Appeared in **Software and Systems Modeling**
Vol. 21, pp. 543-559, DOI: 10.1007/s10270-021-00951-5

Multi-Dimensional Multi-Level Modeling

Thomas Kühne

Abstract The growth of multi-level modeling has resulted in an increase of level-organization alternatives which significantly differ from each other with respect to their underlying foundations and the well-formedness rules they enforce. Alternatives substantially diverge with respect to how level boundaries should govern *instance-of* relationships, what modeling mechanisms they employ, and what modeling principles they establish. In this article, I analyze how a number of multi-level modeling approaches deal with certain advanced modeling scenarios. In particular, I identify *linear domain metamodeling*, i.e., the requirement that all domain-induced *instance-of* relationships align with a single global level-hierarchy, as a source of accidental complexity. I propose a novel multi-dimensional multi-level modeling approach based on the notion of *orthogonal ontological classification* that supports modeling of domain scenarios with minimal complexity while supporting separation of concerns and sanity-checking to avoid inconsistent modeling choices.

1 Introduction

The primary aim of Multi-Level Modeling (MLM) is to reduce accidental complexity in domain models [10]. Formalizing domain-induced classification through a multi-level language/framework, however, also unlocks a further benefit in the form of supporting sanity-checking of MLM models by enforcing respective well-formedness constraints. The type level of traditional two-level approaches is usually not subject to any restrictions unless guidelines are manually checked or proto-metamodeling concepts like powertypes are used. The respective lack of control in two-level technologies opens the door for a number of undesirable modeling mistakes. Such mistakes range from creating models that are internally inconsistent to the omission of required concepts, e.g., when adapting frameworks. Brasileiro et al. have demonstrated that such mistakes are a real-world concern by showing that Wikidata taxonomic hierarchies suffered from inconsistencies to a significant extent [12]. They found that 15177 classes (85%) in Wikidata engaged in a so-called "Anti-Pattern 1" where combinations of classification and generalization relationships support unsound conclusions such as "*Tim Berners-Lee is a Profession*" [12, Fig. 3] (cf. Sect. 2.1). The problem with this

---

Thomas Kühne
Victoria University of Wellington, New Zealand
E-mail: thomas.kuehne@ecs.vuw.ac.nz

and other examples of the same nature is not the occurrence of simple modeling mistakes – in isolation most of the modeled relationships could be given sound interpretations – but that the combination of all claimed facts is not logically consistent and supports incorrect conclusions.

Multi-level modeling approaches are, in principle, better positioned than two-level approaches to prevent such mistakes but unless they feature suitable rules and mechanisms, they cannot capitalize on this potential. While some MLM frameworks [4,13] are designed to support the sanity-checking of models, i.e., to alert modelers whenever models are inconsistent or could give rise to paradoxes [6], and therefore could have alerted the authors of the aforementioned Wikidata models to the problems with them, these frameworks have been criticized for their rigidity [18] and more flexible frameworks have been proposed in response. The latter aim to avoid redundancies sometimes caused by "strictness" [4] enforcing frameworks [18,29,17] and cope with advanced modeling scenarios that present challenges to frameworks imposing strictness rules in order to support some level of sanity-checking [36,1,16].

On the one hand, such flexible frameworks support concise models, with minimal accidental complexity, even when faced with certain challenging modeling scenarios that necessitate the use of complexity-introducing workarounds when using strict frameworks. On the other hand, a relaxation of well-formedness constraints prevents certain nonsensical modeling choices from being detected. In this article, I therefore analyze a number of approaches, point out potential issues, and present a novel approach, named *orthogonal ontological classification*, which handles challenging modeling scenarios without introducing accidental complexity, while still retaining the sanity-checking capabilities of strict frameworks.

2 Background

Common among all multi-level modeling approaches is the use of multiple levels to organize ontological model content which represents some domain of interest. However, the approaches do not agree on

- the precise nature of levels,
- how they should relate to one another, and
- what restrictions should be in place to prevent modelers from creating ill-formed models.

In the following, I cover all major kinds of multi-level approaches along with their level organization and well-formedness schemes – ordered by the level of flexibility they provide – in order to establish the landscape in which problematic multi-level modeling issues occur.

2.1 Strict Metamodeling

One of the first well-formedness schemes proposed for multi-level hierarchies is *strict metamodeling* [4]. It restricts all element relationships to be intra-level relationships with the exception of *instance-of* relationships which connect model elements between adjacent levels. It thus implies that neither associations nor inheritance relationships can cross level boundaries and that no *instance-of* relationships may cross more than one level boundary.

This regime deems models such as the one underlying the "*Tim Berners-Lee is a Profession*"-scenario mentioned in the introduction as ill-formed, i.e., not being in compliance to its well-formedness rules, and is therefore capable of flagging such nonsensical models as being problematic to a modeler. Consider the left-hand side of Fig. 1 which shows an abridged version of the Wikidata information as it was illustrated in [12, Fig. 3]. The *instance-of* relationship between Tim Berners-Lee and Profession is not part of the original model but is derivable as follows: Tim Berners-Lee is declared to be a Scientist and due to the subtype relationship between Scientist and Profession (note that Scientist is modeled to be an *instance-of* Profession as well as a subtype of Profession), every scientist is also a profession. Therefore, Tim Berners-Lee is also (indirectly) an *instance-of* Profession, according to this model.

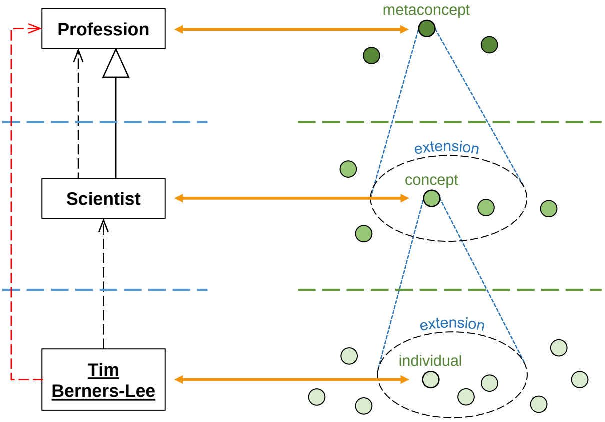


Fig. 1 Ill-formed model (as reported by [12, Fig. 3])

The right-hand side of Fig. 1 illustrates why the above conclusion is not only counter-intuitive but why the model can be detected as being internally inconsistent despite being syntactically correct. The model element Tim Berners-Lee represents an individual, which is depicted as a dot on the right hand side of Fig. 1. In the domain shown at the bottom level of Fig. 1 two further individuals are also scientists which is why they both are also members of the “scientists” set. This set is the extension of a concept residing at a level above, i.e., the concept that is represented by modeling element Scientist. The scientist concept itself is a member of the set of professions, along with one more unnamed concept (see the middle of Fig. 1). Finally, the “professions” set is the extension of the metaconcept which is represented by the modeling element Profession.

The derivable *instance-of* relationship between Tim Berners-Lee and Profession (see the left-hand side of Fig. 1) claims that the individual represented by Tim Berners-Lee is a member of the professions set which is at odds with the fact that the professions set only has sets as its members as opposed to individuals. In other words, there is no sound set-theoretic interpretation of the model on the left-hand side, revealing that the model in Fig. 1 is internally inconsistent.

The *strict metamodeling* doctrine can determine the model to be ill-formed as it deems the use of specialization between elements at different levels as illegal. Note that if the specialization relationship between Profession and Scientist were removed, the model would no longer allow an incorrect inference to be made, as the *instance-of* relationship between Tim Berners-Lee and Profession would no longer be derivable.

Furthermore, note that the presence of an *instance-of* relationship between Scientist and Profession unambiguously places Profession at a higher level than Scientist. If this relationship were removed and Profession would be considered to reside at the same level as Scientist then the counter-intuitive inference of Tim Berners-Lee being an *instance-of* Profession would be possible but the framework would be unable to flag the model as being problematic. This is the result of the framework being ignorant about the implied meaning of names like “Profession”. If the same concept had been named “Person” then no counter-intuitive inference would be possible. In other words, the framework relies on cues like *instance-of* relationships or the manual assignment of elements to levels in order to infer their location in a classification hierarchy. Only by exploiting knowledge about the latter, can the framework make a determination about the logical consistency of a model.

Since the type of relationships that can cross level boundaries is severely restricted in frameworks adhering to *strict metamodeling* and related schemes (e.g. [13]), the above described ability to detect certain ill-formed models can be easily implemented by validating the static semantics of models, i.e., by running well-formedness checks. Employing *strict metamodeling* is hence comparable to using statically-typed programming language as opposed to dynamically typed ones, however, without incurring the cost of requiring additional (type-) annotations. Even without tool support, though, it becomes trivial for a modeler to spot level-crossing relationships which could give rise to inconsistencies, provided that a visualization of all levels and relationships is available.

