeling Research Group (NEMO)
Federal University of Espírito Santo – Vitória, ES – Brazil

[2]Federal Institute of Espírito Santo (IFES) – Colatina, ES – Brazil

[3]Free-University of Bozen-Bolzano – Bolzano – Italy

Abstract. *Conceptual models are often built with techniques which propose a strict stratification of entities into two classification levels: a level of types (or classes) and a level of instances. Despite that, there are several situations in which domains of inquiry transcend the conventional two-level stratification and domain experts use types of types (or categories of categories) to articulate their conceptualizations. In these settings, types are instances of other types and multiple levels of classification can be identified (individuals, classes, metaclasses, metametaclasses, and so on), characterizing what is now called "multi-level modeling". Over the last years, we have worked out a foundational theory for multi-level modeling (dubbed MLT), whose aim is to clarify the basic elements of multi-level conceptual modeling. This paper describes the development of this theory, and reports on some of its applications, namely: the detection of (thousands of) occurrences of anti-patterns in the Wikidata knowledge base and the revision of the powertype pattern in UML.*

1. Introduction

Conceptual modeling is a complex task that involves understanding a subject domain (some portion of "the physical and social world around us" [Mylopoulos 1992]) and producing a symbolic artifact (a conceptual model) to represent the various entities in that domain. Usually, this task is undertaken by capturing invariant aspects of the entities in the domain, which is supported in most conceptual modeling approaches through constructs such as "classes" and "types", reflecting the use of "kinds", "categories" and "sorts" in accounts of a subject domain by subject matter experts.

In the conventional two-level representation scheme, a conceptual model is stratified into entities in two levels: a level of types (or classes) and a level of instances. The level of types captures invariants which apply exclusively to the level of individuals. In this scheme, the subject matter can be understood as consisting of individuals, and the purpose of the conceptual model is to establish which structures of individuals are admissible according to some (shared) conceptualization of the world [Guizzardi 2007].

The two-level scheme, however, reveals its limitations whenever categories of categories are part of the domain of inquiry. In this setting, we are interested not only in invariants about individuals but also about categories themselves [Carvalho and Almeida 2018]. For example, in the military domain, experts often refer to "vessel types", "vessel classes" or "ship classes" in their accounts. Instances of VESSEL TYPE, such as CARGO SHIP, CIVILIAN SHIP, SUBMARINE, SUPERCARRIER or NIMITZ-CLASS AIRCRAFT CARRIER are themselves types, instantiated by individual vessels. For instance, the USS ABRAHAM LINCOLN (an aircraft carrier of the United States Navy) is an instance of both the SUPERCARRIER and NIMITZ-CLASS AIRCRAFT CARRIER types. In their turn, SUPERCARRIER and NIMITZ-CLASS AIRCRAFT CARRIER are instances of VESSEL TYPE. Thus, to describe the conceptualization in this domain, one needs to represent entities of different (but nonetheless related) classification levels, such as specific ships out there in a mission, types of ships, and (even) types of types of ships (VESSEL TYPE, SUBMARINE TYPE, CARGO SHIP TYPE). Other examples of multiple classification levels come from domains such as that of organizational roles (or professional positions) [Carvalho and Almeida 2015], software engineering [Gonzalez-Perez and Henderson-Sellers 2006], biological taxonomy [Atkinson and Kühne 2003] and product types [Neumayr et al. 2009].

The need to support the representation of subject domains dealing with multiple classification levels has given rise to what has been referred to as "multi-level modeling" [Atkinson and Kühne 2001, Neumayr et al. 2009, Almeida et al. 2018]. Techniques for multi-level conceptual modeling must provide modeling concepts to deal with types in various classification levels and the relations that may occur between those types. These approaches embody conceptual notions that are key to the representation of multi-level models, such as the existence of entities that are simultaneously types and instances, the iterated application of instantiation across an arbitrary number of levels, the possibility of defining attributes and values at the various type levels, etc.

Over the last years, we have worked out a foundational theory for multi-level modeling (dubbed MLT), whose aim is to clarify the basic elements of multi-level conceptual modeling. Our investigation was motivated by the lack of (language-independent) foundations for multi-level modeling. In particular, we were driven by the need to clarify (and thereby demystify) the semantics of multi-level modeling constructs as well as to justify modeling choices. This paper describes the development of this theory, and reports on two applications, including: the detection of (thousands of) occurrences of anti-patterns in the Wikidata knowledge base and the revision of the powertype pattern in UML.

This paper is further structured as follows. Section 2 motivates multi-level modeling by discussing the workarounds which are required when we are confined to two levels of classification. Section 3 leads us into multi-level modeling territory, introducing briefly the MLT theory by discussing its key definitions, and presenting rules that arise out of these definitions. Sections 4 and 5 discuss how MLT was applied to identify problems in Wikidata multi-level taxonomies as well as to revise the UML support for powertypes. Finally, section 6 provides concluding remarks and outlines current research in this topic.

(This is an invited paper to accompany the keynote speech delivered by the first author at the 2018 Ontobras Seminar on Ontology Research in Brazil. It focuses on some of the contributions that were offered in successive publications over the last three years, including [Almeida et al. 2017, Carvalho and Almeida 2018, Brasileiro et al. 2016b, Carvalho et al. 2017, Carvalho et al. 2015], a Ph.D. thesis [Carvalho 2016] and two M.Sc. thesis [Brasileiro 2016, Fonseca 2017]. It does not cover formalization and related work. We refer the reader to these other publications for details.)

2. Two-Level Workarounds

Let us consider the biological taxonomy domain as a paradigmatic example of a multi-level domain [Atkinson and Kühne 2003, Brasileiro et al. 2016b]. The biological taxonomy for living beings classifies living beings according to biological taxa (such as, e.g., ANIMAL, MAMMAL, CARNIVORAN, LION), each of which is classified by a biological taxonomic rank (e.g., KINGDOM, CLASS, ORDER, SPECIES) [Mayr 1982]. CECIL (the lion killed in the Hwange National Park in Zimbabwe in 2015) is an instance of LION, which is an instance of SPECIES. SPECIES, in its turn, is an instance of TAXONOMIC RANK.

