In Defence of Deep Modelling

Colin Atkinson[a], Thomas Kühne[b]

[a]*University of Mannheim, B6, C2.11, Mannheim, Germany*
[b]*Victoria University of Wellington, P.O. Box 600, Wellington 6140, New Zealand*

Abstract

Context: Since multi-level modelling emerged as a strategy for leveraging classification levels in conceptual models, there have been discussions about what it entails and how to best support it. Recently, some authors have claimed that the deep modelling approach to multi-level modelling entails paradoxes and significant weaknesses. By drawing upon concepts from speech act theory and foundational ontologies these authors argue that hitherto accepted principles for deep modelling should be abandoned and an alternative approach be adopted instead (Eriksson et al. 2013).
Objective: We investigate the validity of these claims and motivate the need to shift the focus of the debate from philosophical arguments to modelling pragmatics.
Method: We present each of the main objections raised against deep modelling in turn, classify them according to the kinds of arguments put forward, and analyse the cogency of the supporting justification. We furthermore analyse the counter proposal regarding its pragmatic value for modellers.
Results: Most of the criticisms against deep modelling are based on mismatches between the premisses used in published definitions of deep modelling and those used by the authors as the basis of their challenges. Hence, most of the criticisms levelled at deep modelling do not actually apply to deep modelling as defined in the literature. We also explain how the proposed alternative introduces new problems of its own, and evaluate its merits from a pragmatic modelling perspective. Finally, we show how deep modelling is indeed compatible with, and can be founded on, classic work in linguistics and logic.
Conclusions: The inappropriate interpretations of the core principles of deep modelling identified in this article indicate that previous descriptions of them have not had sufficient clarity. We therefore provide further clarification and foundational background material to reduce the chance for future misunderstandings and help establish deep modelling as a solid foundation for multi-level modelling.

Keywords: multi-level modelling, deep modelling, metamodelling, ontological classification, clabjects

1. Introduction

The search for improved modelling infrastructures to supersede the traditional four-layer infrastructure underpinning the UML to support domain modelling and model-driven development has been going on since the first UML specification was published (1). Over that time a range of multi-level modelling approaches based on different modelling architectures have been proposed, from recursively-nested architectures (2), package-based architectures (3) to minimalist architectures (4). One of the proposed enhancements that has recently gained attention (5) is the "Orthogonal Classification Architecture" (OCA), which separates domain-oriented "ontological" classification relationships from infrastructure-oriented "linguistic" classification relationships and organises them according to the tenets of strict modelling (6). The number of tools based on the OCA has steadily grown in recent years and the architecture has been used successfully in numerous industry projects and standardizations efforts (7; 8; 9; 10; 11; 12; 13; 14).

Although the OCA has become a widely adopted infrastructure for multi-level modelling, it has also been

Email addresses: atkinson@informatik.uni-mannheim.de (Colin Atkinson), Thomas.Kuehne@ecs.victoria.ac.nz (Thomas Kühne)

the subject of significant debate. While proponents of the OCA argue that it reduces accidental complexity in multi-level modelling and allows modellers to concisely describe multiple classification levels that often exist in real world domains (15), critics have argued that it is difficult to reconcile with traditional modelling conventions and makes modelling more difficult to understand, especially when combined with the deep instantiation mechanism. To date, this debate has largely focused on syntactical and pragmatic differences between the proposed approaches for multi-level modelling, since it mainly revolves around different strategies for visualising the model elements and their various properties in a multi-level model. However, in a series of recent publications (16; 17; 4), and in particular (18), critics of deep modelling (i.e., the OCA and deep instantiation) have questioned its semantic soundness at a fundamental level and claimed to have uncovered a number of inconsistencies and paradoxes entailed in its use.

If true this would obviously raise serious questions about the usability of the OCA and would significantly impact on a considerable body of work that depends on the OCA as a sound basis (7; 8; 9; 10; 11; 12; 13; 14). However, on closer analysis it turns out that the claimed paradoxes are not in fact paradoxes at all, but rather the result of either –

(a) not applying the OCA's basic premisses and definitions when evaluating it,
(b) not accepting how elements in the OCA are intended to relate to real world entities, and
(c) questioning fundamental tenets of deep modelling based merely on the observation that they are incompatible with other schools of thought.

The main goal of this article is therefore to shed light on the challenges raised in (18) and explain why the claimed paradoxes do not exist (Sect. 2 & Sect. 3). A second goal of this article is to provide a more cogent justification for the OCA in order to avoid future incorrect interpretations (Sect. 4). Finally, the third goal is to replace philosophical arguments about the difference between various schools of thought with pragmatic arguments that focus on a modelling framework's impact on modelling practice. We therefore briefly examine Eriksson et al.'s counter proposal (18) and identify potential weakness of the approach for practical modelling scenarios (Sect. 5).

2. Validity Challenges to the OCA

The orthogonal classification architecture (OCA) is a modelling framework intended to enhance domain modelling with support for multiple classification levels. Designed to improve on the traditional four-layer architecture by the OMG (19), it separates domain-oriented "ontological" classification relationships from infrastructure-oriented "linguistic" classification relationships and organises them according to the tenets of strict modelling (6). The combined use of the OCA with deep instantiation (20) is often referred to as deep (meta-)modelling (9).

The central objection to the OCA laid out by Eriksson et al. (18) is a set of claimed problems which they characterise as manifestations of a fundamental "*paradox*" inherent to the approach. This "*paradox*", which they refer to as the "*linguistic/ontological metamodelling paradox*", is expressed in terms of their understanding of the OCA shown in Fig. 1. This is an exact reproduction of Fig. 2. from Eriksson et al. (18) which is claimed to be a "*slightly modified*" version of a figure that first appeared in one of the first papers introducing the OCA (6, Fig. 3).

We argue that changing the associations between "Object", "Class", and "Metaclass" in the original diagram into what appear to be "instance-of" arrows is more than a *slight modification* and may in fact be the source of some interpretations of the OCA that are incompatible with the latter's tenets (c.f. Sect. 2.2). The original use of associations implied that, for example, instances of "Object" are ontological instances of "Class" instances. Eriksson et al's modification changes this meaning to implying that, for example, "Object" itself is an ontological instance of "Class". This is a fundamental change in semantics, not just a slight modification. We nevertheless use the overall structure of Fig. 1 in order to refer to the main principles of the OCA. In particular, Fig. 1 shows how the linguistic levels are organised horizontally, with the elements in the left-hand column (i.e., linguistic level L_0, c.f. Fig. 4) being instances of linguistic types in the right-hand column (i.e., linguistic level L_1, c.f. Fig. 4)), and the ontological levels are organised vertically (within L_0) with elements in an ontological level (e.g. O_0) being instances of elements in the ontological level immediately above them (e.g. O_1).

In their article (18), Eriksson et al. basically advance four fundamental challenges to the validity of the OCA, relating to the –

C1 location of domain metatypes,
C2 location of the infrastructure element "Metaclass",
C3 correspondence of model elements to real world entities, and
C4 omission of supertypes.

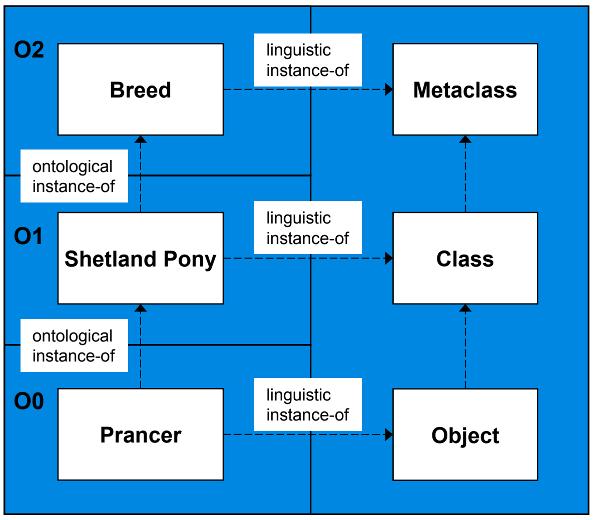

Figure 1: Reproduction of Fig. 2 from (18)

In order to investigate the validity of these challenges, we present them in the following subsections as they were raised and then evaluate their merits.

2.1. Challenge 1: Location of Domain Metatypes

Claimed Problem. Eriksson et al. claim that the OCA gives rise to a paradox regarding the location of domain metatypes. In the context of Fig. 1 they identify the following problem with the way the OCA handles the domain metatype "Breed" –

> *. . . a class in a (say UML) model (here Shetland Pony) is argued to be an instance of another class (here Breed) that is therefore "meta" to the first class (Shetland Pony) but, paradoxically, cannot be considered to be part of the underpinning metamodel.* (18, p. 2101)

By "*underpinning metamodel*" Eriksson et al. are referring to an M_2 level as it occurs in the OMG's four-layer architecture that would contain at least "Object" and "Class" of Fig. 1 (c.f. Fig. 2). They furthermore explain that –

> *. . . the Breed class is a metatype with respect to the Shetland Pony class. But clearly such a Breed metatype could not be expected to be part of the M2 layer defining the UML* [c.f. Fig. 2]*, i.e. one would not expect nor wish to find a Breed class alongside a Class class in an M2-level metamodel of a general purpose modelling language like the UML.* (18, p. 2101)

– assuming that as a type for "Shetland Pony", the type "Breed" must be located in the same level that contains the language definition (here the UML's M_2 level).

This leads Eriksson et al. to conclude that –

> *This wish to create a parallelism between the O0–O1–O2 chain of Fig. 2* [Fig. 1 in this article] *and the M0–M1–M2 chain of Fig. 1* [showing the four-layer architecture of the OMG] *in which Breed (O2) is somehow regarded as 'meta' to the Shetland Pony class creates a paradox since clearly Breed can never be a metaclass in a MOF-based modelling language (such as UML).* (18, p. 2101)

Refutation. The argument is based on the premiss that there is only one notion of "metaness" and, as a result, types (here "Breed" and "Class") of a given model element ("here "Shetland Pony") must all be assigned to the same location (here level M_2). However, the OCA is built on a fundamentally different premiss, namely the existence of two orthogonal forms of classification, and thus by extension, two forms of "metaness". Both forms of classification adhere to all the basic rules of classification underpinned by set theory, but differ in the sense that they classify model elements from the point of view of two distinct concerns. One concern is the form in which model elements are represented (the linguistic concern) and the other is the domain meaning that the model element represents (the ontological concern). Under this dual-classification assumption, the OCA therefore not only provides the user with the notions of domain instances and domain types (c.f. objects and classes in the UML), but also with domain metatypes (and further levels, if required). However, the O_2 level (see Fig. 1), which accommodates domain metatypes, is separate from the linguistic infrastructure level L_1 (see the right-hand column of Fig. 1 or Fig. 3) which has a similar function as level M_2 in the four-layer architecture.