Unfortunately, *strict metamodeling* also excludes modeling scenarios that are not problematic [7, 18, 36] and has been criticized as requiring workarounds and causing redundancy in order to force commonly occurring scenarios into compliance with strictness requirements [18, 29, 17].

2.2 Relaxed Strictness

Gitzel et al. [18] and de Lara et al. [29] relax the rules of *strict metamodeling* by enabling *level-jumping*, i.e., supporting *instance-of* relationships that cross more than one level boundary. They are thus able, among other things, to avoid some level-crossing associations by allowing elements to reside at the same modeling level, which would have been forced to reside at different levels under a *strict metamodeling* regime.

2.3 Orderless Typing

Carvalho et al.’s MLT approach [13] is arguably even more restrictive than *strict metamodeling* with respect to *instance-of* relationships as it is less flexible about the location at which elements reside within the level-hierarchy [27]. In contrast to the *strict metamodeling* adhering *deep modeling* [9], where elements representing individuals may reside at any level, MLT assigns absolute positions to elements based on their order, i.e., their set-theoretic classification power defined by the depth of *instance-of* relationships they can support.

However, Almeida et al.’s MLT* extension introduces flexibility through the use of so-called *orderless types* to support the modeling of scenarios in which multiple elements at different levels need to be classified by types such as Entity, SocialEntity, and Business Asset [1, 16]. As a result, MLT* not only supports the flexible level-crossing associations of MLT [13], but also allows non-strict instantiation via orderless types [1].

2.4 Transitive Levels

A number of approaches that use multiple levels for organizing elements could be regarded as being even more flexible than MLT*. Neumayr et al.'s *M-objects* [35] and Macías' *MultEcore* [31] allow levels to be inserted or removed without invalidating any other levels. Henderson-Sellers et al. similarly do not use levels to achieve a classification-based stratification but rather view levels as packages inspired by organizational concerns, i.e., with level contents driven by which agents are associated with them [19].

2.5 Level Blindness

For a complete picture of multi-level modeling frameworks, those which do not assign any significance to levels should be mentioned as well. Henderson-Sellers et al. view every element as an "object" [22,14] and employ a meta-circular approach. No attempt at enforcing soundness via *instance-of* relationships is made, thus opening up the door to unsound models and even paradoxes [6].

3 Analysis

In the following, I analyze the aforementioned approaches as to how and why they differ in the design choices I discussed and as to what their respective limitations are. The analysis is conducted with a view to ultimately arrive at an approach that supports sanity-checking of multi-level models to the same extent as *strict metamodeling* is able to, i.e., reject ill-formed models, like the one in Fig. 1, that lack internal consistency by virtue of not having a sound set-theoretic interpretation, while avoiding the complications induced by *strict metamodeling* when facing more challenging scenarios. In other words, the aim is to support the modeling of advanced scenarios as straightforwardly as MLT* [16] without forgoing the model-rejection abilities of *strict metamodeling*.

3.1 Relaxed Level-Organization Principles

The approaches mentioned in Sect. 2.4 use a rather different notion of *level* compared to the rest of the approaches mentioned in Sect. 2, the latter of which all employ *classification* as their *level-segregation principle* [27]. A survey conducted ahead of Dagstuhl seminar 17492, suggests that these transitive level approaches should still fit under the "multi-level modeling" umbrella since 16 participants (out of 18 respondents) indicated in their response to the question "*What is multi-level modeling?*" that the segregation principle used in multi-level modeling is *abstraction* [26]. Fewer participants (10), phrased their definition in such a manner that allowed interpreting *classification* to be the abstraction principle, indicating that abstraction principles other than classification appear to enjoy some acceptance in the multi-level modeling community.

However, as it is my goal to support sanity-checking of multi-level models, I will not further consider transitive level approaches in the remaining discussion since they are inherently less suited to uncovering ill-formed structures. Their purposefully introduced additional degrees of freedom, forgoing rigid level structure and/or order-based well-formedness schemes in order to further flexibility and minimize redundancy, make it impossible to syntactically detect ill-formed structures of the kind discussed in Sect. 2.1.

To a lesser extent, this argument also applies to Gitzel et al.'s relaxed strictness definition [18]. It excludes non-*instance-of* relationships from crossing level boundaries (and would as such reject the model in Fig. 1), but allows *instance-of* relationships to cross more than one level. It associates levels with "domain-specificity" and, due to redundancy considerations, rejects any kind of identity/phantom elements (cf. [29]) which could restore strictness. While this scheme obviously allows sound models, not all models that can be detected as being unsound by enforcing *strict metamodeling* are excluded. For example, relaxed strictness would allow an unsound explicit *instance-of* relationship between Tim Berners-Lee and Profession (cf. Sect. 2.1).

3.2 Modeling Context

Strict metamodeling (cf. Sect. 2.1) was originally formulated to manage language definition layers that exert control via linguistic classification. In such a linguistic context, the rather restrictive rules of strict metamodeling are entirely justified since the latter ensure that each layer in a level definition stack forms a cohesive entity which is fully defined by the layer above it, without losing any required expressiveness.

However, approaches such as *deep modeling* [9], that adopted the *strict metamodeling* doctrine for the ontological classification dimension, find themselves in the position that they either cannot accommodate the kind of advanced modeling scenarios that orderless typing can handle without employing complexity-inducing workarounds (cf. Sect. 2.3) or need to adjust their rule sets. Atkinson and Kühne proposed so-called *modeling spaces* [7] to account for level-crossing associations [36] but the resulting framework still cannot cope with scenarios that call for orderless types [1]. Modeling spaces only provide a sound view on non-strict associations between levels, but do not account for the simultaneous classification of elements that reside at different levels (cf. Sect. 2.3).

3.3 Superficial Differences

Some forms of *level-jumping*, i.e., *instance-of* relationships crossing two or more level boundaries, can be regarded as being compliant even to *strict metamodeling* when recognizing so-called *phantom elements* which are considered to be invisible but are assumed to be present conceptually [29]. The Metadepth approach by de Lara et al. [29], which introduces concepts such as *leap potency* and *deep references*, can avoid soundness issues, such as the one illustrated by Fig. 1, by postulating *phantom* model elements that allow deep properties to be passed down to the required level without breaking strictness rules. On the one hand such implicit, mediating model elements can be regarded as affording considerable convenience, but on the other hand, they may prevent the modeler from recognizing certain misconceptions. Scenarios such as Tim Berners-Lee being a direct *instance-of* Profession (without involving Scientist) rely on the modeler to understand and verify that a valid conceptualization of an intermediate *phantom element* must exist.

As a result, while the apparent "level jumps" do not necessarily violate the rules of *strict metamodeling*, the respective frameworks do not support the mechanical recognition of all issues detectable by the use of *strict metamodeling*, since *leaps* may or may not have a sound interpretation. Furthermore, level-jumping cannot deal with certain modeling challenges that are described in the next section.

3.4 Order-Synchronization vs Order-Alignment

Schemes like MLT that prescribe an absolute placement of elements according to their order within a level-hierarchy can be said to utilize *order-synchronization* [27]. The order of an element is its set-theoretic classification power, i.e., the depth of *instance-of* relationships it can support. For example, objects (elements that cannot have instances) must reside at the bottom-most level in an order-synchronized hierarchy. In contrast, so-called *order-aligned* schemes may place objects anywhere and only require relative order differences between elements to align with the level-hierarchy [27].

These level-alignment alternatives naturally result in differences regarding whether associations have to cross level boundaries and how many level boundaries need to be crossed by *instance-of* relationships. De Lara et al. showed that some level-jumping can be avoided by moving down an element so that it resides one level above the element it classifies [29]. Similarly, Atkinson and Kühne showed that certain level-crossing associations can be avoided in an order-aligned scheme by moving elements up or down accordingly [7], exploiting the flexibility of order-alignment over order-synchronization.

However, in some advanced modeling scenarios (cf. Fig. 2) level-crossing associations cannot be avoided simply by performing hierarchy alignments. Atkinson and Kühne had to extend the strict metamodeling doctrine to include *modeling spaces* because strictness violations cannot be avoided by simply shifting elements such as Jony in Fig. 2 up or down.

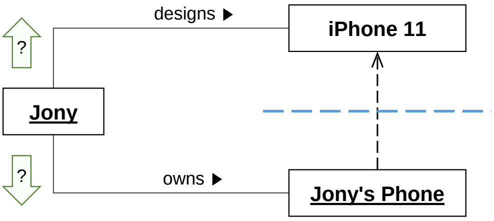


Fig. 2 Non-strict associations

Yet, as alluded to in Sect. 2.3 already, there are further advanced modeling scenarios that even *modeling spaces* cannot account for. Hence, the only scheme capable of maintaining a strictness regime over the *majority* of elements (those classified by ordered types) while supporting all currently known advanced modeling scenarios is the MLT* approach [1].