In the conventional two-level approach, entities in the domain have to be classified either as classes (or types) or as instances. Strictly speaking, there is no room for meta-types such as SPECIES or meta-meta-types such as TAXONOMIC RANK. Workarounds are available as discussed in [Lara et al. 2014, Kühne and Schreiber 2007], but these often introduce accidental complexity. For example, an early approach that has aimed to accommodate multiple domain levels within two modeling levels is the powertype pattern proposed by [Odell 1994]. In this pattern, all types are treated as regular classes, and a "base type" (such as ORGANISM) is related to a "powertype" (such as SPECIES) through a user-defined (and regular) association (such as ISCLASSIFIEDBY), see Figure 1.

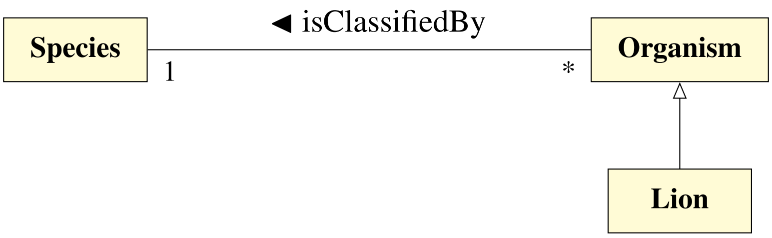

Figure 1. The powertype pattern with a regular user-defined association

This creates a number of difficulties, some of which are discussed in [Lara et al. 2014, Kühne and Schreiber 2007]. First of all, a modeler needs to handle two notions of instantiation, one provided by the modeling technique (and thus between classes and instances) and another that corresponds to the user-defined ISCLASSIFIEDBY association. In the case of the latter, since it is a regular user-defined association, no support for its instantiation semantics is provided by the modeling technique, and hence, instantiation semantics needs to be *emulated* by the modeler. This is particularly involved when the powertype has features of its own (e.g., whether a species is extinct or endangered). Further, the pattern creates a duplication of the same entities, affecting parsimony: this is because instances of the powertype must be admitted both at the instance level (e.g., LION as an instance of SPECIES) and at the same time at the class level (e.g., LION as a specialization of ORGANISM).

In the Semantic Web lightweight ontologies, a workaround called *punning* explores this duplication as a semantic approach. In order to maintain a strict separation of terminological (T-Box) and assertion (A-Box) fragments, a class is also considered as an individual in the A-Box when it instantiates another class. Its interpretations as a class and as an individual are completely independent of each other. This "independent" interpretation means that a constraint stated to a class will not be considered when it is seen as an individual, which leads to non-intuitive interpretations [Pan and Horrocks 2006]. For instance, consider the following statements: (i) HARRY is an instance of GOLDEN EAGLE, and; (ii) GOLDEN EAGLE is the same as AQUILA CHRYSAETOS. With *punning*, statement (i) treats GOLDEN EAGLE as a class, while statement (ii) treats GOLDEN EAGLE as an individual. These two aspects of GOLDEN EAGLE are never considered at the same time for reasoning. Thus, in this approach, it is impossible to infer that HARRY is an instance of AQUILA CHRYSAETOS, which violates our intuition with respect to the multi-level model [Brasileiro et al. 2016b].

3. MLT - A Multi-Level Modeling Theory

The two-level workarounds fail to address the root of the problem: there are in fact two facets to classes, and classes may themselves be regarded as instances of other (meta)classes [Atkinson and Kühne 2000, Foxvog 2005]. This observation leads us to consider techniques which do not limit their support to two levels of classification, often under the banners of "metamodeling" and "multi-level modeling".

3.1. Basic Requirements

First of all, an essential requirement[1] for a multi-level modeling theory is to account for entities of multiple classification levels, which are related through chains of instantiation between the involved entities (requirement R1). This means that the theory must admit entities that represent both types (class) and instances (object) simultaneously [Atkinson and Kühne 2001], diverging thereby from the traditional two-level scheme, in which classification (instantiation) relations are only admitted between classes and individuals.

Second, the size of chains of instantiation may vary according to the nature of the phenomena being captured and according to the model purposes. Because of this, a general-purpose theory should admit an arbitrary number of classification levels (R2) (including the two-level scheme as a special case). The ability to deal with an arbitrary number of levels is identified as a key a requirement by many authors (e.g., see [Atkinson and Gerbig 2012, Frank 2014]. Several examples of three- and four-level models are available in the literature as well as in structured data repositories such as Wikidata (in which there are more than 17,000 classes involved in multi-level taxonomies [Brasileiro et al. 2016a]).

Further, in previous work, some of us have found empirical evidence to support the claim that representations capturing chains of instantiation can benefit greatly from rules for organizing entities into levels [Brasileiro et al. 2016a]. This is because when chains of instantiation relations are allowed, developers need support to avoid producing incorrect models. The so-called "level-blind" approaches [Atkinson et al. 2014] fail to provide such support. We have found that over 87% of the classes in multi-level taxonomies in Wikidata were involved in errors that could have been prevented with some support to detect the inadequate use of instantiation (and its combination with subtyping). Based on this evidence, we consider that a multi-level modeling theory should define principles (rules) for the organization of entities into levels (R3). An example of this sort

[1]These requirements are based in [Brasileiro et al. 2016b, Almeida et al. 2017]. [Carvalho 2016] reviews the multi-level modeling literature to corroborate these requirements.

of principle, which is adopted in some prominent multi-level modeling approaches, is the strict metamodeling principle [Atkinson and Kühne 2000], which prescribes the arrangement of elements into stratified levels mandating that elements of a level only instantiate elements of the level immediately above.