We therefore agree with Eriksson et al. that "Breed" resides one level above "Shetland Pony", but in the context of the OCA it is invalid to conclude that "Breed" would have to be part of "*an M2-level metamodel of a general purpose language like the UML*". If the UML were modified to also support user domain metatypes by extending its M_2 level with the linguistic type "Metaclass", "Breed" would be located at the M_1 level along with "Shetland Pony" and "Prancer". In the OCA, "Breed" is located at the O_2 level for domain metatypes (c.f. Fig. 1 & Fig. 3). Hence, neither the OCA nor the aforementioned extended version of the UML would

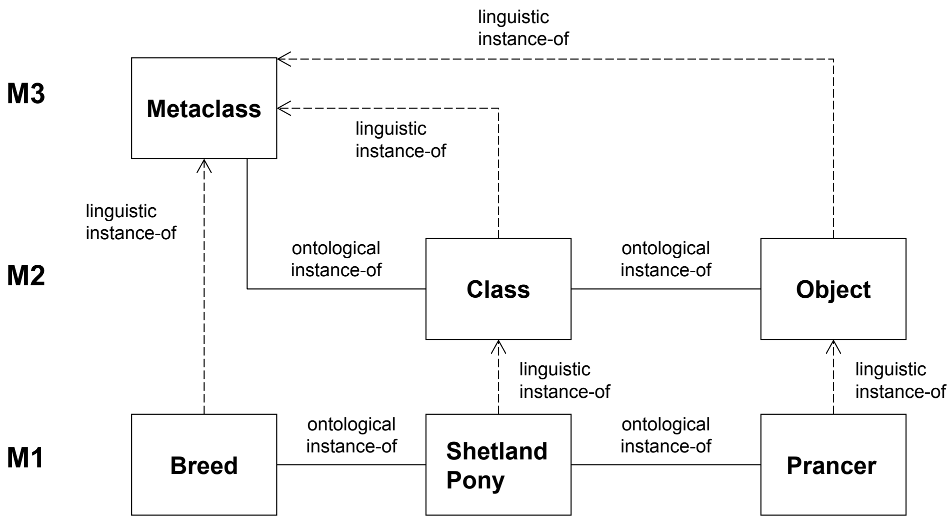

Figure 2: Reproduction of Fig. 4 from (18)

contain a paradox regarding the location of domain metatypes such as "Breed".

2.2. Challenge 2: Location of "Metaclass"

Claimed Problem. Eriksson et al. claim that the OCA makes it impossible to assign the model element "Metaclass", in Fig. 1, to a consistent location (18).

More specifically, they argue that it is impossible for "Metaclass" to simultaneously fulfil the responsibility of being the type for both "Breed" and "Class" from a single, consistent location in the architecture. To advance this argument Eriksson et al. use the diagram in Fig. 2 (a replica of Figure 4 in (18)), which they claim shows the true nature of all the ontological and linguistic instance-of relationships in the example. They then argue –

> *We note that the Metaclass said to be at linguistic level M3 has a linguistic «instanceOf» link to Breed at M1. Since «instanceOf» links are only permitted between consecutive Mx layers in strict metamodelling, a proposed instantiation relationship between M3 and M1 is invalid. One solution is to relocate the Metaclass class to level M2. However, this would make the Metaclass–Class and Metaclass–Object linguistic «instanceOf» links reside within a layer, in contradiction to the strict metamodelling rule that states that linguistic instantiation operates between layers, and only ontological instanceOf relationships are permitted within layers.* (18, p. 2103)

Refutation. The assumption that Eriksson et al. make to arrive at this conclusion is that the model element "Metaclass" depicted in Fig. 1 is a type for "Class", or in other words, that "Class" is an instance of "Metaclass". However, this is not the case in the OCA. According to the published definitions of the OCA, "Class" is neither an ontological instance-of, nor a linguistic instance-of, "Metaclass". This can be clearly seen in the original figure (6, Fig. 3) that Fig. 1 is claimed to be a "*slightly modified*" version of. The relationship between "Class" and "Metaclass" in the original figure (6, Fig. 3) is not an instance-of relationship (which is usually depicted with a dashed line that has an arrowhead) but an association (which is usually depicted as a straight line). Fig. 3 shows the intended relationship between "Class" and "(Ontological) Metaclass" in the OCA.

The relationship between "Metaclass" and "Class" in the OCA is the type for all ontological instance-of relationships between "Metaclass" instances and "Class" instances (just as an association in the UML is a type for links at the level below). However, it is not itself an ontological instance-of relationship.

Several steps would be needed to modify Fig. 2 into a framework that complies to the OCA, one of which would be to move "Metaclass" to level M_2 because it is the linguistic type for "Breed" at level M_1 and must hence reside one level above it at M_2. Eriksson et al.'s claim that this would "*…make the Metaclass–Class and Metaclass–Object linguistic «instanceOf» links reside within a layer, in contradiction to the strict metamodelling rule…*" does not apply because in the OCA there are no such linguistic instance-of relationships.

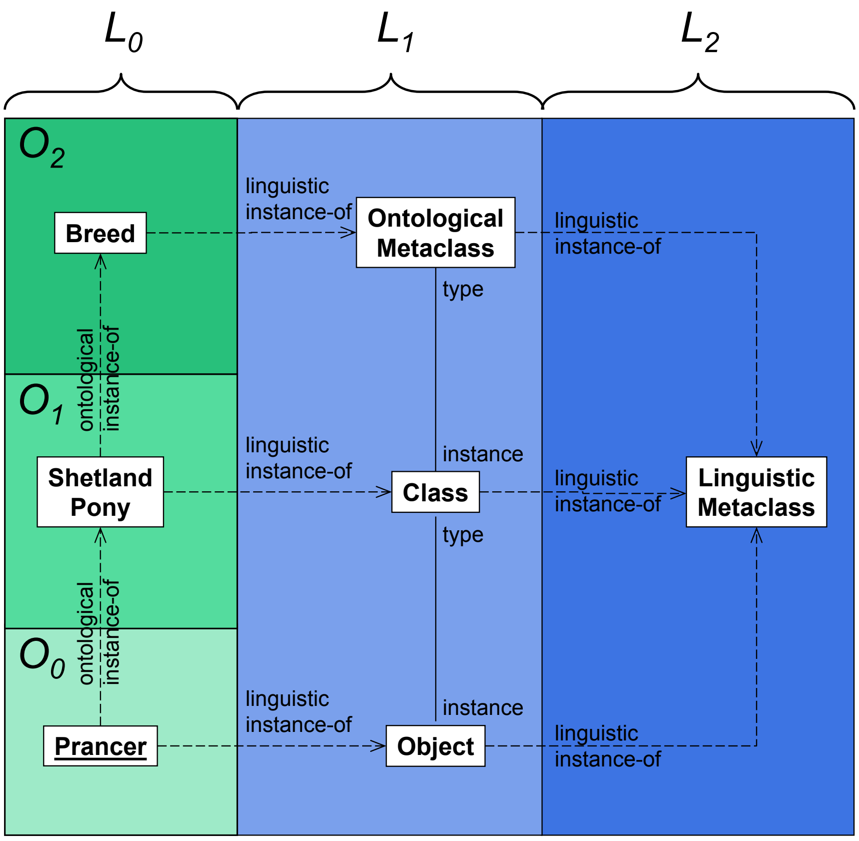

Figure 3: Distinguishing Between Two Metaclass Concepts

Hence, a second step towards improving the compliance of Fig. 2 to the OCA would be to remove all "linguistic instance-of" relationships between "Object", "Class", and "Metaclass". The final step would be to use the standard notation for "instance-of" relationships between "Prancer", "Shetland Pony", and "Breed" (a dashed line with an arrowhead).

Fig. 3 shows a modified version of Fig. 2 that not only shows the appropriate relationships as described above, but also adds a further linguistic level L_2 that accommodates an explicit linguistic type for "Object" and "Class" as it would appear in the OCA, if such a level were required. Note that Eriksson et al.'s "Metaclass" in Fig. 2 attempts to play two roles at the same time. On the one hand it is meant to be "LinguisticMetaclass", i.e., the type for "Object" and "Class" (c.f. Fig. 3) and on the other hand it is meant to be "Ontological Metaclass", i.e. the type for "Breed" (c.f. Fig. 3). Thus, when applied as intended, the principles of the OCA allow "(Ontological) Metaclass" to be assigned a consistent location without any of the problems Eriksson et al. attempted to motivate using Fig. 2. These claimed problems solely stem from introducing relationships that are not consistent with the OCA approach, most likely by confounding the different roles of "LinguisticMetaclass" and "OntologicalMetaclass" (c.f. Fig. 3). This is potentially caused by inappropriately assuming instance-of relationships between "Object", "Class", and "Metaclass" (c.f. Fig. 1).

A more concise version of the OCA which perhaps reduces the likelihood of misunderstandings is shown in Fig. 4. This version shows how the unified model element concept "Clabject", together with the notions of level (encoding level membership) and potency (encoding instantiation power), captures the fact that almost all model elements are both "classes" and "objects" at the same time, with the exception of the elements in the bottom and top levels. An instance of "Class" (e.g., "Shetland Pony") is no different to an instance of "(Ontological) Metaclass" (e.g., "Breed") in that both may have types and instances. In other words, it is possible to do away with the distinction between "Object", "Class", and "Metaclass" altogether. This has the advantages that the number of ontological levels is un-

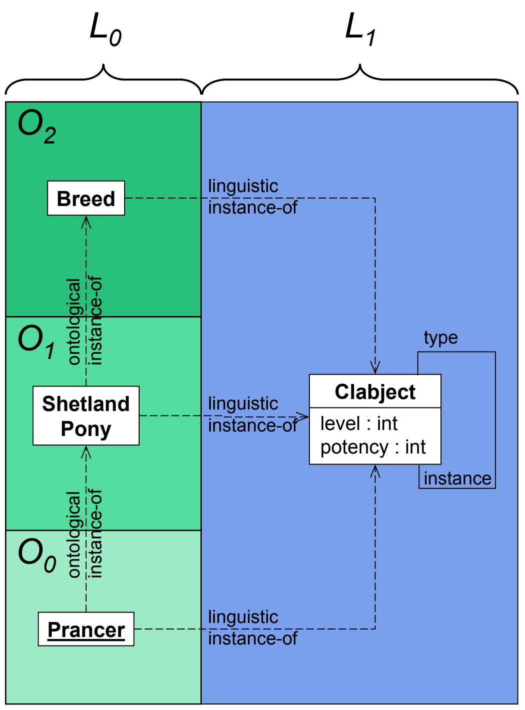

Figure 4: Concise OCA version

bounded and there can be no confusion regarding the relationships between "Object", "Class", "Metclass", etc.

2.3. Challenge 3: Correspondence to the Real World

Claimed Problem. Eriksson et al. claim that the OCA entails incorrect multiplicities between model elements and the real world entities they represent. More specifically, they claim that the OCA does not allow types to have the required 1:M correspondence relationships to the entities they correspond to –

> *. . . an object in the OCA model is only considered to be a representation of a physical thing; thus, there is a 1:1 relationship between the physical thing and the object. An object is a language construct. Before it can be referred to it must be instantiated so as to correspond to real world things, such a correspondence does not imply a 1:1 relationship, it may imply a 1:M relationship as the correspondence relationship between Shetland-Pony and the physical things in Fig. 13.* (18, p. 2112)

Refutation. Eriksson et al. appear to be making two assumptions to arrive at this conclusion –

1. ontological instance-of relationships are not proper classification relationships, but represent a kind of "*granularity abstraction*" (18, p. 2103).
2. objects in the OCA are tokens only, i.e., they only maintain 1:1 relationships with real world entities (18, p. 2112).

However, the first of these assumptions directly contradicts the explicitly stated principles underlying the OCA. As explained before, the idea that there are two orthogonal forms of classification, each fully conforming to the axioms of classification, lies at the very core of the OCA.