3.5 Remaining Issues

Unfortunately, MLT* is not a complete solution because its orderless types re-enable the very inconsistencies that ordered types are designed to prevent. MLT (which does not feature orderless types) strictly separates first-order types from second-order types, etc. and thus avoids potential inconsistencies that arise from mixing elements of different order. MLT* introduced orderless types for legitimate purposes, but unfortunately has no provision to prevent illegitimate usages. For instance, if a modeler wanted to (inconsistently) declare Tim Berners-Lee to be an instance of Profession as well as Scientist (cf. Fig. 1), they could simply declare Profession to be an orderless type, in order to circumvent the safety mechanism established by ordered types. The result, however, would be a model that does not appropriately represents the intended domain (cf. Sect. 2.1).

Furthermore, consider Business Asset in Fig. 3 (cf. [16, Fig. 5]) which needs to be an orderless type as it is both the type for an order-0 object (Jony's Phone) and an order-1 type (iPhone 11), and hence cannot be consistently assigned an order itself [1]. Following the convention used by Almeida et al. [1], in Fig. 3 the orderless type, for which the regular restrictions on *instance-of* relationships are suspended, is placed above the highest-order instance it classifies. Note, however, that in only slightly different scenarios where either of the two other concepts in Fig. 3 or just their interconnecting *instance-of* relationship is missing, Business Asset would be modeled as an ordered type. *Orderlessness* is hence not an intrinsic quality of a concept but rather depends on specific application contexts. Maintaining the notion that *orderlessness* is implied by the domain rather than imposed by hierarchy alignment challenges, would require arguing that concept Business Asset in Fig. 3 is not the same as

1. a Business Asset concept that classifies only one of the instances in Fig. 3, and not the same as
2. a Business Asset concept that simultaneously classifies variants of iPhone11 and Jony's Phone when both of the latter happen to reside at the same level (cf. Fig. 4),

since in both of the above cases, Business Asset would just be be a regular ordered type.

An orderless type like Business Asset poses an additional problem with respect to *deep characterization* [9]. On the one hand, Jony's Phone is meant to be directly classified by Business Asset and accordingly receives a value feature. On the other hand, it would also seem possible for Jony's Phone to be deeply characterized by Business Asset through *deep features* (with potency > 1). The fact that a single element could simultaneously be the target of both shallow and deep characterization suggests that the overall configuration is problematic.

Observe that if we considered level boundaries, as implied by the *instance-of* relationships between Jony's Phone, iPhone 11, and Business Asset, the *instance-of* relationship between Jony's Phone and Business Asset would cross two level boundaries, i.e., would be "level-jumping", or *non-strict*. Given the aforementioned issues, it seems, once again, that non-strictness is a good indicator that closer investigation is warranted, even when the non-strictness is technically averted by declaring Business Asset to be *orderless*.

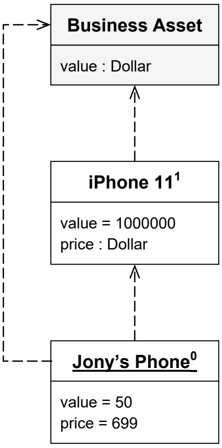


Fig. 3 Orderless Typing

Due to the above issues, despite being currently the most flexible and expressive approach, orderless typing cannot be considered to be a fully satisfactory solution for retaining sanity-checking that covers all elements when dealing with advanced modeling scenarios.

4 Solution

To address the aforementioned issues, I propose an approach that retains the ability to detect inconsistent models in the spirit of *strict metamodeling* but also copes with advanced modeling scenarios involving level-crossing associations [36] and seemingly unavoidable non-strict *instance-of* relationships [1]. I build on the following two insights:

1) Level-crossing associations do not present a consistency problem outside the context of linguistic language definition hierarchies.
2) Level-jumping *instance-of* relationships can be a symptom of aligning inherently unrelated ontological classification relationships within a single linear level-hierarchy.

Level-crossing associations have been described as *metabombs* which collapse a level-hierarchy [7]. While this view is justified for linguistic level hierarchies, it does not apply to ontological hierarchies. Ontological classification has been shown to follow different rules to linguistic classification [11] and with respect to sanity-checking, the only concern is whether a multi-level model represents a domain such that it has a consistent set-theoretic interpretation. As a result, outside language definition hierarchies, there is no need to ensure that connectors between elements are either pure associations (connecting types) or pure links (connecting instances) (cf. Sect. 6.2) [7]. For example, in the OCA [8], one would define a linguistic type Connector which would support the connection of elements at any level, without requiring the participants to reside at the same level.

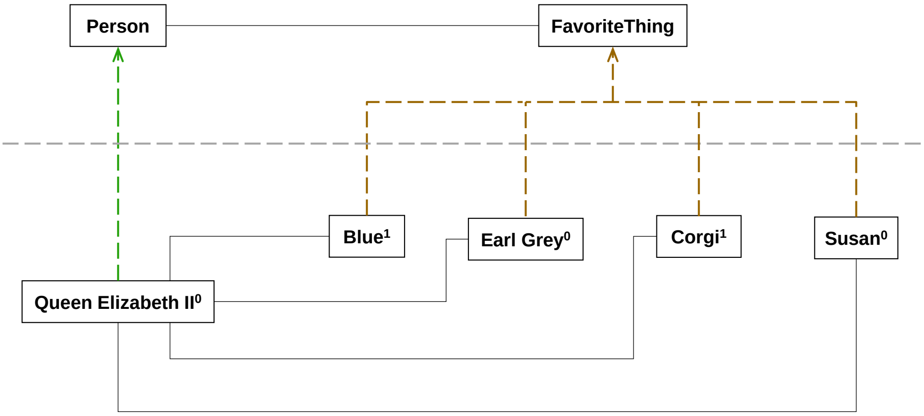


Fig. 4 Flat perspective on entities of different order

Associations/connectors can be considered to be independent relations that reference other modeling elements. No inconsistencies are introduced if the elements referenced from different association ends have nonmatching orders. Consider the Queen's favorite things shown in Fig. 4. They include individuals, like the dog Susan but also universals like the dog breed Corgi. Relationships as those shown in Fig. 4 are entirely unproblematic, since no logical inconsistencies can be introduced by connecting elements of different order. While orthogonal ontological classification inherently supports minimizing level-crossing associations (cf. Sect. 4.1) via its adoption of *order-alignment*, there is no formal need to rule out level-crossing associations.

Regarding insight 2) above, Almeida et al. state that there are types that "*defy stratification into levels*" [1] (cf. Fig. 3). My analysis differs and posits that so-called "*orderless*" types do not inherently lack order, but that the concept of a single order within a global single-dimension level-hierarchy is at odds with orthogonal domain classification as it naturally occurs in domains of discourse. In other words, I identify a *dimension misalignment* – i.e., a clash of two or more incompatible level hierarchies which are forced to align within a single linear scheme – instead of types that inherently defy an order assignment.

Note that if the instances of orderless types like Business Asset did not already participate in another classification hierarchy, they could simply all be regarded as instances which reside one level below Business Asset and the latter could be made an ordered type. Only the overlapping of two classification concerns creates a tension that orderless types seek to resolve via a local relaxation of classification well-formedness rules.

4.1 Orthogonal Ontological Classification

My proposed solution is to allow multiple *orthogonal* classification hierarchies to be defined and interrelated. Figure 5 shows how the scenario of Fig. 3 is captured by a *multi-dimensional* multi-level framework, making use of *orthogonal ontological classification*.

Note that all elements in Fig. 5 continue to have the same features, but there are a number of important differences to Fig. 3:

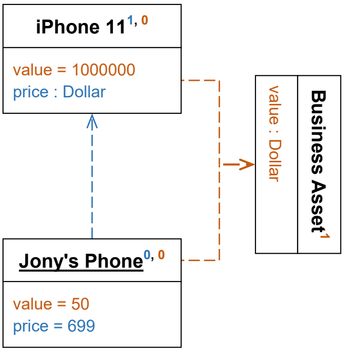


Fig. 5 Orthogonal Classification

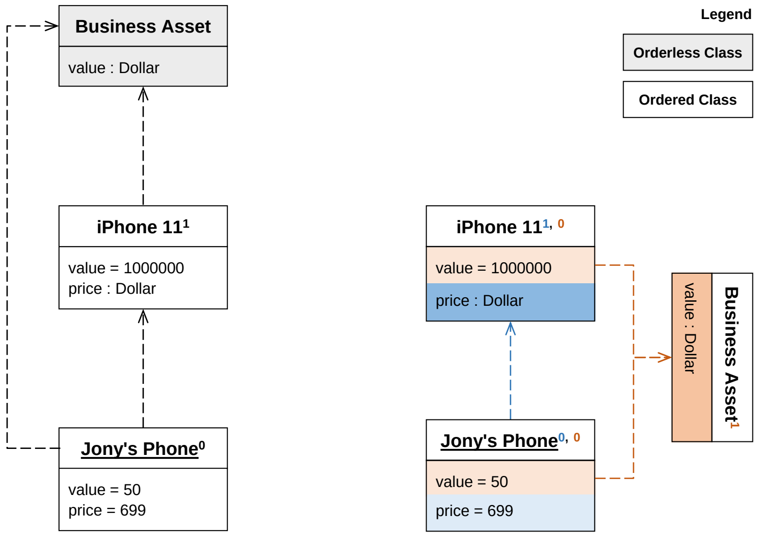
D1 *Reestablished order*: Business Asset is now an order-1 type. By escaping the confines of a linear level-hierarchy, it is now possible to treat the multiple classification of Jony's Phone as occurring from two orthogonal dimensions, instead as from one ordered type (iPhone 11) and one orderless type (Business Asset) (cf. Fig. 3), both requiring alignment in a single linear classification hierarchy. This crucially re-enables sanity-checking in advanced scenarios which hitherto required non-strict treatment. Note that strictness is restored locally for each classification dimension.