Finally, an important characteristic of domains spanning multiple levels of classification is that there are domain rules that apply to the instantiation of types of different levels. For example, in a conceptual model encompassing the notions of DOG BREED and DOG, all instances of DOG BREED (e.g. COLLIE and BEAGLE) are types whose instances are instances of DOG. Hence, in this setting, instances of DOG BREED specialize DOG. Given the recurrence of this kind of scenario [Lara et al. 2014], which in the past motivated the powertype pattern [Odell 1994], a comprehensive multi-level modeling theory should be able to account for the rules that govern the instantiation of related types at different levels (R4).

3.2. Definitions and Resulting Rules

Based on the requirements discussed in the previous section, we have defined an axiomatic theory to establish key notions in multi-level modeling, building up on the primitive notion of instantiation. The resulting theory (MLT) can be considered a reference top-level ontology for types in multi-level conceptual modeling.

The theory quantifies over individuals and types alike. Types are organized into stratified levels related by instantiation. We have called these levels "orders" to distinguish this principle of organization from other principles in the multi-level modeling literature.

In this section, we present informal definitions to introduce the reader to the theory and its applications. For a formalization of the theory in first-order logic, see the original paper describing it [Carvalho and Almeida 2018]. A formalization in the lightweight formal method Alloy is also available in `https://nemo.inf.ufes.br/mlt/`. The formalization in Alloy was instrumental in exploring the various alternatives for MLT; an iterative design cycle with simulation of implied models and verification of the various theorems was performed with the support of the Alloy analyzer.

The following key definitions are proposed for MLT:

- *Individuals* are those entities which cannot possibly play the role of type in the instantiation relation. Examples include ALBERT EINSTEIN, CECIL THE LION and the 1985 MEXICO CITY EARTHQUAKE.
- *First-order types* are those types whose instances are individuals. Examples include PERSON, ADULT, LION, ANIMAL, ORGANISM, EARTHQUAKE, NATURAL DISASTER, EVENT.
- *Second-order types* are those types whose instances are first-order types. Examples include PERSON TYPE BY AGE, ANIMAL SPECIES, SPECIES, EVENT TYPE.
- *Third-order types* are those types whose instances are second-order types (TAXONOMIC RANK), and so on. The topmost order can be established as required by applications, and the scheme can thus be extended to cope with an arbitrary number of levels.

The following rules were established for MLT:

- Any entity is either an individual or a type classified into a particular order;

- Two types are equal iff all their possible instances are the same (i.e., if they are necessarily coextensional).

Using the basic framework outline above, a number of intra-level and cross-level relations are defined in MLT. This was done in order to account for the notion of powertype, more specifically clarifying and positioning conflicting definitions (namely those of [Cardelli 1988] and [Odell 1994]). The following relations were defined:

- *Specialization* was defined as usual, i.e., a type *specializes* a supertype whenever all its instances are also instances of the supertype;
- *Proper specialization* was defined for the cases in which the extension of the specialized type is a proper subset of the extension of the general type [Henderson-Sellers 2012]);
- A *is powertype of* cross-level relation between types was established to capture the notion of powertype as defined by [Cardelli 1988]. A type pt *is powertype of* a (base) type t iff all instances of pt are specializations of t and all possible specializations of t are instances of pt. Powertypes in this sense are analogous to powersets. The powerset of a set A is a set that includes as members all subsets of A (including A itself).
- A *categorization* relation between types was defined to reflect Odell's notion of powertype [Odell 1994]. Differently from Cardelli's, Odell's definition excludes the base type from the set of instances of the powertype. Further, not all specializations of the base type are required to be instances of the powertype. Odell's definition corresponds more directly to the notion of powertype that was incorporated in the UML. Thus, there may be specializations of the base type that are not instances of the categorizing higher-order type. For example, we may define a type named ORGANISM TYPE BY HABITAT (with instances TERRESTRIAL ORGANISM and AQUATIC ORGANISM) that *categorizes* ORGANISM. ORGANISM TYPE BY HABITAT is not a (Cardelli) powertype of ORGANISM since there are specializations of ORGANISM that are not instances of ORGANISM TYPE BY HABITAT (e.g. PLANT and GOLDEN EAGLE) [Brasileiro et al. 2016b].
- Specializations of the *categorization* relation were defined in order to capture different scenarios of categorization: *disjoint categorization*, to accommodate the cases in which each instance of the base type is instance of at most one instance of the higher-order type; *complete categorization*, when each instance of the base type is instance of at least one instance of the higher-order type; and *partitioning*, when an instance of the base type is instance of exactly one instance of the higher-order type.

The consequences of the theory can be used to identify sound multi-level structures, including the following derived rules (these rules are formally proven theorems that follow from the MLT definitions and axioms):

- The *instance of* relation in MLT is irreflexive, antisymmetric and anti-transitive. Further, it only relates entities of adjacent levels, in a strictly stratified scheme;
- *Specialization* is a partial order (i.e., a reflexive, antisymmetric and transitive relation);
- *Proper specialization* is a strict partial order (i.e., an irreflexive, antisymmetric and transitive relation);

- *Specialization* (whether proper or not) cannot cross level boundaries (i.e., a first-order type can only be specialized by first-order types, a second-order type can only be specialized by second-order types, and so on).
- Both relations *is powertype of* and *categorizes* can only be applied between adjacent levels, with the base type one order lower than the high-order type;
- The *powertype* of a base type (in Cardelli's sense) is unique for that base type;
- Types that *categorize* a base type always specialize the base type's *powertype*;
- If two types *partition* the same base type then they cannot specialize each other.

As we will show in section 4, some of these rules can be employed to detect problems in knowledge bases which include multi-level taxonomies. They are not available to level-blind approaches and are also not available to users of the powertype pattern that emulate the instantiation relation with an association without specialized semantics. Further, as we will show in section 5, these rules can serve as a basis to inform the revision of the UML powertype pattern, along with corresponding tool support.