The second assumption fails to recognise the fact that since the OCA does not represent "Shetland Pony" as an object, but as a type at O_1, it can maintain 1:M correspondences with instances whether or not objects at level O_0 are able to.

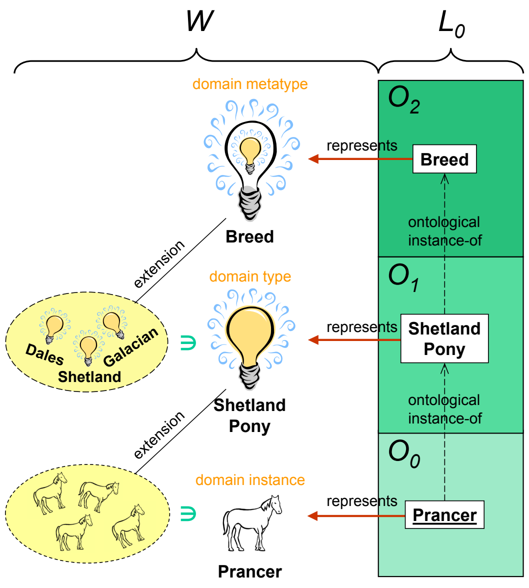

Figure 5: A Deep Model's Relationship to the Real World in the OCA

Fig. 5 depicts how entities in the real world (i.e., level "W") are represented by models elements in level L_0. The model element "Shetland Pony" represents the domain concept "Shetland Pony" through a 1:1 representation relationship and all Shetland ponies in the universe of discourse through a 1:M classification relationship by virtue of concept "Shetland Pony"'s extension. Concept "Shetland Pony" can be associated with a set (its extension), and concept "Breed" can be associated with a set of sets (the set of all pony breed sets). Thus, Eriksson et al.'s claim that the OCA has limitations preventing it from maintaining adequate correspondence relationships to entities in the real world is not compatible with the published definitions of the OCA, but is based on their own specific interpretation of how model elements in the OCA represent elements in the real world.

2.4. Challenge 4: Omission of Supertypes

Claimed Problem. Eriksson et al. claim that the OCA fails to acknowledge the necessity of including supertypes in order to properly provide an identity criterion to instances. Using a line of reasoning they derive from speech act theory (21), Eriksson et al. argue that when the purpose of a collection of subtypes is merely to partition a supertype (as in the powertype pattern (22)), the supertype must be explicitly modelled in order to be able to assign a stable notion of identity to its instances. This is necessary, they argue, because the partitioning subtypes, such as "Shetland Pony", are moment objects as opposed to substantial classes, from a foundational ontology point of view (23), and thus cannot carry a stable notion of identity, i.e., cannot be used to instantiate instances like "Prancer".

Speech act theory is concerned with the meaning of utterances with a particular focus on their intent in communication. For the purposes of this article, it is sufficient to recognise that speech act theory advocates a style of referencing individuals that always involves a "*general term*" (21) (c.f. Sect. 3.2) which needs to be a substantial universal rather than a moment object.

In foundational ontologies, the term "substantial" is used to refer to an endurant that is not a property. "Substantial universals", such as "Animal" or "Plant", have to be rigid, i.e., apply to their instances in every possible world (23, Def. 4.3, p. 101). For instance, "Prancer" the horse is always an "Animal" in all possible worlds. Another name for such a type that captures the essential features of a class of instances is "natural type" (24).

In contrast, non-rigid universals only apply in some worlds. For instance, "HappyHorse" is a classification for "Prancer" that applies to some of "Prancer"'s life span, but is not tied to "Prancer"'s identity. "HappyHorse" is therefore an example of a "phase sortal" (23, Def. 4.6, p. 125), or "state type", whereas "Happy" is an example of a "moment individual", i.e., a property that cannot exist by itself but depends on other endurants, such as "Horse". For the purposes of this article, we may equate a "substantial universal" with a concept that provides identity to its instances, i.e., characterises *what* something is, and a "moment object" as referencing a set of instances that all share a certain property, i.e., characterises *how* something is (or what it exemplifies). In general, the constituents that have a "*functional relationship*" with a moment object could come from many substantial universals, such as "Horse", "Dog", etc. provided that the respective individuals have the capacity of having the property.

Referring to the contents of Fig. 6 (a replica of their Fig. 13 in (18)), Eriksson et al. state:

> *The first problem is that the OCA model* [on the left-hand side] *does not include an explicit horse class or horse concept. Thus the meaning of "Prancer" as an instance of a Horse is undefined; . . . if Prancer is to mean an instance of a horse, a horse class is needed.* (18, p. 2112)

According to Eriksson et al., Fig. 6 shows a "more correct" model of the domain (on the right-hand side) than the OCA model (on the left-hand side) by virtue of the fact that it "*acknowledges the substantial universal and moment universal of the entities involved*".

Refutation. There are two ways in which it can be demonstrated that the claimed problem does not really exist. First, one could question the presumption that types such as "Shetland Pony" are meant to be partitions. For the sake of the argument, let us assume that the only elements in the model are "Prancer" and "Shetland Pony". Would it then be justified to argue that "Shetland Pony" is one of many other, non-existent partitions of a non-existent "Horse" class? Does "Shetland Pony" become a partition as soon it is classified by "Breed" or is it necessary for other partitions, such as "Galician Pony" to be added? If neither siblings nor a type are required to make "Shetland Pony" a partition, why is "Horse" not just a partition of a non-existing "Animal" class that has other (non-existing) partitions such as "Dog", "Cat", etc.? Why does the absence of "Horse" in Fig. 1 represent a "*problem*" but the absence of "Animal" does not (c.f. Sect. 5.2)?

Eriksson et al.'s argument relies on the assumption that "Shetland Pony" is, and must always be part of a partitioning of a superclass. However, we argue that there are models in which "Shetland Pony" is not meant to be a partition but a substantial universal and that furthermore classifying the substantial universal "Shetland Pony" as a "Breed" (see Fig. 1) does not turn the substantial universal "Shetland Pony" into a moment object. Only in the presence of a supertype "Horse", could one argue that "Shetland Pony" is meant to partition "Horse", but the claimed problem relies on "Horse" to be absent.

The second way in which it can be demonstrated that the claimed problem does not really exist is to observe

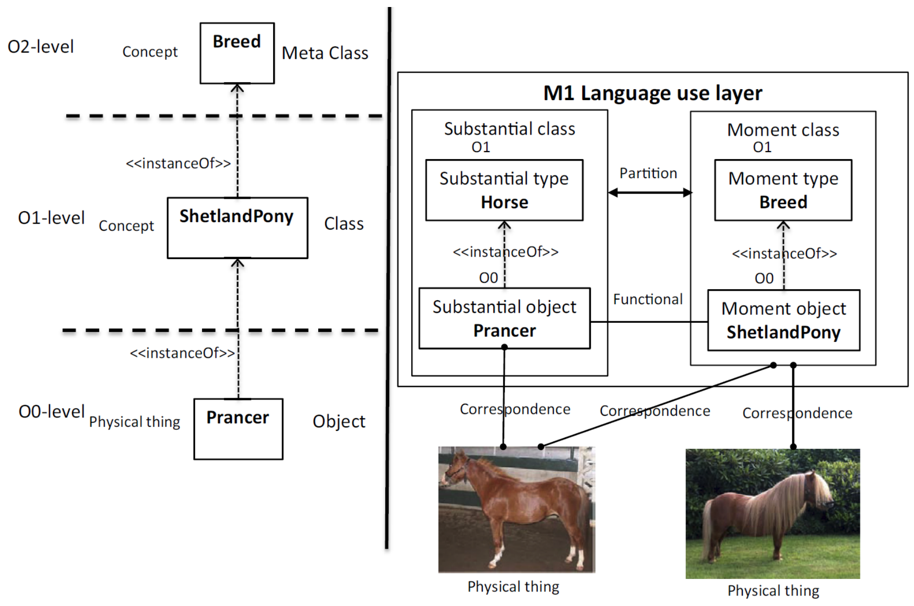

Figure 6: Reproduction of Fig. 13 from (18)

that the OCA makes no prohibition regarding the introduction of supertypes such as “Horse”. Whenever modellers judge the presence of such supertypes to be useful there is nothing in the OCA to stop them from introducing them. The use of specialisation relationships within ontological levels is a basic principle of the OCA and there are many examples in the literature where a supertype for a number of subtypes where the latter can be thought of partitioning the former is explicitly modelled (25). Moreover, not only this modelling style but the actual powertype pattern mechanism is compatible with the OCA (26).

Eriksson et al. continue their quotation above by stating –

> *The omission of Horse also means that Breed could be any Breed because no restriction is set on the Breed class by the Horse class, which consequently means that Breed means any Breed. Clearly, the model relies heavily on implicit knowledge and assumptions.* (18, p. 2112)

The first problem with this criticism is that it could well be the intention of the modeller to represent any breed with “Breed”, such as dog and cat breeds in addition to horse breeds. In this case the concept “Breed” would be broad in meaning but would have the desired precision. If, however, the modeller actually only intends to model horse breeds, it is very easy to appropriately name the concept “HorseBreed” and add desired characteristics such as a temperament (“hot bloods” vs “cold bloods), etc. It is certainly not necessary to introduce a “Horse” supertype in order to make “HorseBreed” more specific, especially when considering the potency mechanism that allows lifting all “Horse” features to the higher abstraction level of “HorseBreed” (20).

Eriksson et al. furthermore claim:

> *However, we can understand why the Horse class is left out, because this could mean that there would be four ontological levels needing explanation, perhaps with the horse concept being a linguistic instantiation of a metametaclass.* (18, p. 2112)

This is an invalid criticism since the introduction of a supertype (such as “Horse”) to an existing type (such as “Shetland Pony”) does not add another ontological level. The supertype “Horse” would just be added in level O_1 (see Fig. 1) and no “*metametaclass*” is required.

Finally, Eriksson et al. state:

> *However, leaving out the Horse class in the model is one reason for the linguistic/ontological paradox. It is symptomatic of a confounding between Horse and Breed, seen more obviously in some software engineering papers where, for example, Task is confounded with TaskKind.* (18, p. 2112)

Being able to introduce “Horse” as a supertype of “Shetland Pony”, if so desired but without being required to do so, is an advantage of the deep modelling approach, not a symptom of incorrect modelling. There is no confounding of “Horse” and “Breed”. Here, “Horse” corresponds to “Task” while “Breed” (or “HorseBreed”/”HorseKind”) corresponds to “TaskKind”. The fact that deep modelling permits the absence of “Task” does not imply that there is a confounding of “Task” with “TaskKind”.

2.5. Summary

In the previous subsections we presented core criticisms levelled at the OCA by Eriksson et al. along with the assumptions/premisses that they used to justify their claims. In the respective refutation sections we explained how the assumptions/premisses made are incompatible with the true principles and tenets of the OCA.

As an approach can only be shown to have inconsistencies and give rise to paradoxes by adopting the premisses of the approach, we maintain that the criticisms raised by Eriksson et al. do not apply to the OCA. In order to point out actual paradoxes inherent to the OCA, Eriksson et al. would have to use scenarios and arguments that actually comply to the published OCA principles, rather then derive them from their own interpretations. In other words, this section was not concerned with whether the OCA is right or wrong, its aim was to establish that the criticism raised does not apply to the OCA.