D2 *Deep characterization*: Unlike orderless typing, the approach supports consistent deep characterization for all classification concerns. It would be possible for Business Asset to be classified by an element that deeply characterizes all business asset instances. This is a consequence of avoiding the incongruous classification path lengths that exist in the orderless typing scenario (cf. Fig. 3). Fig. 6 makes it easy to recognize that the orderless modeling variant (Fig. 6 (a)) suggests a distance of two between Jony's Phone and Business Asset, plus a shortcut instance / metatype relationship which is absent in the orthogonal ontological classification modeling variant (Fig. 6 (b)).

D3 *Separation of concerns*: Placing Business Asset into an orthogonal *business asset* classification concern, which is independent from the *phone* concern, confers two order values upon phone elements. They have their original phone orders within the *phone* concern (1 and 0, respectively, cf. Fig. 3), but simultaneously are order-0 elements with respect to the *business asset* concern. Note that this separation allows phone element features to be organized according to their origin as indicated by the coloring used in Figs. 5 and 6.

D4 *Inconsistency detection*: If a new feature is to be added to Jony's Phone then a modeler has to deliberate about whether the respective feature declaration belongs to the *phone* concern or to the *business asset* concern. This helps modelers to be clear about what the meaning of say price vs value is. In particular, should a modeler be tempted to create an *instance-of* relationship between Tim Berners-Lee and Profession (cf. Sect. 3.4), the modeler will:

1. receive feedback that such a shortcut is unsound, as Profession is in the same classification dimension as Tim Berners-Lee and Scientist, and
2. if insisting on the change, be forced to move Profession into a different classification dimension. In this case, the modeler will be facing the fact that Profession then no longer has a classification distance of two to Tim Berners-Lee and hence would be required to reconceptualize/rename Profession.



(a) Orderless Typing

(b) Orthogonal Ontological Classification

Fig. 6 Linear vs Multi-Dimensional Classification

The addition of an *instance-of* relationship between Tim Berners-Lee and Profession can be recognized as unsound because it would not observe the *level-respecting* property [25] that the approach enforces for each classification dimension respectively (cf. Sect. 4.3). Corresponding feedback will either cause the modeler to realize that the intended new classification relationship was the result of erroneous thinking or force the modeler to reconsider the previously existing element structure.

As a result, if orthogonal ontological classification is used rather than orderless typing, there is a significant difference in the feedback a modeler receives when attempting to establish certain unsound relationships.

4.2 Non-Overlapping Orthogonal Classification

Figure 5 demonstrates the important case when two separate classification concerns overlap with respect to some modeling elements. However, there are also cases of non-overlapping concerns that are nevertheless best captured by not forcing them into a single linear level-hierarchy.

Consider Atkinson and Kühne's *processes and products* example [7, Fig. 12 & Fig. 15]. Figure 7 recreates the spirit of their example and adds a Product concept that orthogonally classifies elements that are members of a *phone* concern. Note that the *process* hierarchy on the left and the *phone* hierarchy on the right do not share any elements with respect to overlapping classification. Without Product, Fig. 7 would not feature any overlapping concerns.

I added Product in order to demonstrate that the links between Sam Validating and the phone elements Jony's Phone and iPhone 11 can be understood as instances of the validates association between DesignActivity and Product. Fig. 8 illustrates the respective elements from

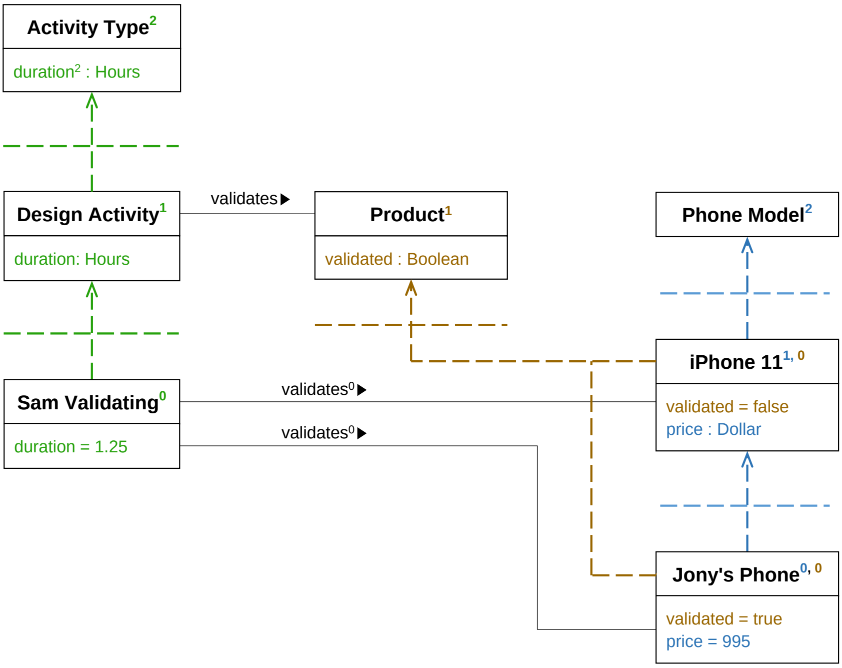


Fig. 7 Overlapping and Disjoint Classification Concerns

Fig. 7 using a 3D perspective to visually emphasize this dimension-linking aspect of orthogonal ontological classification. Note how the initially problematic lack of alignment of links – Sam Validating connects to two elements which are of different order – can thus be cleanly understood as resulting from an orthogonal *product* perspective on the *phone* concern. The Product concept thus acts as an explanatory component that illuminates how the process and products hierarchies can be understood to interact cleanly.

The fact that at least one of the links between Sam Validating and the phone elements would traditionally be considered *non-strict*, prompted Atkinson and Kühne to propose *modeling spaces* in order to separate the *process* hierarchy on the left from the *phone* hierarchy on the right with the intention to create separate level hierarchies and hence address the non-strict nature of the links connecting processes with their subjects.

In contrast, using my proposed multi-dimensional multi-level approach, i.e., orthogonal ontological classification, the links between processes and their subjects would not be considered to be problematic to begin with. Moreover, orthogonal ontological classification allows the fact to be recognized that the links can be understood to connect elements of the same order after all (here order 0), as long as the elements are viewed from the orthogonal *product* perspective.

4.3 Suggested Well-Formedness Constraints

Orthogonal ontological classification is an approach that could potentially be realized in a number of forms. In the following, I present one particular design variant which somewhat limits expressiveness in favor of straightforwardness.

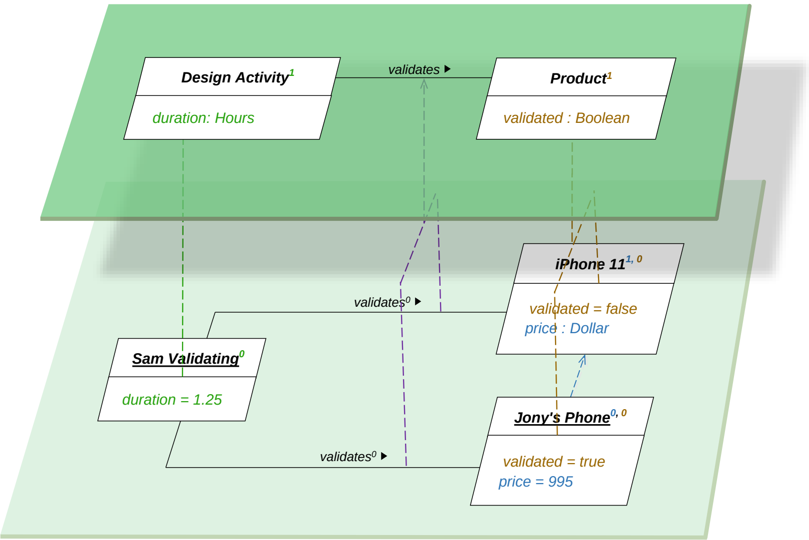


Fig. 8 Orthogonal Concerns

C_1 *Disjoint feature sets*: The feature sets of classifiers interacting in overlapping classification must be disjoint. It is conceivable that, e.g., a feature like *value* may be used in multiple classification dimensions (e.g., in both *phone* and *business asset* concerns), however in order to avoid having to introduce a disambiguation mechanism, i.e., requiring access to properties to be performed with fully qualified names, I currently require renaming (e.g., to *business value* in the business concern) in such cases.