The theory also accounts for attributes of high-order types, and a phenomenon called *deep instantiation* [Atkinson and Kühne 2003], when the attributes of a higher-order type affect entities at lower levels. We omit this discussion here, and details can be found in [Carvalho and Almeida 2018].

4. Assessing Wikidata's Multi-Level Taxonomic Structures with MLT

An early application of MLT concerned the assessment of the multi-level taxonomies in the structured Wikidata knowledge base. The Wikidata project aims to create a structured version of the content in Wikipedia, thereby facilitating automated querying and processing. Wikidata structures its knowledge base into the so-called "items" encompassing classes and individuals alike. "Statements" capture information about "items", using property–value pairs resulting in a linked data structure.

Two properties are central to structure the content in Wikidata and can be applied to any item: the *instance of* (P31) and *subclass of* (P279). The definition of *instance of* provided in Wikidata is informal and silent about its formal logic properties (concerning symmetry, reflexivity and transitivity). In its turn, *subclass of* is explicitly characterized as transitive and asymmetric (i.e., antisymmetric and irreflexive) (corresponding to *proper specialization*). By chaining the *instance of* property, Wikidata can represent many levels of instantiation (R1, R2). However, since Wikidata is silent about formal logic properties of *instance of*, it offers no special support both for organization of entities along levels (not satisfying R3). We have observed that this leads to a number of representation issues.

To access data, we used the simplified and derived RDF dumps of Wikidata from January 4th, 2016, available at RDF Exports from Wikidata[2]. Moreover, we have queried these using SPARQL, where *instance of* and *subclass of* are represented as `rdf:type` and `rdfs:subClassOf`, respectively. The total number of classes involved in taxonomic hierarchies is 337,102. This number is obtained by counting the items that are either a subject or an object in *subclass of* statements. From this total number of items, 17,819 classes are also the object of *instance of* statements, which means they are simultaneously classes and instances of other classes, and thus involved in hierarchies spanning more than one level of classification (our target classes for this investigation).

[2] `http://tools.wmflabs.org/wikidata-exports/rdf/`

In order to obtain some indication of the use of multi-level hierarchies in Wikidata, we have queried for three simple cases of anti-patterns that violate the MLT stratification of types into orders [Brasileiro et al. 2016a].

We highlight here two anti-patterns detected. For the first one (AP1 in [Brasileiro 2016]), we have found 15,177 occurrences, covering many domains, such as software, sports, biology, food, professions, events. Considering that the total number of classes involved in multi-level taxonomic hierarchies is 17,819, violations of stratification according to AP1 appear in 85% of all cases involving multi-level modeling.

The structure of this first anti-pattern is shown in Figure 2. The pattern violates stratification of orders with a combination of subclassing (any depth) and instantiation. Occurrences of this pattern in Wikidata are shown in the right-hand side of the figure. In the first fragment, COMPUTER SCIENTIST is a subclass of PROFESSION. Since TIM BERNERS-LEE is declared to be an instance of COMPUTER SCIENTIST we are forced to conclude that Tim is an instance of PROFESSION(!), which clearly violates our sense of what professions and persons are. Formally, these statements are inconsistent in the light of MLT: since *instance of* is anti-transitive and COMPUTER SCIENTIST is *instance of* PROFESSION, Tim cannot be declared *instance of* PROFESSION. The mistake is probably the subclassing of PROFESSION into CREATOR. If the subclassing statement is removed, we are able to stratify the model into levels, and recognize that PROFESSION is a second-order type, while its instances are first-order types. In the second fragment, EARTHQUAKE is considered both an instance of, and a specialization of NATURAL DISASTER. Considering that instances of NATURAL DISASTER are specific events, i.e., specific occurrences of natural disasters, then these instances are attributed a point in time. For example, we can say that the "1985 Mexico City earthquake" took place on "September 19th, 1985". However, since EARTHQUAKE is also declared to be an instance of NATURAL DISASTER, EARTHQUAKE itself—a type!—can be said to occur at a specific point in time. In this example, it seems that the undesired relation is the *instance of* relation between EARTHQUAKE and NATURAL DISASTER, and perhaps, there is a missing NATURAL DISASTER TYPE second-order class.

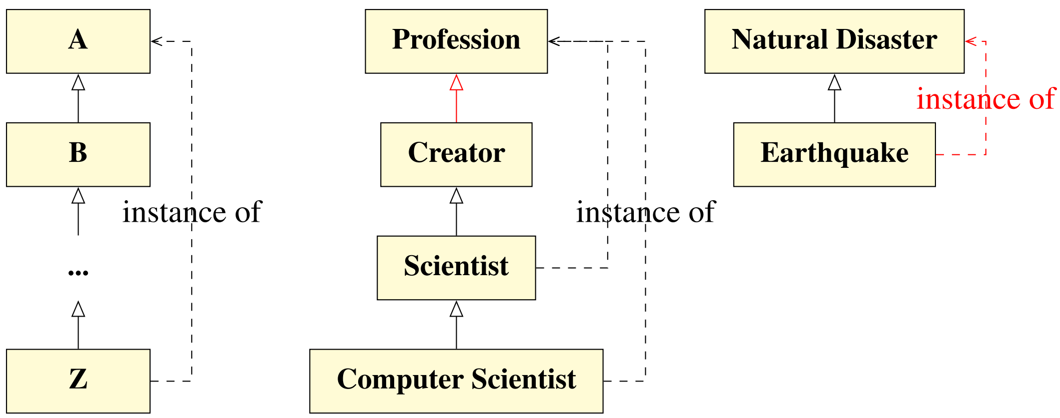

Figure 2. The Structure and Occurrences of AP1 in Wikidata

For a second anti-pattern (AP3 in [Brasileiro 2016]), we have found 7,802 occurrences, again covering many domains. The structure of this pattern is shown in Figure 3. In this pattern, instantiation relations prevent the model from being stratified into orders.