A different line of argumentation is of course to directly challenge the premisses of the OCA, ideally demonstrating that their adoption would be detrimental to modelling practice. We have reserved the discussion of such arguments to the next section and will return to the notion of emphasizing modelling practice in Sect. 5.

3. Challenging Deep Modelling

In this section we discuss arguments by Eriksson et al.’s that cannot simply be dismissed on the basis of not being applicable to the OCA. In particular, to be generous to Eriksson et al., we may assume that when observing the absence of a supertype “Horse” in the model shown by Fig. 1, they did not presume it could not be added – in which case the claim could be straightforwardly refuted (c.f. Sect. 2.4) – but actually intended to make the stronger claim that the OCA permits the creation of ill-formed models by allowing the omission of supertypes such as “Horse”. In other words, we may assume that they argue that all models intending to have meaningful “Prancer” instances, but omitting a supertype such as “Horse” are ill-formed because subtypes such as “Shetland Pony” are not capable of giving rise to instances of their own, and that the OCA should prevent the construction of such models.

In contrast to the previous challenges, where Eriksson et al. derive a set of apparent problems from a set of premisses that are incompatible to those made in the OCA, this criticism directly challenges one of the core tenets of OCA, namely that a model element can serve a dual role as an instance and a type. It is this duality which allows models to be created in which a single model element (e.g. “ShetlandPony”) participates in some ontological instance-of relationships playing the role of a type (e.g., for “Prancer”) and others playing the role of an instance (e.g., of “Breed”). The OCA intrinsically supports such multi-instance-of chains, where all the model elements in inner levels play both roles.

One could summarise this major criticism of Eriksson et al. as the postulate that deep instantiation chains are invalid. To simplify the discussion, we separate our refutation of this fundamental challenge into two parts, i.e., challenges C5 and C6.

C5 infeasible deep instantiation chains.
C6 inability of subtypes to have instances of their own.

We conclude this section by discussing a further concern that Eriksson et al. raise regarding deep modelling. This is a pragmatic concern claiming that the OCA provides no systematic guidance to modellers regarding the assignment of model elements to levels. We characterise this challenge as:

C7 ad hoc assignment of levels.

3.1. Challenge 5: Infeasible Instantiation Chains

Claimed Problem. Eriksson et al. claim that the OCA promotes the creation of “*invalid*” instantiation chains – namely, instantiation chains with an instantiation depth of more than two, such as that between “Breed”, “ShetlandPony” and “Prancer” in Fig. 1. They state

> *. . . the paradox is that it is assumed that (1) Breed is a metaclass of Shetland Pony, and that (2) Prancer is an instance of Shetland Pony, and Shetland Pony is an instance of Breed, which involves a chain of two or more type models.* (18, p. 2103)

Referencing their Fig. 3 (see Fig. 7), they state –

> *The figure shows on the left-hand side, Horse as a class that is also an instance of Class, which is impossible within the conventional object-oriented paradigm. . .* (18, p. 2102)

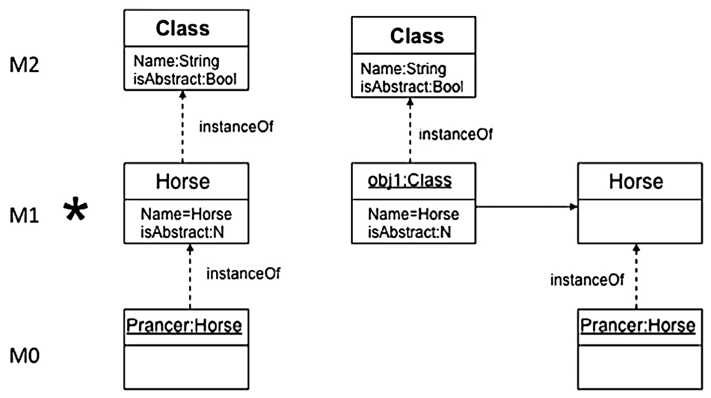

Figure 7: Reproduction of Fig. 3 from (18)

More fundamentally, they claim –

> *. . . an instance (e.g. Horse as an instance of Class in Fig. 3* [Fig. 7]*) cannot be further instantiated since it is already an individual (an object – an object is an instance of a class in object-oriented modelling). In other words, a concept cannot be an instance of another concept. . .* (18, p. 2103)

– suggesting that this would be "*contradictory to the set-based rules of modelling*" (18, p. 2103).

Refutation. We first observe that chains of the kind occurring on the left-hand side of Fig. 7, which Eriksson et al. claim to be "*impossible*", are the basis of the four-layer architecture of the UML. For instance, in the chain "Horse" → "(UML-)Class → "(MOF-)Class" the UML concept "Class" at level M_2 is considered to be an instance of the MOF concept "Class" at level M_3 but at the same time a type for "Horse" at level M_1. Eriksson et al. suggest that such chains are a contracted form, involving an "*isotypical mapping*" (here between "obj1:Class" and "Horse", see Fig. 7), of a more complicated situation (the right-hand side of Fig. 7). We note that a so-called "*isotypical mapping*" is always required when trying to support multi-level technology on the basis of two-level technology, an approach that has been dubbed "*two-level cascading*" in (27). Whether or not one uses such an approach or native multi-level technology that allows the instance facet "obj1:Class" and the type facet "Horse" to be represented within a single clabject ("Horse"), the widespread use of such instantiation chains and the absence of any literature reporting corresponding problems, suggests that there are no fundamental issues associated with them. Indeed, Eriksson et al. do not point out any concrete negative consequences.

In a similar vein, Eriksson et al. state with respect to powertypes –

> *Although useful in certain aspects of metamodelling (e.g. [41]), the powertype, as used in software engineering today, ignores both the ontological ideas of a foundational ontology, such as the Unified Foundational Ontology [42], as well as the pragmatic framework of language use theory.* (18, p. 2104)

Here, they even explicitly ascribe practical value to powertypes, but nevertheless reject them as a sound basis for modelling because they do not comply to certain theories.

We maintain that compliance to a certain theoretical underpinning should not be a criterion to reject or adopt a modelling approach. The exception would be a theoretical underpinning that deserved gold standard status because it had been proven that any deviations from it result in suboptimal solutions. This is clearly not the case for the quoted theoretical underpinnings. In our view, it would be futile to argue whether "language use theory" or "set theory" is a better theoretical underpinning for domain modelling. Ultimately, only the modelling pragmatics of the solution in question matter, not which theoretical framework it conforms to.

It is importance to observe, however, that both powertypes and deep instantiation chains as they occur in deep modelling can be given a set-theoretic interpretation (28). In fact, Eriksson et al. themselves observe –

> *. . . in Fig. 2* [Fig. 1]*, we see that Shetland pony (a class) is an ontological instance of Breed (also a class). In set theoretic terms, this can only be valid if Breed is a power set (or more strictly a family of sets – see above) (Fig. 6) such that each element of the Breed set maps*

to a part of a Horse set (itself a partition). (18, p. 2104)

Indeed, "Breed" can be given the meaning of a set of sets. This, however, contradicts Eriksson et al.'s earlier statement that "*a concept cannot be an instance of another concept*" (see above) because with the aforementioned set-theoretic interpretation, the concept "Shetland Pony" (which has the meaning of a set of instances) is an element of the set "Breed" and the latter can hence be given the meaning of a set of sets. Instantiating "Breed" means choosing an element (e.g., "Shetland Pony") from its extension set and since the chosen element is a set itself, it can be instantiated again. We do not dispute that Eriksson et al.'s chosen approach prohibits a concept to be an instance of another (meta-) concept, but we conclude that their postulate regarding the infeasibility of deep instantiation chains does not apply to deep modelling.

Even though deep instantiation chains are feasible, there could still be disadvantages associated with their use. It may be the case that Eriksson et al. have no objections to the widely used multi-level hierarchies in the linguistic classification dimension but take issue with multi-level hierarchies in the ontological dimension. Their argument that certain concepts, such as "Shetland Pony", that are instances of other (meta-) concepts are unable to provide a notion of identity to their instances, suggests that they believe deep instantiation chains are inadequate for domain modelling purposes. We discuss this argument in the following subsection.

3.2. Challenge 6: Inability of Subtypes to have Instances

Claimed Problem. Eriksson et al. claim that "Prancer" is "*undefined*" unless a "Horse" class is added to a scenario in which "Prancer" is considered a "Shetland Pony" (see the first quote in Sect. 2.4), and reference speech act theory to claim that the kinds of types (e.g., "ShetlandPony") representing the partitioning instances of a powertype (e.g,. "Breed") are not "proper" types that can be used for further instantiation. Eriksson et al. argue that they are moment objects which are tied to the existence of another type (i.e. the supertype they partition, e.g., "Horse") and thus do not have the capacity required to convey a notion of identity to their instances. Eriksson et al. therefore relegate such types to a kind of property or label attached to instances of the supertype (c.f. the "*functional relationship*" between "Fido" and "Collie" in Fig. 11). In the example, therefore, Eriksson et al. view "ShetlandPony" as "*a moment individual of the Breed class*" (18, p. 2112) so that "Prancer" essentially has a "Breed" feature whose value is "Shetland-Pony" (18, Tab. 4).

We note that, if true, the criticism by Eriksson et al. would have far and widespread consequences. They essentially challenge —

- the powertype pattern (22) (c.f. (18, p. 2104) in which the core principles of the powertype pattern are said to be "*untenable*" and "*misleading*").
- object-oriented metamodelling, for instance as the basis for the CDIF standard, or the OMG's four-layer architecture (1).
- all modelling approaches that permit the use of a type like "Shetland Pony" without enforcing the presence of a supertype "Horse".

Refutation. We believe there are multiple arguments against modelling membership to a subkind (e.g., a breed such as "Shetland Pony") through instance properties, and that it is not only justifiable but also advisable to use classification instead. In the following subsections we present five of them.

Multiple Types Argument

One objection Eriksson et al. raise, quoting Searle (21), is that an object can only be the instance of one class:

> *. . . an object can only be instantiated in one class and subsequently be referred to belonging to that one class, because in order to* 'secure continuity of reference, we need a criterion of identity, and the general term associated with the name provides that criterion' *[23, p.167] . . .* (18, p. 2107)

This states that "Prancer" needs to be an instance of the "*general term*" "Horse" and can therefore not be an instance of "Shetland Pony" at the same time.

Yet, there is a long tradition of modelling with specialisation hierarchies that contradicts this position. For example "Prancer" is typically regarded as an instance of "Shetland Pony", an indirect instance of "Horse", an indirect instance of "Animal", etc. up to "Thing". In fact, the counter argument to the position of Eriksson et al. would appear to date back to Aristotle:

> *Thus, 'man' is predicated of the individual man; but 'animal' is predicated of 'man'; it will, therefore, be predicable of the individual man also: for the individual man is both 'man' and 'animal'.* (29, Part 3)

But of secondary substances, the species is predicated of the individual, the genus both of the species and of the individual. (29, Part 5)

Also note that Eriksson et al.'s claim does not follow from their quotation of Searle (see above), since Searle does not make it mandatory for there to be a *single* general term only. Indeed, after arguing that a "*general term*" is needed to define an identity criterion in order to reference an object, such as the former French president "de Gaulle", Searle writes:

. . . some term or range of terms is analytically tied to the name 'de Gaulle'. (21, p. 167)

Wiggins, working in the field of ontologies, does not see an issue with the idea of an object being characterised by two types either. He accepts the idea of generalisation hierarchies consisting of substantial universals, as long as their notion of identity is compatible with each other (23, p. 99).