C_2 *Bottom-level overlapping*: Classification concerns may only overlap if, at the overlapping element, at most one of the concerns has a *potency* greater than zero. There is no need to require an *order* value of zero, since I only require the inability to spawn further instances from all but at most one classification concern. It is conceivable to have deep classification concerns overlap at elements with multiple potency-1 or higher potency elements. However, this would necessitate a mechanism to specify into which classification dimension one wishes to instantiate such mixed elements.

C_3 *Connected classification clusters*: The *instance-of* relationships within one classification concern must form a tree. This restricts classification concerns to tree-shaped *instance-of clusters*. At this early stage of investigation, I cannot rule out the possibility that more relaxed forms may be meaningful but the required semantics are currently unclear. Enforcing tree-shaped clusters ensures that there is always a consistent bottom-up interpretation for orthogonal classification.

C_4 *Sound meta-hierarchies*: Within each classification cluster I require the *level-respecting* and *acyclic* constraints for meta-hierarchies [25] to hold. Furthermore, subtype relationships may only be used within a level and between elements of the same order. In combination these constraints, in addition to the usual intra-level constraints such as acyclic specialization hierarchies, ensure that there is a sound set-theoretic interpretation of all elements and their relationships. In particular, they eliminate the modeling scenario that enables the "*Tim Berners-Lee is a Profession*" inference [12, Fig. 3]. Note that the aforementioned constraints subsume the soundness criteria that strict metamodeling implies for *instance-of* relationships. Orthogonal ontological classification can therefore be understood as preserving *local strictness*; it simply forgoes the ambition to fit every *instance-of* relationship into one global linear classification hierarchy.

Furthermore, note that C_4 not only supports the detection of Brasileiro et al.'s "Anti-Pattern 1" but all of the three anti-patterns Brasileiro et al. considered [12]. In fact, an unbounded number of inconsistent modeling scenarios is ruled out by the such established sanity-checking because the latter is based on ensuring a sound set-theoretic interpretation of the model as opposed to detecting a finite number of known problematic patterns.

5 Discussion

The proposed orthogonal ontological classification approach is a particular case of multiple classification. Unlike traditional multiple classification, e.g., as occurring in the classic *amphibious vehicle* example (cf. Sect. 6.1), it does not require all overlapping classification concerns to have the same order at the overlapping element, though. As a matter of fact, orderless typing [1] can also be regarded as a form of multiple classification since an element may be classified by an ordered type and an orderless type at the same time. However, Almeida et al.'s MLT* approach differs from orthogonal ontological classification in three important ways:

1) order, and therefore the ability to perform order-based sanity-checks, is abandoned for orderless types. While there are valid applications for orderless types, orderless types may also be used to create scenarios which have no sound set-theoretic interpretation. Such orderless types could be considered to be lacking cohesion, as they would classify entities of different order within a single classification dimension. In contrast, orthogonal ontological classification only allows multiple classification that is truly orthogonal, i.e., where the order of an element in one dimension is meaningless with respect to the order of the element in other dimensions.
2) there is one distinguished hierarchy of ordered types and all other classification concerns that do not align with it are represented within one "*orderless*" category. Within the latter category there would be multiple secondary "roots" organizing the respective concerns, so there would typically be one primary ordered classification and many secondary orderless classficiations. In contrast, orthogonal ontological classification allows all classification concerns to be considered as having equal importance.
3) modelers have to make decisions as to whether the use of orderless types is warranted on a case by case basis, potentially converting previously ordered types into orderless ones, and vice versa, depending on the modeling context.

5.1 Reduction of Accidental Complexity

In addition to supporting sanity-checking of models, due to its *separation of concerns* quality, orthogonal ontological classification also serves the primary purpose of multi-level modeling of reducing accidental complexity [10]:

1) It avoids complex "level-jumping"-involving scenarios which are the result of shoehorning multiple independent classification concerns into a single linear level-hierarchy. MLT* technically places orderless types outside the linear hierarchy implied by ordered types but the *instance-of* relationships pointing to orderless types do not align with the hierarchy and all published MLT* applications place the orderless types relative to the ordered ones, effectively resulting in level-jumping *instance-of* relationships.

2) Orthogonal ontological classification does not require additional mechanisms such as *leap potency*, *identity instantiation*, *orderless* vs *ordered* types, etc. These mechanisms, if present, not only need to be mastered but the level-jumping they support blurs the difference between genuine two-level distances and artificially stretched but actual one-level distances. Multi-dimensional classification is undoubtedly richer than single-dimension classification, but orthogonal ontological classification simply *uniformly* applies the same principle (i.e., *domain-induced classification*) in multiple dimensions without introducing additional mechanisms within a dimension.

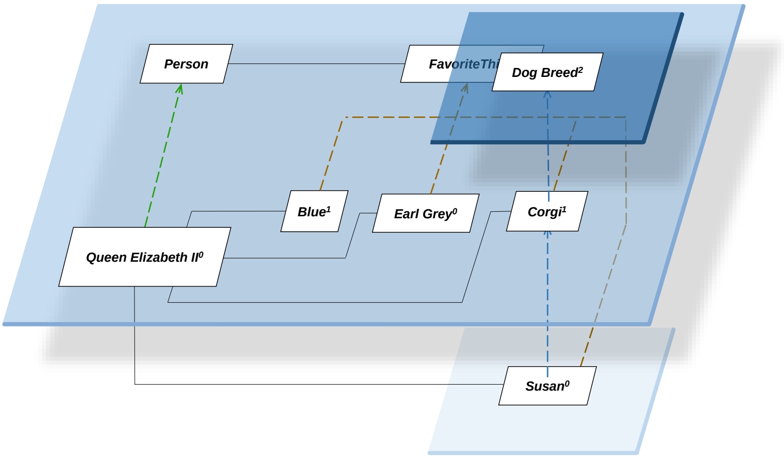


Fig. 9 Recognizing orthogonal classification dimension

3) There is no need to nominate one classification dimension as *the* primary one which then implies level boundaries for all other classification dimensions. Singling out a dominant classification concern over others may have significant consequences, in particular in order-synchronized schemes. Fig. 9 shows an augmented version of Fig. 4, elaborating the fact that Corgi may have an instance (Susan) and a type (Dog Breed). The other elements and their relationships remain unaffected by this recognition of an additional classification dimension. However, we could have equally started with the Susan, Corgi, and Dog Breed hierarchy and then added the Favorite Thing hierarchy as a secondary classification dimension, i.e., no dimension dominates the other.
Note that while Fig. 9 uses a 3D-perspective for clarity, diagrams would normally be arranged using a 2D-layout, as employed by Fig. 7. The 3D-perspective simply helps to visualize that there are no non-strict instantiation path lengths differences between Susan, Corgi, and Favorite Thing. In a flat, uni-colored rendering that does not visualize the orthogonal dimensions in any way, it would appear that there is both a one-step connection between Susan and Favorite Thing (reflecting Susan's capacity as a favorite thing), and a two-step connection (via Corgi, which is also a favorite thing). In MLT*, this would have forced Favorite Thing to become an orderless type. In contrast, orthogonal ontological classification recognizes that the connection between Susan and Favorite Thing via Corgi changes dimensions (from the *dog* dimension to the *favorite thing* dimension), i.e., does not create a dilemma regarding Favorite Thing's order.
4) Concepts like Business Asset receive a stable interpretation, i.e., they are not subject to reconsideration depending on which instances at what level are associated with them. In

contrast, MLT* makes the *ordered* vs *orderless* status of Business Asset depend on where the latter's instances reside. In other schemes (cf. Sect. 2), the placement of an element within the type hierarchy depends on what associations it has with other elements, i.e., is not solely determined by classification hierarchies.

I maintain that an element kind, such as *ordered* vs *orderless*, and the placement of an element within a level-hierarchy, should depend on intrinsic qualities of the element based on an ontological anchoring, as opposed to potentially circumstantial usage contexts which are subject to change. An ontological anchoring provides far more stability as it mirrors immutable logical classification relationships in the domain. As long as the domain does not require extensive reinterpretation, the ontological classification relationships in the model will remain stable.