A fragment with an occurrence of this pattern is shown in the right-hand side of Figure 3. CENTRAL PARK[3], the public park at the center of Manhattan in New York City, is considered an instance of both URBAN PARK and PARK, while URBAN PARK is also an instance of PARK. As a subclass of GEOGRAPHIC LOCATION (not shown in the figure), PARK defines the property *coordinate location*. This is not a problem for CENTRAL PARK since it has a specific location. However, this is problematic for the URBAN PARK class, as it does not have a specific location. In this example, it seems that the incorrect relation is the *instance of* relation between URBAN PARK and PARK, which should possibly be replaced by a *subclass of* relation.

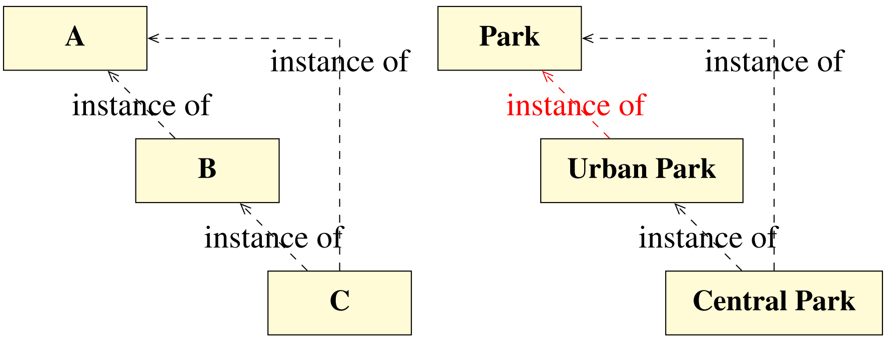

Figure 3. The Structure and Occurrence of AP3 in Wikidata

5. Revising UML Powertype Support with MLT

A reference ontology can be used to inform the revision and redesign of a modeling language, not only through the identification of semantic overload, construct deficit, construct excess and construct redundancy [Guizzardi 2015], but also through the definition of modeling patterns and *semantically-motivated syntactic constraints* [Carvalho et al. 2014]. Using MLT as a reference ontology, we have proposed a well-founded revision for the powertype support in UML 2.x, which resulted in a lightweight UML profile reflecting the MLT rules [Carvalho et al. 2016][4]. Our aim is to equip the modeler with means to avoid common modeling issues such as those shown in section 4.

Basically, the powertype pattern in UML follows Odell's work. However, differently from what was discussed in section 2, the UML 2.x support for powertypes relies on a construct called "generalization set". A generalization set is a means to identify sets of specializations (or generalizations as they are called in UML). Each generalization is a binary relation between two classes, and these binary relations are bundled in a "generalization set". The generalization set is then annotated with constraints and, optionally, a powertype (in Odell's sense) of the type being specialized. Figure 4 revisits the example of Figure 1, now using UML 2.x powertype support. Although a regular association is still used to relate the base type to the powertype, the powertype becomes a powertype in the model through its role in a generalization set (hence, differently from the two-level workaround discussed in section 2, there is some explicit support for powertypes).

[3]In Figure 3, we abuse the diagrammatic notation and do not distinguish between classes and individuals.

[4]We emphasize the version number here, since the support for the powertype pattern in the now obsolete version 1.x was quite different.

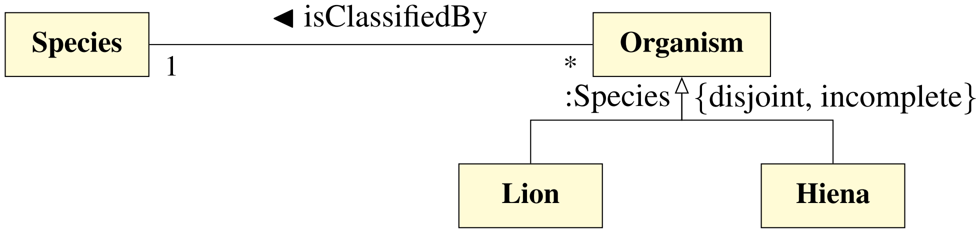

Figure 4. The powertype pattern in plain UML 2.x

The only syntactic constraint defined in UML concerning powertypes is that "the classifier that maps to a generalization set may neither be a specific nor a general classifier in any of the generalization relationships defined for that generalization set" [OMG 2017]. While this rule prevents the powertype from being involved in the generalization set defined to represent its own relation with the base type, this constraint is insufficient to rule out scenarios in which the powertype is incorrectly related by generalization with types of any other levels. Thus errors in multi-level models would remain unnoticed in UML. This has led us to incorporate MLT rules in UML tool support [Carvalho et al. 2016].

Further, a key observation is that for a classifier to be considered a "powertype" in UML, it must be related to a generalization set. Thus, in UML, the powertype pattern can only be applied when specializations of the base type are explicitly modeled (otherwise there would be no generalization set). We consider this undesirable as it rules out simple models such as one defining SPECIES as a "powertype" of ORGANISM, without forcing the modeler to "hardcode" specific instances for SPECIES. Thus our first recommendation concerning the UML support for powertypes is to drop the reliance on generalization sets.

Our solution is to mark the association between the base type and the higher order type with the ≪instantiation≫ stereotype, in order to distinguish it from other domain relations that do not have an instantiation semantics. An association stereotyped ≪instantiation≫ represents that instances of the target type are instantiated by instances of the source type and, thus, denote that there is a categorization relation between the involved types (regardless of possible generalization sets). Variations of the categorization—disjoint categorization, overlapping categorization and partitioning—can be identified through the target cardinality of the ≪instantiation≫ association (0..* in the case of categorization, lower bound 1 in the case of complete categorization, upper bound 1 in the case of disjoint categorization and exactly 1 in the case of partitioning). Figure 5 shows this recommendation in use, revisiting the biological taxonomy example, with no explicit subtypes of ORGANISM required.