However, even if Eriksson et al. were to agree to the idea of an object having multiple substantial universals as (indirect) types, they would still contest that "Shetland Pony" is a substantial universal, claiming that it is "*a moment individual of the Breed class*" (18, p. 2112).

Natural Type Argument

We agree that it would, in general, be incorrect to regard "Prancer" as an instance of a role (e.g. "DressageHorse") or a state (aka, "phase" (23), e.g. "HappyHorse"). Such types are characterised by their volatile nature, i.e., a horse may be used as a dressage horse by a particular rider for a while, but such a relationship could change to that of being a race horse for another rider or to the absence of any relationship without affecting the identity of a horse. Similarly, being happy or sad is a property of an instance that may change at any moment of time, but without affecting the identity of a horse. These types thus share the fact that "Prancer" can be an instance of them, but is not an instance of them in all possible worlds. Even though "Prancer" has the potential to be happy and a race horse, there are worlds in which "Prancer" is unhappy or not a race horse. In reference to the examples of "DressageHorse" and "HappyHorse", Eriksson et al. are correct to state –

Attributes are used to describe individual objects, not to sub-classify them. (18, p. 2121)

We also agree with them that, in general, "Racehorse" – one of their other examples, along with "Carthorse" (18, p. 2112) – is unsuitable as a substantial universal, i.e., a type that can provide identity to its instances. A horse breed is characterised by the fact that its members consistently transmit distinctive characteristics, such as conformation, colour, or disposition to their offspring. This is certainly not true for race horses. A race horse may be more likely to have offspring that are good race horses as well, but such transmission is not guaranteed. The example "Carthorse" is also problematic because this is not an official horse breed either and it is debatable whether it refers to a role – if it did, instances of real horse breeds, such as Arabian, could play the role of a cart horse – or a superset of actual horse breeds, including "Dutch Heavy Draft", "Estonian Draft", etc.

Be that as it may, "Shetland Pony" (or "Collie"; the type originally used in the figure from which Fig. 1 was derived), can justifiably be regarded as a substantial universal. The features that make "Prancer" a member of the breed "Shetland Pony" are constant for "Prancer" from cradle to grave. In all possible worlds "Prancer" will satisfy the criteria to be considered a "Shetland Pony". Being a "Shetland Pony" is essential to "Prancer", as opposed to having four legs, for example. Even with three legs "Prancer" would still be a "Shetland Pony". Just as "Prancer" cannot stop being a "Horse" without losing his identity, he cannot stop being a "Shetland Pony" either. If any of the criteria associated with being a "Shetland Pony" would stop applying to "Prancer" at some point in time, this would raise serious questions about "Prancer"'s identity. In other words, the "*property*" – as Eriksson et al. put it – of belonging to the subset of horses that form the subkind "Shetland Pony" is not just any property. Its constant nature gives rise to a substantial universal.

Classification in Logic Argument

According to Lorenzen, "Shetland Pony" is therefore a predicator, rather than an appredicator (30, p. 52). Classic logic regards both "Horse" and "Clever" (the latter as a nominalised property) as predicators and thus cannot distinguish between Prancer being a "clever horse" as opposed to a "horseish clever". However, clearly horses should be regarded as having the ability of being clever as opposed to the other way round. Therefore "Clever" should have secondary status compared to "Horse". Typically, a conjunction of such secondary characterizations is used to determine whether an instance falls under a concept. As a result, "Shetland Pony" should be accepted as a perfectly valid classifier for "Prancer".

Eriksson et al. have other ideas about the function of

"Shetland Pony" in their specific approach, but in fact we argue that Eriksson et al. overlook the significance of certain properties that are referenced in a concept's intension because they are essential to the instances that fall under the concept.

Species Argument

Provided that "Shetland Pony" is an acceptable type for "Prancer", it can be argued that it is indeed a more suitable type than "Horse". Going back to Aristotle (29), a classic way to distinguish between more general and more specific types is to refer to the former as the "genus" of an instance and the latter as the "species" of an instance. For example, Plato can be considered to have the genus "Animal" and the species "Man". Plato is thus a direct instance of "Man" and an indirect instance of "Animal".

> *Of secondary substances, the species is more truly substance than the genus, being more nearly related to primary substance. For if any one should render an account of what a primary substance is, he would render a more instructive account, and one more proper to the subject, by stating the species than by stating the genus. Thus, he would give a more instructive account of an individual man by stating that he was man than by stating that he was animal, for the former description is peculiar to the individual in a greater degree, while the latter is too general.* (29, Part 5)

In our example, "Shetland Pony" also gives a more complete characterisation of "Prancer" than "Horse". As a result, "Prancer" should be modelled as an instance of "Shetland Pony" (its species) with the latter having a supertype "Horse" (its genus).

A species is defined by its genus and its differentia, with the differentia specifying what distinguishes various species within a genus. The differentia therefore correspond to a UML "generalisation set" (31) and thus define partitions of the genus. Eriksson et al. acknowledge the partitioning function of "Shetland Pony" and other breeds, but do not regard "Prancer" as an instance of "Shetland Pony". This is in stark contrast to many classical treatments of the matter that regard "Prancer" as an instance of its species rather than an instance of its genus with the added information that the value of its "species" property is "Shetland Pony". In other words, in many classic works the species is the direct type of an instance, not the property (moment) of an instance.

Abstraction Argument

Eriksson et al. regard "Prancer" as an instance of "Horse", with breeds such as "Shetland Pony" and "Apaloosa" having no relevance to "Prancer" and other horses other than providing a partitioning rule for the set of all horses. It could be argued, however, that universals such as "Horse", "Shetland Pony" and "Apaloosa" are merely fictitious and that "Prancer" is the only real entity. This is the so-called "nominalist" school of thought in philosophy (32). While "Prancer" is the only real entity, we can nevertheless apply abstraction and classify both "Prancer" and "Socks" with the concept "Shetland Pony". Other individuals such as "Shergar", "Red Rum", and "Seabiscuit" can be classified with the concept "Apaloosa", etc. In a further abstraction step, we can then generalise "Shetland Pony" and "Apaloosa" to "Horse". The concept "Horse" is hence twice removed from instances such as "Prancer" in the sense that it is an abstraction (generalisation) of an abstraction (classifier "Shetland Pony"). Again, this suggests that "Shetland Pony"' is a perfectly valid type for "Prancer" without needing to depend on a "*general term*".

3.3. Challenge 7: Ad Hoc Assignment of Levels

Another of the "*key problems*" with the OCA that Eriksson et al. refer to is the lack of a systematic way of assigning model elements to levels. They state –

> *No explicit principle exists for judging at which level a particular element should reside.*(18, p. 2123)

Refutation. We agree that the existing literature on deep modelling does not contain an account of how to assign a particular modelling element to a level in an absolute manner. We, however, argue that an OCA modeller does not need to care about explicitly assigning model elements to levels. The position of an element in the O-level hierarchy is not assigned manually but is implied by its network of ontological instance-of relationships with other elements. In other words, the meta-level boundaries between the O-levels are implicitly defined. The levels that emerge in this manner are not arbitrary, however, because ontological instantiation is constrained to be anti-cyclic and level-respecting (33). This rules out paradoxical situations such as types being both instances and subtypes of other types at the same time. Should the modeller create a network of elements that rules out any sound hierarchy, a good tool will alert the modeller to this fact and indicate any cycles and/or violations of the "level-respecting" property.

In (34) we provided three explicit litmus tests that can be used to determine whether a modelling element should reside at the same level or one level higher than a related model element. These litmus tests can thus be used to work out the appropriate relative instantiation relationships and the latter then simply imply the absolute location of elements in the hierarchy.

Note that relationships other than "ontological instance-of", such as specialisations, aggregations, compositions, etc. are of course admitted between model elements. The only restriction is that they must only relate elements within one and the same *O*-level in order to maintain strictness (35). The "*key problem*" identified by Eriksson et al. that –

> *The architectures implies* [sic] *that all relationships between elements are type–instance relationships [18].*(18, p. 2123)

– does therefore not apply to the OCA.

3.4. Summary

As mentioned previously, the core challenges discussed in this section are different to the challenges discussed in Sect. 2 in that they question the fundamental tenets and properties of deep modelling, as opposed to observing problems that are created by applying incompatible premisses. In defence of deep modelling we –

(a) ascertained that there are no logical problems with supporting deep instantiation chains,

(b) noted that there no known concerns regarding deep instantiation chains when applied in the linguistic dimension (such as in the four-layer architecture),

(c) argued that it is inadequate to regard an instance (e.g., "Prancer") as an instance of a general concept (e.g., "Horse") and treat its membership to a subtype (e.g., "Shetland Pony") as a property, when the property is constant in nature and thus essential to the instance. We argued that in such cases, the instance (e.g., "Prancer") should be regarded as an instance of the subtype (e.g., "Shetland Pony"), and an indirect instance of the supertype (e.g., "Horse"). We showed the latter approach to be consistent with classic works and foundational ontologies, and

(d) pointed out that litmus tests for determining correct instantiation relationships between ontological model elements exist and that these are sufficient to derive absolute level membership from the formal requirements attached to ontological instantiation.

We conclude that upon closer scrutiny Eriksson et al.'s final challenges towards deep modelling are not sound. Eriksson et al. did not reveal any internal inconsistencies inherent to deep modelling nor demonstrate that adopting deep modelling principles entails disadvantages for modellers.

4. Foundations for Deep Modelling

One of the reasons why these unwarranted criticisms may have been levelled at deep modelling is that existing descriptions in the literature have lacked the clarity required to preempt them. We therefore now provide a more explicit exposition of the foundations for deep modelling. Here, we use the original example concepts used to discuss the OCA (6, Fig. 3); we only used Eriksson et al.'s example concepts earlier in order to facilitate reference to their criticism. The change from "Shetland-Pony" to "Collie" and "Prancer" to "Lassie" does not introduce any qualitative difference, facilitates references to earlier work on multi-level modelling.

4.1. Realism Approach to Universals

Without endorsing a particular school of thought on the age-old debate about universals, our approach of viewing model elements in various *O*-levels as representing instances, concepts, meta-concepts, etc. is best aligned with the so-called "realist" position in philosophy which assumes that universals exist independently of individuals and thus enjoy some form of "existence" in the real world (32) (c.f. Fig. 5).

This, along with the next subsection, clarifies the relationship of model elements in the OCA with the real word, making explicit that we intend each modelling element to have a "token representation" relationship to a real world entity (the latter potentially being an abstract concept only), and all model elements with a type function to also maintain type-instance correspondences to the instances they classify (potentially being multiple levels removed when they are higher than O_1 and their correspondence to material real world entities is of concern).

4.2. Sound Semiotic Interpretation

Deep instantiation chains of the kind occurring in deep modelling can be explained by using a tool from semiotics: the so-called semantic triangle (aka, "meaning triangle", "semiotic triangle", etc.). The particular version popularised by Ogden and Richards (36) in 1923 has been widely used in linguistics ever since.

Fig. 8 shows a semantic triangle that explains that a sign or symbol (here the UML object "Lassie") symbolises or evokes a mental notion (here the thought of Lassie) that refers to a real object (here the real Lassie).