5.2 Dynamic Classification

Interestingly, in the example of Fig. 7, the *phone* concern appears to suggest *players* (Jony's Phone & iPhone 11) that are suitable for a product *role* [40]. This view could be realized by allowing the *players* to enter and leave the Product classification dynamically. Another classic dynamic classification dimension is established by so-called phases which elements go through during their lifetime. However, note that incorporating such dynamic classifications would imply asymmetric dimensions with some of them representing rigid and others non-rigid or anti-rigid classification dimensions [20]. Currently, orthogonal ontological classification is not designed to systematically distinguish and support such classification kinds, however it gives modellers control over which element aspects are essential. Essential features can be enforced via deep characterization specified at the root of a classification dimension whereas optional features can be mixed in at various depths of orthogonal dimensions. To the best of my knowledge, orthogonal ontological classification is the first approach to offer modelers such a choice in multi-level modeling in combination with sanity-checking.

5.3 Deep Characterization

Some usages of deep characterization show that a very specific way of classifying a number of elements at different classification levels is possible, including feature specification (e.g., price for products), using a combination of deep instantiation and features with a *durability* that lets them span multiple levels [28]. However, this only works if two classification concerns unfold in parallel and a linear fashion. Furthermore, for instance in the case of *Melanee* [5], this approach implies that elements belonging to a *product* concern would be distributed over a number of classification levels, i.e., they would not be members of a single "Product" set (cf. Fig. 7).

In contrast, orthogonal ontological classification allows one to individually select elements for classification. For example, it is possible to only chose a subset of the phone elements to be business assets. If the property of being a business asset were conferred via deep characterization then every phone instance would have to be a business asset without exception. Summarizing, orthogonal ontological classification uniquely

- allows classification concerns to orthogonally overlap,
- does not require unnecessary, and at times impossible, alignment of concerns, and
- enforces local, rather than global, well-formedness checking, comprehensively covering all elements as opposed to a subset of (ordered) elements only.

6 Related Work

A considerable amount of related work has already been referenced in Sections 2, 3, and 5. In the following, I discuss further work including approaches that do not necessarily contribute to multi-level modeling per se but are nevertheless related.

6.1 Traditional Multiple Classification

Multiple classification as such is not a novel concept. The UML acknowledges it in the form of allowing generalizations sets to be "overlapping", as opposed to being "disjoint" [37]. The novel contribution of this article is therefore not to reiterate that multiple classification is sound but to observe that an important class of apparent "level-jumping" phenomena can be understood as being a symptom of shoehorning multiple ontological classification dimensions into a single dimension.

Furthermore, the goal of regular multiple classification is typically to implicitly define a new concept by *mixing* existing concepts, rather than attempting to capture orthogonal facets of modeling elements. This is why regular multiple classification needs to support resolution strategies for cases when two original concepts, say Boat and Car, are both said to be types of an instance that represents an amphibious vehicle. Often times clashing features, such as an engine feature defined by both Boat and Car, must be renamed to support separation and unambiguous access, or must be joined in order to express that they refer to a singular feature in the instance[1].

The well-formedness rules for orthogonal ontological classification proposed in Sect. 4.3 do not suggest such resolution strategies but rather assume facets that are strictly orthogonal. In other words, the reason why classification concerns overlap in orthogonal ontological classification is not because they interact to implicitly define new kinds but rather due to the fact that certain elements can be viewed from multiple angles with their identity being the primary property that causes them to be shared among these angles.

6.2 Linguistic Dimension

Orthogonal ontological classification is not the first approach to suggest a multi-dimensional approach of any kind to metamodeling. Atkinson and Kühne distinguish between ontological and linguistic classification with their OCA approach [8]. However, this work only considers two dimensions, of which linguistic classification targets a concern which is fundamentally different to those discussed here. In contrast to the two fixed ontological vs linguistic dimensions in the OCA, orthogonal ontological classification puts no limit on the number of ontological classification concerns. Orthogonal ontological classification could thus be used to enhance frameworks like the OCA by supporting an unlimited number of ontological classification dimensions, in lieu of a simple linear domain hierarchy.

Atkinson's strictness requirements, which included ruling out associations or links from crossing metalevel boundaries [4], make perfect sense in the context of organizing linguistic levels that separate a language definition from the language usage [37]. In such a context it is critical that language defining elements do not cross into the language usage level and

---

[1] The need for such resolution mechanisms is therefore a good justification for the use of *multiple inheritance* as an indirect way of supporting multiple classification, since the resolution can be achieved at the type level, rather than for each instance individually.

language usage does not reference language definition (other than through classification). However, when the language is defined in a level that spans all ontological levels, as in the OCA [8], then it becomes unnecessary to rule out associations that cross ontological level boundaries. These remain well-defined from a language definition perspective and do not give rise to model inconsistencies. This is why cross-level associations are allowed in the orthogonal ontological classification approach, even when they connect elements that can be seen to have different order from certain classification perspectives (cf. Sect. 4).

Álvarez et al. proposed a nested metamodeling architecture arguing that strict metamodeling is not compatible with linear metamodeling [3]. They considered the linguistic dimension only, though, and the OCL [8] is an example for how the UML's infrastructure can be strictly defined using linear metamodeling, as long as the ontological and linguistic dimensions are recognized as being separate from each other. Unlike the linguistically-oriented approach by Álvarez et al., orthogonal ontological classification does not imply a hierarchical spiral but arranges all classification dimensions in a flat manner. While subordination could be imposed on the separate ontological dimensions, it depends on the modeler's perspective which dimension should be regarded as primary, secondary, etc.

6.3 Orderless Types

Orderless types [1,16] acknowledge that a different kind of type is required to support certain modeling scenarios that ostensibly require level-jumping *instance-of* relationships, i.e., deviations from a multi-level hierarchy based on ordered types. MLT* therefore explicitly distinguishes between regular *instance-of* relationships and "orderless *instance-of* " relationships, and can thus technically avoid any kind of non-strict level crossing, as respective strictness requirements only apply to regular *instance-of* relationships. However, note that orderless typing

- abandons any sanity-checking along secondary dimensions. In other words, problematic modeling scenarios can only be rejected within the realm of regular *instance-of* usage. In contrast, orthogonal ontological classification exploits the fact that all secondary classification can be treated as primary classification with a changed perspective and can be submitted to well-formedness constraints that apply per dimension.
- imposes a "standard" vs "exceptional" classification dichotomy, akin to the "tyranny of the dominant decomposition" notion known from concern-oriented approaches [41]. In comparison, orthogonal ontological classification treats all classification dimensions symmetrically.
- requires modelers to make decisions about which types should be allowed to entertain non-strict relationships. As mentioned before, such decisions will depend on context, i.e., a particular type – such as Business Asset may be a regular ordered type in one context but has to be viewed as an orderless type in another context.

6.4 Supplementary Classification

MultEcore recognizes three specific kinds of classification dimensions: application, supplementary, and data type dimensions but allows an unlimited number of classification hierarchies [32,33]. Supplementary dimensions can be used to "mix in" additional facets to modeling elements. MultEcore therefore also treats multiple classification as a first-class citizen and achieves the same respective separation of concerns as orthogonal ontological classification but is distinguished from the latter in a number of ways:

- the formal underpinning is significantly more advanced featuring a comprehensive semantics definition based on fibered semantics [39] and category theory [31].
- the inter-level relationship is deliberately underspecified and levels are referred to as "abstraction levels" rather than "classification levels". Since MultEcore prioritizes flexibility over the use of levels as a sanity-checking supporting device, it does not consider the problems and solution identified in this article as relevant to its framework in the first place.
- hierarchies are not meant to be as independent from each other as aimed at by the initial design for orthogonal ontological classification (cf. Sect. 4.3). Elements only have one potency interval which can be constrained by multiple overlapping classification concerns [31, p. 44], as opposed to the per-dimension potency approach of orthogonal ontological classification.
- concern interaction is much more permissive and resolution mechanisms that are deemed as needing further work in Sect. 4.3 to be properly evaluated in a context that intends to preserve a maximum amount of sanity-checking, are present and fleshed out in MultEcore [38].

In summary, the initial motivations for supporting overlapping multiple classification are entirely different. MultEcore's aims to support flexibility and reusability contrast with the aim of orthogonal ontological classification to maximize the potential for sanity-checking of models. As a result, MultEcore supports level-jumping within hierarchies whereas orthogonal ontological classification uses multiple hierarchies to avoid level-jumping.

6.5 Separation of Concerns

The separation of concerns achieved by multiple classification has similarities to aspect-orientation which has been used both in programming [23] and modeling [24] in order to increase modularity.

Any classification concern in orthogonal ontological classification that overlaps with other concerns with a potency greater than zero could be interpreted as a primary classification concern (e.g., the *phone* concern in comparison to the secondary *product* concern in Fig. 7). I expect real-world models to feature a number of disjoint primary classification concerns comprising rigid types [20] (e.g., *process* and *phone* concerns) with secondary concerns (e.g., *business asset* and *product*) overlapping as required.