Figure 5. Partitioning with the MLT profile (no generalization set required)

Whenever a modeler wants to make generalization sets explicit, this is allowed in the usual way, as shown in Figure 6. Note that the constraints on the generalization set are not identical to those in the categorization. In this particular case, SPECIES partitions ORGANISM, which means that it completely and disjointly categorizes ORGANISM.

This reflects a domain conceptualization that considers every organism to belong to exactly one species. However, the generalization set is marked incomplete, reflecting the fact that there are other instances of SPECIES that are not explicitly enumerated in the generalization set.

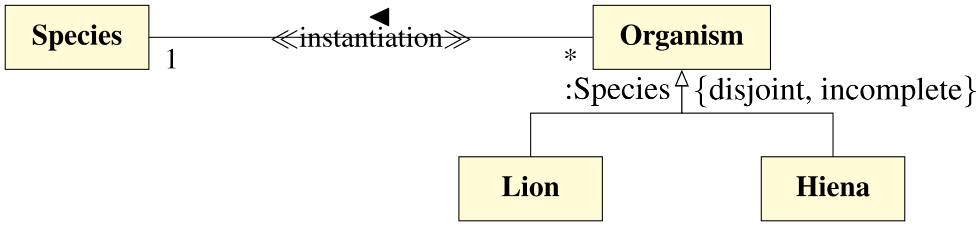

Figure 6. Making some instances of the higher-order type explicit

The MLT profile establishes rules concerning the constraints that can be employed in combination with ≪instantiation≫ associations: (i) in the case of a (non-complete) categorization (lower bound 0 in the target of the ≪instantiation≫), a generalization set marked with the higher-order class cannot carry a *complete* constraint; (ii) in the case of disjoint categorization and partitioning (upper bound 1 of the instantiation relation), a generalization set marked with the higher-order class cannot carry an *overlapping* constraint. Figure 7 shows an example of an invalid combination of constraints that can be detected by the syntactic constraints in the MLT profile. If the generalization set is marked *complete*, than any instance of PERSON must instantiate also at least one of the subclasses MAN or WOMAN. However, the lower bound 0 indicates that there are instances of PERSON which do not instantiate instances of PERSON TYPE BY GENDER. This leads to a contradiction, and either: (i) the constraint should be amended (e.g., changed to *incomplete*, thereby admitting in the conceptualization persons with no gender), or (ii) the lower bound should be fixed to 1 (to establish that every person instantiates a PERSON TYPE BY GENDER).

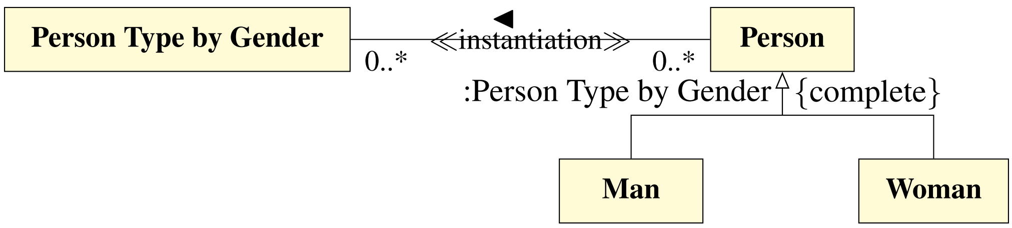

Figure 7. An *invalid* combination of constraints ruled out by the MLT profile

Our final recommendation is to use the ≪powertype≫ stereotype to represent Cardelli's notion of powertype. If a class stereotyped ≪powertype≫ is the target of an ≪instantiation≫ association, then it is the (Cardelli) powertype of the base type. An example of this is shown in Figure 8, with the introduction of the ORGANISM TYPE ≪powertype≫ for ORGANISM. Because of the semantics of Cardelli's powertype, all other specialization of ORGANISM are instances of ORGANISM TYPE, including in this example MALE LION and LIONNESS, which are not instances of SPECIES. The lower bound of the association attached to ORGANISM TYPE is 1, because all instances of ORGANISM are at least instances of ORGANISM, which is, by definition, an instance of

ORGANISM TYPE. Finally, the specialization between SPECIES and ORGANISM TYPE can be inferred automatically, since all types that partition a base type specialize that type's unique ≪powertype≫. This corresponds to our intuition that every SPECIES is an ORGANISM TYPE.