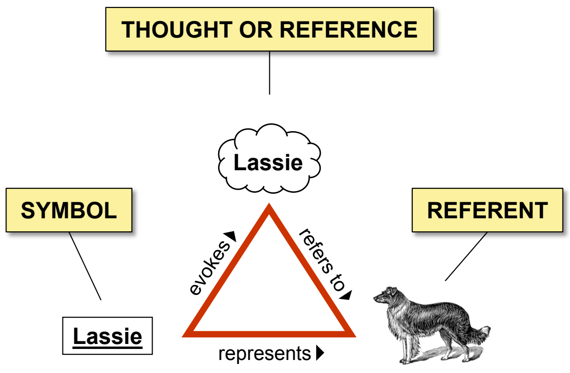

Figure 8: Semantic Triangle

The bottom base of the triangle (labelled "represents" in Fig. 8) is typically thought to be an imputed (i.e., derived) relationship. In the real world, signs or symbols in general cannot directly refer to their referents but must rely on the mediating function of the mental notion at the top of the triangle. It is worth remarking, though, that in the context of prescriptive software engineering models, symbols can easily function as proper names, i.e., unique identifiers for their referents, that do not necessarily need a mediating instance. This fact can significantly simplify the challenge of referencing an element since no detour via a "general term", as suggested in speech act theory, is then required.

It is possible to link semantic triangles so that the referent of a semantic triangle is viewed as a symbol to be interpreted by a neighbouring triangle. For example, Sowa uses the chain "Yojo : Cat" $\xrightarrow{name}$ "Yojo : Word" $\xrightarrow{quote}$ "Yojo : String" in which each arrow corresponds to a semantic triangle that connects a symbol (left) to a referent (right) using a particular interpretation (37, Fig. 3).

In Fig. 9 we link two types of semantic triangles whose function is "representation" and "classification" respectively. In the bottom-most representation-triangle, the symbol (model element "Lassie") is on the right-hand side while its referent (real Lassie) is on the left-hand side. The classification-triangle to the top left of it shares both the reference (mental notion "Lassie") and the referent (real Lassie) with the bottom-most representation triangle, but views the latter's referent as the object to be classified. Note that unless we are assuming that the real world elements are artefacts, we are using the real Lassie to take on the role of a sign in this particular triangle. This is not a problem, however, as real physical objects can take on many roles such as signs (e.g., a street sign) or models (e.g., a cat statue in

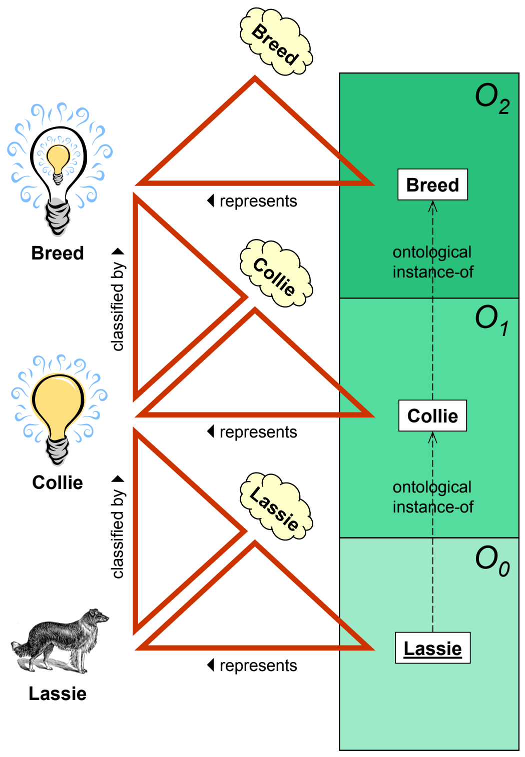

Figure 9: Linking Semantic Triangles

ancient Egypt). One could argue that the real "Lassie" has prototype status in this context as it is chosen to represent (but not "*is*") the type "Collie".

The referent "Collie" of the bottom-most classification triangle then becomes the object for the classification triangle above it and is represented by the model element "Collie". Fig. 9 hence depicts how a multi-level domain and its associated model elements can be given a sound interpretation by linking multiple semantic triangles.

Fig. 10 gives meaning to the notion of "ontological classification" – here between the model elements "Lassie" and "Collie" – by linking three semantic triangles with each other. The left-hand classification-triangle explains classification in the domain and the right-hand classification-triangle explains how this domain classification is mirrored by the "ontological instance-of" relationship between the model elements.

4.3. Semantics Based on Set Theory

Clabjects (such as "Collie") have the capacity to be the type for other instances (such as "Lassie") despite the fact that they are instances themselves (e.g., of "Breed"). Although Eriksson et al. categorically reject this capacity, it can be given a set-theoretic semantics

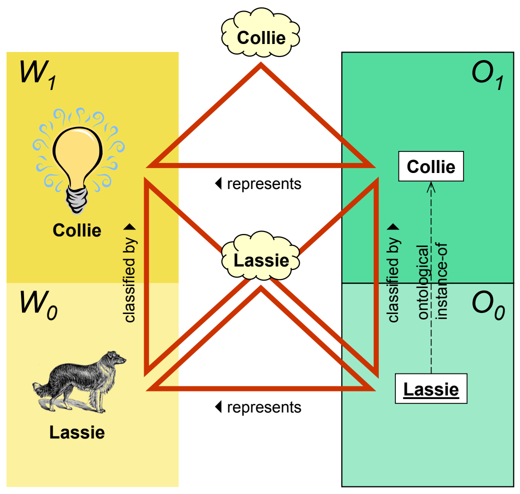

Figure 10: Ontological Classification

by understanding the extensions of clabjects such as "Breed" to be sets of sets. Fig. 5 shows how the extension of "Breed" contains elements, which are shown as named light bulbs, that can be regarded as sets in a pure set-theoretic interpretation. The light bulb named "Collie" is then regarded as being the set that "Lassie" is an element of. Kühne and Steimann give such a set-theoretic interpretation not only of deep instantiation but also of "potency" (28).

The nesting depth of such sets of sets depends on the potency of the concept. If the concept can give rise to instantiation depth two, then the respective set of sets is sometimes regarded as the extension of a so-called "powertype" (22; 3). Powertypes add the additional constraint that there is an explicit named superset of which all sets that are members of the powertype extension are subsets.

4.4. Validation through Biological Taxonomies

Metatypes, such as "Breed", do not just provide names for partitioning sets such as "Dog", as Eriksson et al. claim (18, p. 2104), their instances, such as "Collie", when partitioning a superset (here "Dog"), imply subsets within that superset. The crucial point is that they are actually *relevant* for the instances in the subsets they identify. For example, all "Collie" instances are constrained to have one of the allowed breed colours specified in type "Collie". Likewise, all instances of "Mammal" (which is a partition of the clade "Mammaliaformes" and an instance of "Class") have a neocortex. Hence, despite being instances themselves, partitions like "Collie" and "Mammal" unmistakeably characterise the elements in the subsets they identify. In other words, they are types for the elements in the subsets they identify.

The knowledge representation language Telos (38) and the tool ConceptBase (39) are examples of approaches that essentially use clabjects to capture multi-level domains, such as biological classification.

5. Pragmatic Problems

In the previous sections we focused on responding to the challenges to deep modelling raised by Eriksson et al. and on clarifying the philosophical foundations for the approach. In this section we turn our attention to the counter proposal put forward by Eriksson et al. and discuss to what extent, if any, it can be regarded as having any practical advantages for modelling typical domain scenarios.

The alternative proposal put forward by Eriksson et al. is depicted in Fig. 11 which is an exact reproduction of Fig.12 from (18, p. 2111). Eriksson et al. characterise this as a –

> *Complete multilevel framework based on language use – to replace the strict metamodelling architecture of Fig. 1 when modelling in information systems development and software engineering.* (18, p. 2111)

– and justify its form based on speech act theory and foundational ontologies. The key idea is to enforce the notion that partitioning subclasses of a class (of the kind that give rise to the powertype pattern) must not be thought of as types, but must be modelled as objects that are furthermore regarded as properties of instances of the superclass.

Region A in Fig. 11 contains the predefined language concepts used to represent domain content. As such it corresponds to the language metamodel in the OMG's four-layer architecture (i.e. M_2) and to the linguistic type model in the OCA (i.e. L_1). It is Eriksson et al.'s single "*underpinning metamodel*" (18, p. 2101). The key difference between Eriksson et al.'s framework and the OCA is manifest in regions B and C of Fig. 11 which Eriksson et al. refer to as the "language use" (M_1) level. This is the level at which users instantiate constructs of the language defined in M_2 to represent some domain in the real world. In contrast to the OCA, which splits this "language use" level into multiple ontological levels, Eriksson et al.'s framework splits this level into two separate regions – one (region B) containing all the supposed types and the other (region C) containing all the supposed instances.

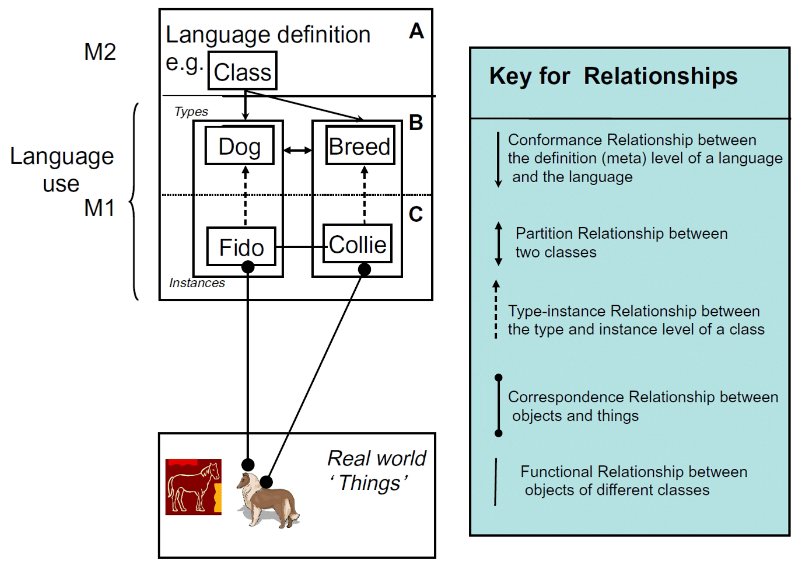
M2
Language definition
e.g. Class
A
Language use
M1
Types
Dog
Breed
B
C
Fido
Collie
Instances
Real world 'Things'
Key for Relationships
Conformance Relationship between the definition (meta) level of a language and the language
Partition Relationship between two classes
Type-instance Relationship between the type and instance level of a class
Correspondence Relationship between objects and things
Functional Relationship between objects of different classes

Figure 11: Reproduction of Fig. 12 from (18)

The key insight, or innovation, that this framework is claimed to offer is that powertypes, such as Breed, no longer need to be regarded as existing at a higher ontological classification level than the type they partition (e.g. Dog) (thereby avoiding the need for multiple-instance-of chains which Eriksson et al. claim to be logically objectionable), but can exist as the same level (level B) thanks to the availability of a new "*partition relationship*". Instances of the partitioned type (e.g. "Fido") can then be linked to instances of the partitioning type (e.g. "Collie") via so-called "*functional relationships*" at the instance level (level C). In addition, a third kind of relationship is needed at the language use level to fully capture the scenario – a type-instance relationship between a domain type (e.g. "Dog") and a domain instance (e.g. "Fido").