There is no need, however, to assume one dominant classification which is supplemented with other secondary concerns. Depending on the perspective, either of the multiple classifications could be regarded as the dominant one. Therefore, the notion of "weaving" which is common to many aspect-oriented approaches that use so-called pointcut specifications to inject specifications into existing base specifications, is irrelevant for orthogonal ontological classification.

There are other techniques for combining specifications which are not weaving-based, though. For instance, the Kompose tool has been used to compose metamodels via a generic model composition operator [15]. This approach, while suitable for separating and subsequently combining concerns, does not target multi-level modeling specifically and therefore does not attempt resolve strictness challenges within multi-level modeling models.

De Lara et al. proposed *facets* to address rigid classification in model-driven engineering [30]. These facets also support multiple perspectives and can be managed with *facet interfaces* and *facet laws*. Unlike the multiple classification suggested in this article, they are specifically intended to be dynamic, i.e., can be acquired or dropped as needed. They also support overlapping of features and may require their synchronization. However, none of the three application scenarios described in [30] refers to addressing strictness challenges in multi-level modeling. Nevertheless, *facets* could potentially be used to inspire replacements of the minimal design outlined in Sect. 4.3. This is not a foregone conclusion, though, as *facets* are clearly a more complex solution and unless the flexibility that they provide is actually desired, a simpler mechanism would be sufficient and preferable to address the strictness challenges described in Sect. 2.

The multiple classification suggested by orthogonal ontological classification could be regarded as a form of "Subject-Orientation" [21] in which all subjects are isolated. Again, the work on subject-orientation did not aim at resolving strictness concerns, it only shares the recognition of multiple perspectives with orthogonal ontological classification and other concern-oriented approaches.

6.6 Classification Ensembles

In [27], I speculated about "*connected classification ensembles*" without providing a concrete definition. My *connected classification clusters* (cf. Sect. 4.3) are a concrete but more general manifestation of the idea expressed in [27]. While I derived the notion of orthogonal ontological classification from the overlapping case, my definitions and well-formedness constraints clearly continue to be functional for the scenarios discussed in [27], which can be regarded as featuring *disjoint classification concerns*.

7 Future Work

In this article, I presented a novel multi-level modeling approach but much remains to be worked out. As mentioned in Sect. 4.3, I described a rather minimal version of the approach that could potentially be much richer in its expressiveness. For instance, overlapping of concerns could be allowed to occur between multiple non-zero potency elements. This in turn would open up the question as to whether it should be possible for one concern to influence the type facets of other concerns. I deliberately made conservative choices in Sect. 4.3 in order to have a high level of confidence regarding viability, but there is a significant potential for future exploration.

A formalization of the rules presented in Sect. 4.3 would aid the development of more liberal schemes while ensuring that no inconsistencies are introduced. Empirical validations could subsequently be used to determine which balance between expressiveness and simplicity exhibits optimal efficacy.

It seems plausible that *views*, i.e., filtered versions of a comprehensive model, may help in order to ergonomically deal with multiple classification concerns. Respective *classification perspectives* could help to manage large complex models with many overlapping concerns, should they prove to be as prevalent as I expect them to be. For instance, the model shown in Fig. 9 could be rendered to look like the model shown in Fig. 4 in order to focus on a single classification concern.

Ultimately, multi-dimensional multi-level modeling will not only require adequate tools but also methodological support so that modelers can produce models that adequately reflect their domains. For instance, modelers will need methodological guidance as to how to choose between *supertypes* vs *deep metatypes* vs *orthogonal types*. However, it should be noted that orthogonal ontological classification does not increase the need for methodological guidance regarding properly choosing language constructs compared to competing approaches but rather focuses the required deliberations on domain properties as opposed to technical matters of the supporting framework.

8 Conclusion

Multi-level modeling has seen a remarkable level of interest in recent years but has also been held back by a divergence of views over its very foundations. Clarifying the nature of levels, what purpose they serve, etc., has been an express goal for many years (cf. the CfPs for MULTI 2014–2019 [34]) and has been flagged as a matter of utmost importance by the community [2, Sect. 3.2].

In my view one of the reasons for the divergence is the variety of choices that have been made regarding how elements should be dealt with that do not align within a single linear level-hierarchy. Current approaches either

- are incapable of handling such scenarios, or
- shoehorn multiple concerns into a single linear hierarchy, or
- create alternative realms in which rules applying to regular elements are suspended.

Through the lens of orthogonal ontological classification, ostensibly level-crossing associations and non-strict *instance-of* relationships are symptomatic of a failure to adequately provide *separation of concerns*. Using this perspective, level-crossing associations occur when *classification concerns* do *not globally align*, and current approaches can be understood as having responded by sacrificing a simple *instance-of* scheme through allowing level-jumping to realign the level-crossing associations. Moreover, according to my analysis, the special treatment of certain types by declaring them “*orderless*” is best understood as being required whenever *orthogonal* classification concerns *overlap*. In other words, there are no types that defy stratification into levels, there are only types which do not fit into a single linear metamodeling hierarchy because they classify elements that are at the cross section of orthogonal, domain-induced classification dimensions.

I postulate that domains not only inherently imply multiple classification levels – a phenomenon to which multi-level modeling is an answer – but additionally imply multiple orthogonal classification concerns. The emergence of approaches such as subject-orientation [21] and aspect-orientation [23] corroborate this latter view. I argue that it is unnecessary to introduce special mechanisms such as *leap potency*, *identity instantiation*, *orderless typing*, etc., to account for multiple classification dimensions. I maintain it is preferable to have a minimal set of concepts – e.g., the notion of *ontological* (*domain-induced*) classification – and uniformly apply them repeatedly, in this case for each orthogonal classification concern separately. Such minimalist approaches imply a gentler learning curve and a more intuitive application of concepts.

Interestingly, orthogonal ontological classification thus echos a principle that gave rise to multi-level modeling. While it is possible to add mechanisms like powertypes, stereotypes, constraints, etc. to two-level modeling in order to address some multi-level modeling challenges, it is more elegant to simply repeat the known relationships between objects and types to create a multi-level hierarchy. Likewise, instead of adding new forms of mechanisms to deal with some strictness challenges in multi-level modeling, it is more elegant to use the known notion of classification to create a multi-dimensional multi-level hierarchy.

In addition to being a more parsimonious solution, orthogonal ontological classification results in more stable models by using ontological anchoring, compared to approaches in which the choice of mechanisms depends on variable usage contexts, rather than the much more stable ontological nature of relationships between elements.

Note that orthogonal ontological classification cannot automatically detect all illogical modeling scenarios. For instance, if a modeler only connects Tim Berners-Lee and Profession via an *instance-of* relationship then orthogonal ontological classification cannot detect a problem, provided conformance between the elements applies, as the only problem could be the inappropriate name of the Profession concept. Orthogonal ontological classification relies on inconsistent modeling relationships to detect unsoundness. However, unlike strictness schemes that are based on linear hierarchies, it avoids false positives and thus does not require any additional mechanisms to deal with them.

Finally, orthogonal ontological classification not only retains classification order as a basis for *sanity-checking* but also supports deep characterization for all classification dimensions.

By untangling multiple classification concerns for the purpose of supporting well-formed individual classification hierarchies, the notion of orthogonal ontological classification makes a contribution to understanding what the nature of levels is and what role they should play. Eliminating at least one of the motivations for level-jumping removes some of the confusion about what the significance of level boundaries should be and could ultimately lead to a convergence of level concepts. Some multi-level modeling approaches clearly use level-segregation principles that do not support level stratification and thus cannot exploit stratification for sanity-checking (cf. Sect. 2.4). However, all approaches that use classification as their level-segregation principle could potentially embrace orthogonal ontological classification as a simple yet powerful mechanism that deals with modeling scenarios that hitherto required relaxing strictness requirements.

Therefore, the various implicit technical ways in which classification concerns have been dealt with could ideally be replaced by a single, *domain-oriented* approach for dealing with orthogonal domain concerns.

Acknowledgments

I would like to thank Colin Atkinson for numerous debates which helped sharpen some of the ideas presented in this article and significantly informed their presentation. I would furthermore like to thank João Paulo Almeida and Victorio Carvalho for their work and a number of discussions which helped me form the ideas for this article. Finally, I would like to thank SoSyM's anonymous reviewers whose comments further improved the presentation of this article.