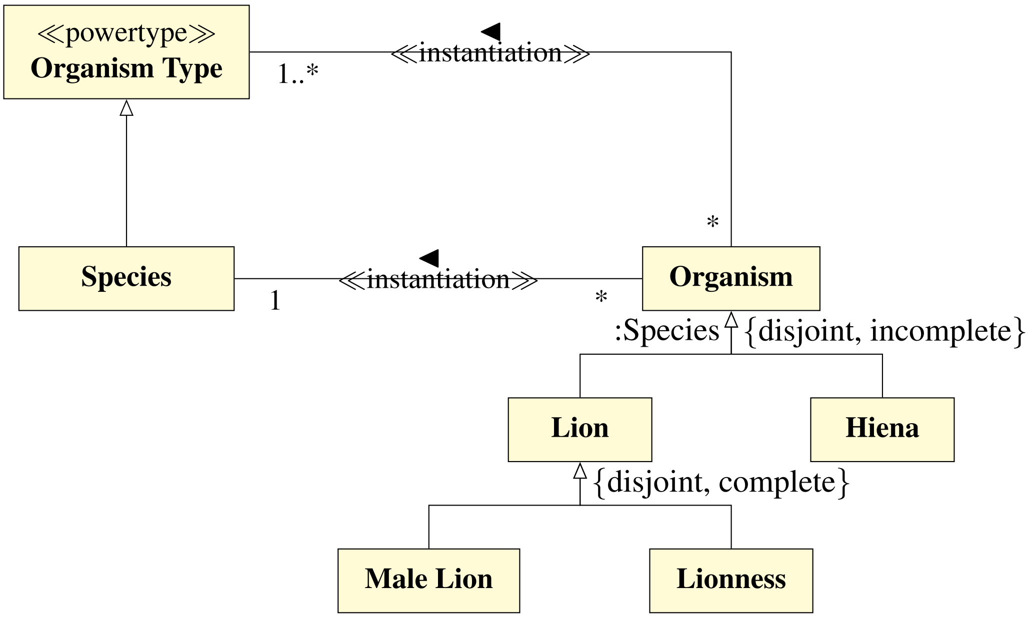

Figure 8. Cardelli's powertype in the MLT profile

6. Concluding Remarks

In this paper, we have briefly described the MLT Multi-Level Theory and discussed two of its applications. Other applications of MLT thus far include: (i) a combination with the Unified Foundational Ontology (UFO) to equip it with means to deal with second-order types in ontology-driven conceptual modeling [Carvalho et al. 2017]; (ii) the design of a core ontology for organizational structure modeling [Carvalho and Almeida 2015]; and (iii) the design of an OWL vocabulary to support the representation of multi-level vocabularies in the Semantic Web [Brasileiro et al. 2016a].

MLT differs from a number of related works in multi-level conceptual modeling given its focus on conceptual modeling (rather than language engineering) and further given the use of formal tools in its conception. A contribution of MLT to the literature on multi-level modeling concerns the study and harmonization of different notions of powertype in the literature, as well as a formal harmonization of powertype and clabject-based approaches.

Recently, we have extended MLT in order to account for types that do not fit a particular order and thereby defy a strict stratification scheme [Almeida et al. 2017]. This was necessary as this scheme rules out abstract and general types such as ENTITY and TYPE (which are instances of themselves). We have observed that these types correspond to general notions that are ubiquitous in comprehensive conceptualizations, see e.g., (foundation) ontologies such as UFO [Guizzardi 2015], Cyc [Foxvog 2005], DOLCE and BFO with their notions of ENTITY or THING, Telos [Mylopoulos 1992] with the notion of “Property” and the Semantic Web with its `rdfs:Resource` and `owl:Thing` elements. The extended theory—which we call MLT*—generalizes the two-level scheme and the strictly stratified scheme [Almeida et al. 2017]. With the extended theory, we can account for types that defy stratification and still provide guidance for modelers when using stratified types. This places MLT* in a unique position, combining benefits of level-blind and level-adjuvant approaches (using the terminology from [Atkinson et al. 2014]).

Using MLT*, we have designed a text-based multi-level modeling language called ML2 [Fonseca 2017, Fonseca et al. 2018]. ML2 incorporates the rules of MLT* into its syntax, and an Eclipse-based editor that supports live verification of ML2 models ensures adherence to the theory. The use of a formally-verified semantic foundation is one of the distinctive features of ML2 when compared to other multi-level modeling approaches in the literature [Fonseca et al. 2018].

Our ongoing investigation concerning the theoretical aspects os multi-level modeling includes:

(i) the full integration of MLT* as a microtheory of the Unified Foundational Ontology (following up on [Carvalho et al. 2017]);
(ii) the study of the ontological interpretations for types (following up on some earlier work on this issue [Guizzardi et al. 2015]); and,
(iii) the study of the role of high-order types in ontology evolution.

Concerning the various applications, we intend to investigate in the near future:

(i) the incorporation of the UML profile for MLT into the OntoUML profile [Guizzardi 2015];
(ii) the development of an integrated constraint language to further increase the expressiveness of ML2;
(iii) the development of transformations from ML2 into the Semantic Web approach discussed in [Brasileiro et al. 2016a] and into the Alloy-based model simulation tooling we have built for OntoUML [Braga et al. 2010, Benevides et al. 2010]; and,
(iv) the investigation of suitable visual syntaxes to complement ML2’s textual syntax.

Acknowledgments. This work has been partially supported by CNPq (grants number 407235/2017-5, 312123/2017-5 and 312158/2015-7), CAPES (23038.028816/2016-41), FAPES (69382549) and FUB (OCEAN Project). Freddy Brasileiro, Claudenir M. Fonseca and Victorio A. Carvalho participated in the research discussed in this paper while affiliated with the Federal University of Espírito Santo. Victorio A. Carvalho and João Paulo A. Almeida would like to acknowledge fruitful discussion on this topic with participants of the Dagstuhl Seminar 17492 [Almeida et al. 2018].

References

Almeida, J. P. A., Fonseca, C. M., and Carvalho, V. A. (2017). A comprehensive formal theory for multi-level conceptual modeling. In *Conceptual Modeling*, pages 280–294. Springer.

Almeida, J. P. A., Frank, U., and Kühne, T. (2018). Multi-Level Modelling (Dagstuhl Seminar 17492). *Dagstuhl Reports*, 7:18–49.

Atkinson, C. and Gerbig, R. (2012). Melanie: Multi-level modeling and ontology engineering environment. In *Proceedings of the 2nd International Master Class on Model-Driven Engineering: Modeling Wizards*, MW '12, pages 7:1–7:2, New York, NY, USA. ACM.

Atkinson, C., Gerbig, R., and Kühne, T. (2014). Comparing multi-level modeling approaches. In *Proceedings of the 1st Workshop on Multi-Level Modelling co-located with the 17th ACM/IEEE International Conference MODELS 2014*, volume 1286 of *CEUR Workshop Proceedings*, pages 53–61.

Atkinson, C. and Kühne, T. (2000). Meta-level independent modelling. In *International Workshop on Model Engineering at 14th European Conference on Object-Oriented Programming*, volume 12, page 16.

Atkinson, C. and Kühne, T. (2001). The essence of multilevel metamodeling. In *International Conference on the Unified Modeling Language*, pages 19–33. Springer.

Atkinson, C. and Kühne, T. (2003). Model-driven development: a metamodeling foundation. *IEEE Software*, 20(5):36–41.