While Eriksson et al. claim compliance of their approach to speech act theory and foundational ontologies, we argue that such compliance is not of value in of itself. In contrast, we suggest that the merits of conceptually sound modelling frameworks should be judged by the advantages they offer for practical modelling. In the following, we analytically challenge Eriksson et al.'s proposal with respect to some apparent complications it appears to entail for domain modellers.

5.1. Complex modelling

One significant problem is that Eriksson et al. effectively deny the existence of proper names, such as "Lassie", that are usually assumed to be available to uniquely identify particular objects:

> ... *it is assumed* [by a particular OCA model] *that the dog named Fido has a meaning without there being a dog type – which is impossible* .... (18, p. 2113)

Following Searle (21), Eriksson et al. maintain that (unique) identifiers like "Lassie" or "Fido" are insufficient to identify objects. Instead they claim it is necessary to involve a reference to a substantial class (aka, sortal universal), e.g., "Dog", because the latter supplies an identity criterion that is required for an existential proposition that is always implied when referring to an object, e.g., "Lassie" (18, p. 2107).

We acknowledge that the speech act theory they draw upon to justify this position adopted this particular approach to address certain metaphysical issues. However, we challenge the idea that modelling in a software engineering context benefits from adopting this approach, let alone requires it. The correspondence between a model element such as "Lassie" and the real collie entity it represents is inevitably loose; it escapes a purely formal treatment. However, should the need arise, the model element "Lassie" can be given a unique identifier by which it can be unambiguously referenced and which may also be attached to the real Lassie individual. Unique identifiers are a tried and tested mechanism to reference elements and while they do not solve all metaphysical challenges of referencing entities in the real world the latter is not required in the context of software engineering models. Eriksson et al. may believe that there is merit in mirroring referencing mechanisms required for real world entities in a formal modelling approach, however, we note that they do not provide arguments as to why such a position should be adopted. It is furthermore not clear that a speech act based approach to referencing would be the optimal choice.

Philosophically, using a unique identifier to unambiguously reference a model element could be regarded as using a causal theory of reference based on baptism (32), i.e., one that uses so-called proper names for entities, rather than involving general terms. It could further be argued that philosophers such as Ockham, Buridan, and Kripke favour such a causal theory of reference over approaches that require universals such as "Dog" (32). Ultimately, however, it only matters whether not using a general term (such as "Dog") to reference instances (such as "Lassie") has any detrimental effects in practical modelling. We note that Eriksson et al. do not point out any practical disadvantages resulting from not using their reference approach and conjecture that requiring real world modellers to use Eriksson et al.'s approach would actually complicate matters, making the use of general terms/supertypes mandatory and requiring considerable philosophical sophistication on behalf of modellers as to why a term such as "Dog" provides identity, but a term like "Collie" does not.

5.2. *Unstable Classification*

A consequence of insisting that "Dog" be regarded as the "correct" type for "Lassie" while treating "Collie" as the value of Lassie's "breed" property (18, Tab. 8) appears to be that classification becomes unstable.

What if we switch from considering the pair "Collie" & "Dog" to the pair "Dog" & "Animal"? It appears that now "Lassie" should be considered an instance of "Animal" while "Dog" becomes the value of "Lassie's" "subspecies" property. Just as dog breeds like "Collie" and "Poodle" previously partitioned the "Dog" set, subspecies like "Dog" and "Dingo" would now partition the "Animal" set. This being the case, why is it not true that "Animal" provides an adequate identity criterion for "Lassie" and that belonging to the subspecies of "Dogs" is just as much a property as belonging to the breed of "Collie"? How is a modeller supposed to answer such questions?

Moreover, can a model that only uses "Collie" and "Rough Collie" – where the latter is an even more specific natural type for "Lassie" – be correct or must every model include "Dog" because the latter is the only type that supplies an adequate identity criterion?

It is difficult to see how the absolute claims regarding "Dog" being a substantial type and "Collie" being a moment object can be maintained, given that the respective partitioning argument can always be made with respect to more general types. As long as the same identity criterion is applied, the required assignment of "substantial type" versus "moment object" roles always appears to be a relative, and thus volatile one.

In contrast, if "Lassie" can be regarded as an instance of "Collie" and an (indirect) instance of "Animal" then later discoveries of concepts such as "Living Being" or "Rough Collie" do not invalidate the former facts. Classification in specialisation hierarchies is stable (modulo the most specific type known).

5.3. *Increased Accidental Complexity*

A major concern regarding the use of speech act theory and foundational ontologies as the underpinning for multi-level infrastructures is the fact that models become more complex in several ways.

5.3.1. *Mandatory Supertypes*

Eriksson et al. reject the idea of a "clabject", i.e., an element (e.g., "Collie") that is both an instance (e.g., of "Breed") and a type (e.g., for "Lassie") (6). For example, they regard "Collie" as an instance of "Breed" that cannot function as a type anymore "*since it is already an individual*". (18, p. 2102). Eriksson et al. hence require another element (e.g., "Dog") to act as the type for "Lassie".

As a result, a user-defined model that only requires three elements using deep modelling, requires four elements in language-based approach (c.f. Fig. 6, a replica of (18, Fig. 13)). Making supertypes like "Horse" and "Dog" – that are optional when using deep modelling – mandatory, increases the complexity of user models without any concrete benefits for the modeller.

5.3.2. *Extra Consistency Constraints*

In the language-based approach "Collie" does not inherit attributes from "Dog", as would be the case in a powertype-based approach or in the OCA when "Dog" is used as a supertype for "Collie". A dog's "breedcolour" feature is derived via "Dog", rather than from "Collie", yet the respectively allowed values are defined in "Collie" (c.f. (18, Fig. 16)). This necessitates a consistency requirement:

> *. . . the attribute value Collie* [or "Shetland Pony"] *of the object fido* [or "Prancer"] *must match the restriction represented by the "Collie"* [or "Shetland Pony"] *object in the Breed class.* (18, p. 2116)

We argue that such additional consistency requirements in the language-based approach are symptomatic of the need to synchronise two parts of a model that should not have been separated in the first place. Modellers not only have to deal with more elements, they also need to maintain their consistency. The same scenario within the OCA does not require such additional consistency constraints since it only uses standard object-oriented classification, repeatedly applied over multiple levels. For instance, the check that "Prancer"'s height is indeed within the range allowed by "Shetland Pony" is provided by the standard semantics that the values of an instance must conform to the attributes defined in its type. Eriksson et al. must introduce an extra mechanism because their moment *object* "Shetland Pony" is not capable of playing the role of a type.

5.3.3. Hallmarks of a Workaround

Interestingly, the structure imposed by the language-based approach – with "Horse" & "Breed" considered as types for "Prancer" & "Shetland Pony" respectively (c.f. Fig. 6) – exactly matches the structure of the design patterns "Item-Descriptor" (40) and "Type Object" (41). Just as in these patterns, "Prancer" is not considered an instance of "Shetland Pony", but of "Horse" instead.

The aforementioned design patterns take this approach to facilitate the creation and modification of types like "Shetland Pony" at runtime. In standard object-oriented programming, "Prancer" would be an instance of "Shetland Pony" but because of the desire to have a dynamic type level – supporting the introduction and removal of types at runtime – these design patterns represent "Shetland Pony" as an object that plays the role of a type to "Prancer". As a consequence, "Prancer" and "Shetland Pony" are connected by a "describedBy" (or similar) link instead of a standard, built-in instance-of relationship. Hence, just as in the language-based approach, "Shetland Pony" is modelled as a property of "Prancer". Further emulation machinery is required to support inheritance between types represented as objects (42). All this extra accidental complexity added by the design patterns – a reflection of these workarounds having to capture three domain levels with only two infrastructure levels (26) – is mirrored in the mechanics of the language-based approach: Something that deep modelling captures with just the notions of "clabject" and ontological instantiation, requires the use of "Class", "Type level of Class", "Object Instance level of Class", "Partition", and "Functional" in Eriksson et al.'s approach (c.f. (18, Fig. 16)).

We maintain that the need to use the above variety of notions for multi-level modelling scenarios is a case of *construct redundancy* (43). The more elaborate language-based model is considerably more complex than its (real world) subject in the sense that it includes more elements and employs more different kinds of entities and relationships. The model does not capture any additional information beyond that present in an equivalent deep modelling based model, but requires additional consistency constraints (c.f. Sect. 5.3.2). As a result, the very same criticism that was previously levelled at ageing workarounds such as the "Item-Descriptor" pattern(15; 26), also applies to Eriksson et al.'s language-based approach.

Another perspective on the same point is to realise that in Eriksson et al.'s approach "Shetland Pony" could be regarded as a stereotype for the object "Prancer". Not only does stereotype "Shetland Pony" place "Prancer" into an equivalence class (partition) of all horses that are also Shetland ponies, but it also equips it with additional values (so-called "tagged values" (1)). Stereotypes are generally regarded as a workaround that attempts to provide multiple levels of classification with two-level technology that comes with the price of adding complexity (15).

5.3.4. Summary

In summary, while philosophically motivated attempts to address metaphysical challenges concerning ontological and epistemological aspects of communication as pursued, e.g., by Searle (21) are in general to be welcomed, we see no point in adopting an infrastructure inspired by ideas from speech act theory and foundational ontologies unless there is a promise of concrete benefits for modellers. However, Eriksson et al. do not describe such concretes benefits or point out concrete disadvantages with competing approaches other than the fact that they do not conform to their language-based approach.

Far from improving matters for modellers, therefore, the above analysis would suggest that a language-based approach actually makes matters considerably more complicated than they need be. From a pragmatic stance, we therefore argue that multi-level modelling infrastructures should not attempt to mirror human psychology or conform to a particular school of philosophy just for the sake of it. They should only do so only if there can be an expectation for concrete practical benefits to modellers. As it stands, Eriksson et al.'s proposed solution appears to create more problems than it solves.

6. Conclusion

Given the growing interest in multi-level modelling both in industry and academia, it is important that the premisses, theories and practices used in different multi-level modelling approaches be subjected to rigorous analysis and debate. The recent analyses and criticisms of the deep modelling variant of multi-level modelling (16; 17; 4; 18) are therefore to be welcomed as an attempt to expose weaknesses in the approach and move the general discipline of multi-level modelling forward. However, as pointed out in the introduction, in these recent publications the nature of the challenges raised against deep modelling have changed from observations about syntax and pragmatics to fundamental claims of invalidity and inherent unsoundness. Since such problems, if they were to exist, would render the deep modelling approach unusable as a basis for multi-level modelling and would furthermore have significant implications on many other approaches that are based on some notion of "clabject", it is imperative that they be investigated and if necessary refuted.

In the course of this article we believe we have successfully shown that the claims of invalidity and unsoundness levelled against deep modelling are not applicable to deep modelling. More specifically, we have shown that the arguments put forward by Eriksson et al. are of two basic kinds: The majority of the arguments claiming to reveal "*paradoxes*", "*contradictions*", and "*ambiguities*" in deep modelling suffer from the basic flaw that they are based on assumptions and premisses that are not reconcilable with the published principles of the OCA (Sect. 2). In order to be able to demonstrate any paradoxes and contradictions, Eriksson et al. in fact had to leave the defined OCA paradigm and change some of its basic tenets. Thus, the paradoxes and contradictions that the authors claim to have identified must be rejected as criticisms of the OCA as defined in the literature.