References

1. Almeida, J.P.A., Fonseca, C.M., Carvalho, V.A.: Comprehensive formal theory for multi-level conceptual modeling. In: Proceedings of 36th International Conference on Conceptual Modeling, vol. LNCS 10650. Springer (2017)
2. Almeida, J.P.A., Frank, U., Kühne, T.: Multi-Level Modelling (Dagstuhl Seminar 17492). Dagstuhl Reports **7**(12) (2018). DOI 10.4230/DagRep.7.12.18. URL `http://drops.dagstuhl.de/opus/volltexte/2018/8675`
3. Álvarez, J., Evans, A., Sammut, P.: MML and the metamodel architecture. Workshop on Transformations in UML (WTUML'01), associated with the fourth European Joint Conference on Theory and Practice of Software (ETAPS'01), Genova, Italy (2001)
4. Atkinson, C.: Meta-modeling for distributed object environments. In: Enterprise Distributed Object Computing, pp. 90–101. IEEE (1997)
5. Atkinson, C., Gerbig, R.: Melanie: Multi-level modeling and ontology engineering environment. In: Proceedings of Modeling Wizards'12. ACM (2012)
6. Atkinson, C., Gerbig, R., Kühne, T.: Comparing multi-level modeling approaches. In: Proceedings of the 1st International Workshop on Multi-Level Modelling co-located with the 17^{th} ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2014), *CEUR Workshop Proceedings*, vol. Vol-1286, pp. 43–52 (2014)
7. Atkinson, C., Kühne, T.: Processes and products in a multi-level metamodeling architecture. International Journal of Software Engineering and Knowledge Engineering **11**(6), 761–783 (2001). DOI 10.1142/S0218194001000724
8. Atkinson, C., Kühne, T.: Model-driven development: A metamodeling foundation. IEEE Software **20**(5), 36–41 (2003)
9. Atkinson, C., Kühne, T.: Rearchitecting the UML infrastructure. ACM Transactions on Modeling and Computer Simulation **12**(4), 290–321 (2003)
10. Atkinson, C., Kühne, T.: Reducing accidental complexity in domain models. Software and Systems Modeling **7**(3), 345–359 (2008). DOI 10.1007/s10270-007-0061-0
11. Atkinson, C., Kühne, T.: Demystifying ontological classification in language engineering. In: Modelling Foundations and Applications, vol. LNCS 9764, pp. 83–100. Springer (2016)
12. Brasileiro, F., Almeida, J.P.A., Carvalho, V.A., Guizzardi, G.: Applying a multi-level modeling theory to assess taxonomic hierarchies in wikidata. In: Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion, pp. 975–980. International World Wide Web Conferences Steering Committee (2016). DOI 10.1145/2872518.2891117
13. Carvalho, V.A., Almeida, J.P.A.: Toward a well-founded theory for multi-level conceptual modeling. Software & Systems Modeling **17**(1), 205–231 (2018). DOI 10.1007/s10270-016-0538-9
14. Clark, T., Gonzalez-Perez, C., Henderson-Sellers, B.: A foundation for multi-level modelling. In: Proceedings of the Workshop on Multi-Level Modelling, *CEUR Workshop Proceedings*, vol. CEUR 1286 (2014)
15. Fleurey, F., Baudry, B., France, R., Ghosh, S.: A generic approach for automatic model composition. In: H. Giese (ed.) Models in Software Engineering, pp. 7–15. Springer (2008)
16. Fonseca, C.M., Almeida, J.P.A., Guizzardi, G., Carvalho, V.A.: Multi-level conceptual modeling: From a formal theory to a well-founded language. In: Proceedings of the 37th International Conference on Conceptual Modeling (ER 2018), LNCS 11157. Springer Verlag (2018)
17. Frank, U.: Multilevel modeling - toward a new paradigm of conceptual modeling and information systems design. Business & Information Systems Engineering **6**(6), 319–337 (2014). DOI 10.1007/s12599-014-0350-4
18. Gitzel, R., Merz, M.: How a relaxation of the strictness definition can benefit MDD approaches with meta model hierarchies. In: Proceedings of the 8^{th} World Multi-Conference on Systemics, Cybernetics and Informatics, vol. IV, pp. 62–67 (2004)
19. Gonzalez-Perez, C., Henderson-Sellers, B.: A powertype-based metamodelling framework. Software and Systems Modeling **5**(1), 72–90 (2006). DOI 10.1007/s10270-005-0099-9
20. Guarino, N.: Concepts, attributes and arbitrary relations. Data & Knowledge Engineering **8**(3), 249–261 (1992)
21. Harrison, W., Ossher, H.: Subject-oriented programming: A critique of pure objects. In: Proceedings of the Eighth Annual Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA '93, pp. 411–428. ACM, New York, NY, USA (1993). DOI 10.1145/165854.165932
22. Henderson-Sellers, B., Clark, T., Gonzalez-Perez, C.: On the search for a level-agnostic modelling language. In: Proceedings of the 25th International Conference on Advanced Information Systems Engineering, CAiSE'13, pp. 240–255. Springer-Verlag, Berlin, Heidelberg (2013). DOI 10.1007/978-3-642-38709-8_16

23. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J.M., Irwin, J.: Aspect-oriented programming. In: Proceedings of ECOOP'97, LNCS 1241, pp. 222–242 (1997)
24. Kienzle, J., Al Abed, W., Fleurey, F., Jézéquel, J.M., Klein, J.: Aspect-Oriented Design with Reusable Aspect Models, pp. 272–320. LNCS 6210. Springer (2010). DOI 10.1007/978-3-642-16086-8_8
25. Kühne, T.: Matters of (meta-) modeling. Software and System Modeling **5**(4), 369–385 (2006). DOI 10.1007/s10270-006-0017-9
26. Kühne, T.: What is multi-level modeling? Dagstuhl 17492 Presentation, Shared Documents of Dagstuhl Seminar 17492 (2017). URL `https://materials.dagstuhl.de/files/17/17492/17492.ThomasK%C3%BChne4.Slides.pdf`
27. Kühne, T.: A story of levels. In: Proceedings of MULTI 2018 co-located with the 21^{th} ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2018), *CEUR Workshop Proceedings, ISSN 1613-0073*, vol. Vol-2245, pp. 673–682 (2018). URL `http://ceur-ws.org/Vol-2245/multi_paper_5.pdf`
28. Lange, A., Atkinson, C.: Multi-level modeling with melanee. In: Proceedings of the MODELS 2018 Workshops, *CEUR Workshop Proceedings, ISSN 1613-0073*, vol. 2245, pp. 653–662 (2018)
29. de Lara, J., Guerra, E., Cobos, R., Moreno-Llorena, J.: Extending deep meta-modelling for practical model-driven engineering. The Computer Journal **57**(1), 36–58 (2012). DOI 10.1093/comjnl/bxs144
30. de Lara, J., Guerra, E., Kienzle, J., Hattab, Y.: Facet-oriented modelling: Open objects for model-driven engineering. In: Proceedings of the 11th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2018, pp. 147–159. ACM, New York, NY, USA (2018). DOI 10.1145/3276604.3276610
31. Macías, F.: Multilevel modelling and domain-specific languages. Ph.D. thesis, Faculty of Mathematics and Natural SciencesUniversity of Oslo (2019)
32. Macías, F., Rutle, A., Stolz, V., Rodríguez-Echeverría, R., Wolter, U.: An approach to flexible multilevel modelling. Enterprise Modelling and Information Systems Architectures **13**, 10:1–10:35 (2018)
33. Macías, F., Wolter, U., Rutle, A., Durán, F., Rodríguez-Echeverría, R.: Multilevel coupled model transformations for precise and reusable definition of model behaviour. Journal of Logical and Algebraic Methods in Programming **106**, 167 – 195 (2019). DOI https://doi.org/10.1016/j.jlamp.2018.12.005
34. MULTI-LEVEL MODELING WIKI. http://homepages.ecs.vuw.ac.nz/Groups/MultiLevelModeling/ (2014)
35. Neumayr, B., Grün, K., Schrefl, M.: Multi-level domain modeling with m-objects and m-relationships. In: Proceedings of APCCM'09, pp. 107–116. Australian Computer Society (2009)
36. Neumayr, B., Schuetz, C.G., Jeusfeld, M.A., Schrefl, M.: Dual deep modeling: multi-level modeling with dual potencies and its formalization in f-logic. Software & Systems Modeling pp. 1–36 (2016). DOI 10.1007/s10270-016-0519-z
37. OMG: Omg unified modeling language$^{\text{TM}}$, infrastructure version 2.4.1. `http://www.omg.org/spec/UML/2.4.1` (2011)
38. Rodríguez, A., Rutle, A., Kristensen, L.M., Durán, F.: A foundation for the composition of multilevel domain-specific languages. In: MODELS Companion, pp. 88–97 (2019)
39. Rossini, A., de Lara, J., Guerra, E., Rutle, A., Wolter, U.: A formalisation of deep metamodelling. Formal Aspects of Computing **26**(6), 1115–1152 (2014). DOI 10.1007/s00165-014-0307-x
40. Steimann, F.: On the representation of roles in object-oriented and conceptual modelling. Data and Knowledge Engineering **35**(1), 83–106 (2000)
41. Tarr, P., Ossher, H., Harrison, W., Stanley M. Sutton, J.: N degrees of separation: Multi-dimensional separation of concerns. In: Proceedings of the International Conference on Software Engineering (ICSE 21) (1999)