Benevides, A. B., Guizzardi, G., Braga, B. F. B., and Almeida, J. P. A. (2010). Validating modal aspects of ontouml conceptual models using automatically generated visual world structures. *Journal of Universal Computer Science*, 16(20):2904–2933.

Braga, B. F. B., Almeida, J. P. A., Guizzardi, G., and Benevides, A. B. (2010). Transforming ontouml into alloy: towards conceptual model validation using a lightweight formal method. *Innovations in Systems and Software Engineering*, 6(1):55–63.

Brasileiro, F. (2016). *Representation of multi-level domains on the web*. Federal University of Espírito Santo.

Brasileiro, F., Almeida, J. P. A., Carvalho, V. A., and Guizzardi, G. (2016a). Applying a multi-level modeling theory to assess taxonomic hierarchies in wikidata. In *Proc. 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 975–980. International World Wide Web Conferences Steering Committee.

Brasileiro, F., Almeida, J. P. A., Carvalho, V. A., and Guizzardi, G. (2016b). Expressive Multi-level Modeling for the Semantic Web. In *15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I*, pages 53–69. Springer.

Cardelli, L. (1988). Structural subtyping and the notion of power type. In *Proceedings of the 15th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '88, pages 70–79, New York, NY, USA. ACM.

Carvalho, V. A. (2016). *Foundations for Ontology-based Multi-level Conceptual Modeling*. PhD thesis, Federal University of Espírito Santo.

Carvalho, V. A. and Almeida, J. P. A. (2015). A semantic foundation for organizational structures: A multi-level approach. In *2015 IEEE 19th International Enterprise Distributed Object Computing Conference*, pages 50–59.

Carvalho, V. A. and Almeida, J. P. A. (2018). Toward a well-founded theory for multi-level conceptual modeling. *Software & Systems Modeling*, 17(1):205–231.

Carvalho, V. A., Almeida, J. P. A., Fonseca, C. M., and Guizzardi, G. (2015). Extending the foundations of ontology-based conceptual modeling with a multi-level theory. In *Conceptual Modeling*, pages 119–133. Springer.

Carvalho, V. A., Almeida, J. P. A., Fonseca, C. M., and Guizzardi, G. (2017). Multi-level ontology-based conceptual modeling. *Data Knowl. Eng.*, 109(C):3–24.

Carvalho, V. A., Almeida, J. P. A., and Guizzardi, G. (2014). Using reference domain ontologies to define the real-world semantics of domain-specific languages. In *Advanced Information Systems Engineering, 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014, Proceedings*, pages 488–502. Springer.

Carvalho, V. A., Almeida, J. P. A., and Guizzardi, G. (2016). Using a Well-Founded Multi-Level Theory to Support the Analysis and Representation of the Powertype Pattern in Conceptual Modeling. In *Advanced Information Systems Engineering, 28th International Conference, CAiSE 2016, Ljubljana, Slovenia, June 13-17, 2016. Proceedings*, pages 309–324. Springer.

Fonseca, C. M. (2017). *ML2: An Expressive Multi-Level Conceptual Modeling Language*. Federal University of Espírito Santo.

Fonseca, C. M., Almeida, J. P. A., Guizzardi, G., and Carvalho, V. A. (2018). Multi-level conceptual modeling: From a formal theory to a well-founded language. In *Conceptual Modeling*. Springer.

Foxvog, D. (2005). Instances of instances modeled via higher-order classes. In *Workshop on Foundational Aspects of Ontologies (FOnt 2005), 28th German Conference on Artificial Intelligence*, pages 46–54.

Frank, U. (2014). Multilevel modeling. *Business & Information Systems Engineering*, 6(6):319–337.

Gonzalez-Perez, C. and Henderson-Sellers, B. (2006). A powertype-based metamodelling framework. *Software & Systems Modeling*, 5(1):72–90.

Guizzardi, G. (2007). On ontology, ontologies, conceptualizations, modeling languages, and (meta) models. *Frontiers in artificial intelligence and applications*, 155:18.

Guizzardi, G. (2015). *Ontological Foundations for Structural Conceptual Models*. PhD thesis, University of Twente.

Guizzardi, G., Almeida, J. P. A., Guarino, N., and de Carvalho, V. A. (2015). Towards an ontological analysis of powertypes. In *Proceedings of the Joint Ontology Workshops 2015 Episode 1: The Argentine Winter of Ontology co-located with the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, July 25-27, 2015.*, volume 1517 of *CEUR Workshop Proceedings*.

Henderson-Sellers, B. (2012). *On the mathematics of modelling, metamodelling, ontologies and modelling languages*. Springer Science & Business Media.

Kühne, T. and Schreiber, D. (2007). Can programming be liberated from the two-level style: Multi-level programming with deepjava. In *Proceedings of the 22nd Annual ACM SIGPLAN Conference on Object-oriented Programming Systems and Applications*, OOPSLA '07, pages 229–244, New York, NY, USA. ACM.

Lara, J. D., Guerra, E., and Cuadrado, J. S. (2014). When and how to use multilevel modelling. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 24(2):12.

Mayr, E. (1982). *The growth of biological thought: Diversity, evolution, and inheritance*. Harvard University Press.

Mylopoulos, J. (1992). Conceptual modeling and telos. chapter 2. *Conceptual Modeling, Databases, and CASE*, pages 49–68.

Neumayr, B., Grün, K., and Schrefl, M. (2009). Multi-level domain modeling with m-objects and m-relationships. In *Proceedings of the Sixth Asia-Pacific Conference on Conceptual Modeling-Volume 96*, pages 107–116. Australian Computer Society, Inc.

Odell, J. (1994). Power types. *JOOP*, 7(2):8–12.

OMG (2017). OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.5.1.

Pan, J. Z. and Horrocks, I. (2006). OWL FA: a metamodeling extension of OWL DL. In *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, pages 1065–1066. ACM.