The second group of arguments put forward by Eriksson et al. directly challenge the validity of one of the fundamental tenets of the OCA (the use of deep ontological classification chains) because it does not comply to their reading of one particular philosophy (speech act theory). However, we pointed out that the philosophy used by Eriksson et al. does not necessarily lead to the conclusions that they reach and we refuted claims of the infeasibility of deep classification by referencing respective set-theoretic semantics and definitions from foundational ontologies (Sect. 3).

Nevertheless, the criticisms levelled at deep modelling have helped highlight the importance of creating a clearer description of how deep modelling is founded on mathematical and philosophical principles. The criticisms have also reinforced the fact that for practical modelling, modelling pragmatics are at least as important as philosophical justification. With this in mind, we have complemented our analysis and refutation of the criticisms laid out in the cited papers with two additional sections: one (Sect. 4), describing how the foundations of deep modelling are supported by a large body of widely accepted classical work, including the semantic triangle that lies at the heart of linguistics, and the other (Sect. 5), analysing the pragmatic consequences of the proposal put forward by the authors (assuming the philosophical arguments were sound).

In this latter section we concluded that the alternative framework suggested by Eriksson et al. does not in fact result in any discernible practical benefits for modellers, but on the contrary arguably has some distinct disadvantages. Eriksson et al. focused on explaining the compliance of their approach to various schools of thought but did not present any arguments as to why the resulting approach should give rise to better modelling practice. We contest the notion that compliance to particular schools of thought in itself should be regarded as suggesting the superiority of an approach.

In fact, one of the lessons that can be drawn from the debate about deep modelling is the need for more objective criteria for judging the pragmatic effectiveness of modelling languages. The approach defined by Wand and Weber (43) provides a good starting point, but we believe that further criteria are needed to judge how relevant foundational concepts are to support practical modelling needs. As demonstrated in Sect. 5, the unconditional adoption of ideas developed in a different field without regard for modelling pragmatics can lead to complications for modellers that are not justified by respective benefits.

Although we believe the cited authors' attempts to use speech act theory and foundational ontologies to demonstrate the infeasibility of deep modelling were unsuccessful, we do agree that foundational ontologies contain many concepts, such as roles, phases, etc., that could be usefully incorporated into existing deep modelling approaches as supported by MetaDepth (9), Melanee (12), etc. However, it is a deep question as to which of these concepts should be incorporated into the underpinning infrastructure and which should be provided as predefined ontological libraries in the style of domain-specific libraries (25). We believe this is an important topic for future research.

In summary, far from being unsound and impractical, we believe deep modelling will play a big role in the future of modelling technologies. We also believe it will provide the foundation for unifying semantic representation technologies with model-driven development technologies and thus for increased productivity in many areas of model-driven development. We hope the clarifications and insights laid out in this article will contribute to this process.

[1] Rational, UML specification version 1.1, ad/97-08-11, Rational (Nov. 1997).

[2] J. Álvarez, A. Evans, P. Sammut, 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 (Jan. 2001).

[3] C. Gonzalez-Perez, B. Henderson-Sellers, A powertype-based metamodelling framework, Software and Systems Modeling 5 (1) (2006) 72–90. doi:10.1007/s10270-005-0099-9.

[4] B. Henderson-Sellers, T. Clark, C. Gonzalez-Perez, On the search for a level-agnostic modelling language, in: Proceedings of the 25th International Conference on Advanced Information Systems Engineering, CAiSE'13, Springer-Verlag, Berlin, Heidelberg, 2013, pp. 240–255. doi:10.1007/978-3-642-38709-8_16.

[5] C. Atkinson, G. Grossmann, T. Kühne, J. de Lara (Eds.), Proceedings of the Workshop on Multi-Level Modelling co-located with ACM/IEEE 17th International Conference on Model Driven Engineering Languages & Systems (MoDELS 2014), Vol. 1286 of CEUR Workshop Proceedings, http://ceur-ws.org/, 2014.

[6] C. Atkinson, T. Kühne, Model-driven development: A metamodeling foundation, IEEE Software 20 (5) (2003) 36–41.

[7] T. Asikainen, T. Männistö, Nivel: A metamodelling language with a formal semantics, Software and System Modeling 8 (4) (2009) 521–549.

[8] T. Aschauer, G. Dauenhauer, W. Pree, Multi-level modeling for industrial automation systems., in: EUROMICRO-SEAA, IEEE Computer Society, 2009, pp. 490–496.

[9] J. de Lara, E. Guerra, Deep meta-modelling with metadepth, in: TOOLS (48), 2010, pp. 1–20.

[10] B. Volz, S. Jablonski, OMME – A flexible modeling environment, in: Proceedings of SPLASH Workshop on Flexible Modeling Tools (FlexiTools), 2010.

[11] A. Jordan, G. Grossmann, W. Mayer, M. Selway, M. Stumptner, On the application of software modelling principles on ISO 15926, in: Proceedings of the Modelling of the Physical World Workshop, MOTPW '12, ACM, New York, NY, USA, 2012, pp. 3:1–3:6. doi:10.1145/2491617.2491620.

[12] C. Atkinson, R. Gerbig, Melanie: Multi-level modeling and ontology engineering environment, in: Proceedings of the 2nd International Master Class on Model-Driven Engineering: Modeling Wizards, ACM, 2012.

[13] Y. Lamo, X. Wang, F. Mantz, W. MacCaull, A. Rutle, DPF Workbench: A diagrammatic multi-layer domain specific (meta-)modelling environment, in: R. Y. Lee (Ed.), Computer and Information Science 2012, Vol. 429 of Studies in Comutational Intelligence, Springer, 2012, pp. 37–52.

[14] M. Ltd, Resourcebus – A new substrate for model-driven creations, http://resourcebus.com/ (2014).

[15] C. Atkinson, T. Kühne, Reducing accidental complexity in domain models, Software and Systems Modeling 7 (3) (Springer Verlag, 2008) 345–359. doi:10.1007/s10270-007-0061-0.

[16] B. Henderson-Sellers, On the Mathematics of Modelling, Metamodelling, Ontologies and Modelling Languages, Springer, Berlin, 2012. doi:10.1007/978-3-642-29825-7.

[17] B. Henderson-Sellers, O. Eriksson, C. Gonzalez-Perez, P. J. Ågerfalk, Ptolemaic Metamodelling?: The Need for a Paradigm Shift, IGI Global, 2013, Ch. 4, pp. 90–146.

[18] O. Eriksson, B. Henderson-Sellers, P. J. Ågerfalk, Ontological and linguistic metamodelling revisited: A language use approach., Information & Software Technology 55 (12) (2013) 2099–2124.

[19] OMG, Meta object facility (MOF) specification, OMG document formal/00-04-03, Version 1.3 (Mar. 2000).

[20] C. Atkinson, T. Kühne, The essence of multilevel metamodeling, in: M. Gogolla, C. Kobryn (Eds.), Proceedings of the 4*th* International Conference on the UML 2000, Toronto, Canada, LNCS 2185, Springer Verlag, 2001, pp. 19–33.

[21] J. Searle, Speech Acts: An Essay in the Philosophy of Language, Cambridge University Press, 1969.

[22] J. Odell, Power types, Journal of Object-Oriented Programming 7 (2) (1994) 8–12.

[23] G. Guizzardi, Ontological foundations for structural conceptual models, Ph.D. thesis, University of Twente, Enschede, The Netherlands. (2005). doi:ISBN90-75176-81-3.

[24] N. Guarino, Concepts, attributes and arbitrary relations,, Data & Knowledge Engineering 8 (1992) 249–261.

[25] C. Atkinson, T. Kühne, A tour of language customization concepts, in: M. Zelkowitz (Ed.), Advances in Computers, Vol. 70, Academic Press, Elsevier, 2007, Ch. 3, pp. 105–161.

[26] T. Kühne, D. Schreiber, Can programming be liberated from the two-level style? – Multi-level programming with DeepJava, in: R. P. Gabriel, D. F. Bacon, C. V. Lopes, G. L. S. Jr. (Eds.), Proceedings of the 22*nd* ACM SIGPLAN conference on Object-oriented Programming, Systems, Languages, and Applications (OOPSLA), ACM, NY, USA, 2007, pp. 229–244. doi:http://doi.acm.org/10.1145/1297027.1297044.

[27] C. Atkinson, T. Kühne, Concepts for comparing modeling tool architectures, in: L. Briand (Ed.), Proceedings of the ACM/IEEE 8*th* International Conference on Model Driven Engineering Languages and Systems, MoDELS / UML, Springer Verlag, 2005, pp. 398–413.

[28] T. Kühne, F. Steimann, Tiefe charakterisierung, in: B. Rumpe, W. Hesse (Eds.), Proceedings of Modellierung 2004, LNI (45), GI, 2004, pp. 121–133.

[29] Aristotle, Categories, Internet Classics Archive, MIT, http://classics.mit.edu/Aristotle/categories.html http://www.gutenberg.org/files/2412/2412-h/2412-h.htm (350 B.C.E).

[30] P. Lorenzen, Lehrbuch der konstruktiven Wissenschaftstheorie, Bibliographisches Institut, ISBN 3-476-01784-2, 1987.

[31] OMG, Unified Modeling Language Superstructure Specification, Version 2.1.1, oMG document formal/07-02-05 (Feb. 2007).

[32] P. King, The problem of individuation in the middle ages, Theoria 66 (2000) 195–184.

[33] T. Kühne, Matters of (meta-) modeling, Software and Systems Modeling 5 (4) (2006) 369–385.

[34] C. Atkinson, T. Kühne, Processes and products in a multi-level metamodeling architecture, International Journal of Software Engineering and Knowledge Engineering 11 (6) (2001) 761–783.

[35] C. Atkinson, Meta-modeling for distributed object environments, in: Enterprise Distributed Object Computing, IEEE Computer Society, 1997, pp. 90–101.

[36] C. K. Ogden, I. A. Richards, The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science

of Symbolism, Routledge & Kegan Paul, 1923, the Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism, 1923.

[37] J. F. Sowa, Ontology, metadata, and semiotics., in: B. Ganter, G. W. Mineau (Eds.), ICCS, Vol. 1867 of Lecture Notes in Computer Science, Springer, 2000, pp. 55–81, http://dblp.uni-trier.de/db/conf/iccs/iccs2000.html.

[38] J. Mylopoulos, A. Borgida, M. Jarke, M. Koubarakis, Telos: Representing knowledge about information systems, Information Systems 8 (4) (1990) 325–362.

[39] M. Jarke, R. Gallersdörfer, M. A. Jeusfeld, M. Staudt, S. Eherer, ConceptBase—A deductive object base for meta data management, Journal of Intelligent Information Systems 4 (2) (1995) 167–192. doi:http://dx.doi.org/10.1007/BF00961873.

[40] P. Coad, Object-oriented patterns, Communications of the ACM 35 (9) (1992) 152–159.

[41] R. Johnson, B. Woolf, Type object, in: R. C. Martin, D. Riehle, F. Buschmann (Eds.), Pattern Languages of Program Design 3, Addison-Wesley, 1997, pp. 47–65.

[42] F. D. Lyardet, The dynamic template pattern, in: Proceedings of the Conference on Pattern Languages of Design, 1997.

[43] Y. Wand, R. Weber, On the ontological expressiveness of information systems analysis and design grammars., Inf. Syst. J. 3 (4) (1993) 217–237.