A story of levels

Thomas Kühne

Victoria University of Wellington,
P. O. Box 600, Wellington 6140, New Zealand
Thomas.Kuehne@ecs.victoria.ac.nz

Abstract. Despite being one of the fundamental concepts of multi-level modeling – to the extent of occurring in the name of the discipline – the concept of "level" has no universally agreed upon meaning among multi-level modeling researchers. There is no consensus on what the nature of a level is nor on how levels should be used to organize modeling elements.

In this paper, I aim to initiate a discussion on what the options for defining levels in multi-level modeling are and how they could be systematically characterized.

Keywords: multi-level modeling, level, order

1 Introduction

Multi-level modeling has demonstrably matured from a research idea to an approach of practical significance. The complexity-reducing properties of multi-level modeling [11], have been shown to be applicable in a number of real world models, e.g., 35% of OMG specifications and 20% in the ReMoDD repository [24], and some real world solutions already have been tackled with multi-level technology [27,18,2].

However, there is no discipline-wide consensus on what the term "Level" in "Multi-Level Modeling" represents. The majority of approaches align their "level" concept with the notion of a classification stratum [14,24,6], however with considerable variations. There are furthermore approaches that use broader notions of abstraction between levels, including relationships that are akin to generalization [15,26]. The spectrum continues with approaches where levels do not intrinsically emerge from level content but are defined by association with stakeholders [17], or are even deemed unnecessary [16].

An investigation into the differences of these various interpretations of "level" and what their impact on multi-level modelers is indicated for a number of reasons:

 i) level underpinnings should be made explicit in order to support an informed growth of approaches and to avoid misunderstandings and unwarranted debates between proponents of different approaches.
 ii) a potential consolidation of ideas would lend more strength to particular schools of thought.
iii) trade-off analyses regarding the impact on multi-level modelers should give the latter a way to choose the optimal approach for their application.
iv) leveraging the sanity-checking ability of some level-based well-formedness constraints has significant potential to reduce errors in modeling.

Regarding the last aspect, Brasileiro et al. have shown that taxonomic hierarchies in Wikidata contain an alarming amount of statements that are inconsistent with each other [12]. A staggering 85% of classes in Wikidata participate in what Brasileiro et al. refer to as "Anti-Pattern 1", which uses an illogical combination of classification and generalization to allow conclusions such as "Tim Berners-Lee is an *instance of* Profession", which is an actual example from the analysis. Fragments of knowledge representation that, when combined, allow one to infer such nonsensical conclusions are inadmissible in so-called *level adjuvant* schemes ([4]) that not only use levels as an organizational device but also as a safeguard against ill-formed models [3,5,20,13].

Considering the aforementioned reasons, I therefore could not agree more with Almeida who stated "*it is paramount for multi-level modeling as a discipline to investigate the guiding notion of 'level'*" and with his proposed five point investigation into the nature of levels [1, Sec. 3.2]. Due to space constraints, I will only consider simple linear levels schemes, in particular those used for the ontological dimension of multi-level modeling approaches. The treatment of ω-levels [25] and/or spanning levels [10], or levels schemes that are organized as a lattice are left to future work.

2 Background

The classic OMG four-layer architecture (see Fig. 1(a)) has been a popular subject of study regarding metamodeling principles and also serves an example for how drastically an initially nondescript layer or level hierarchy may change its appearance upon closer examination. Figure 1(a) shows the original depiction of the four-layer architecture that
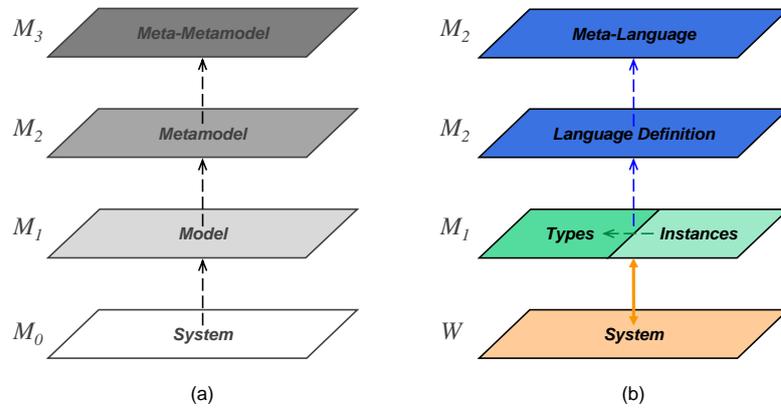
Fig. 1. Original Four-Layer Architecture vs Revised Conceptualization

used three *instance of* arrows of the same kind. A closer examination, however, revealed that three different relationships are involved, of which one (connecting the system under study with the model stack proper) is a representation relationship that often is not related to classification at all [9]. Only the upper two-layer connecting relationships are of the (linguistic classification) kind originally suggested by the OMG.

The first metamodeling principle that aimed at introducing sanity-ensuring rules was the so-called *strict metamodeling* doctrine [5]. The first version of *deep metamodeling* ([6]) chose classification as the relationship between levels and embraced the strictness doctrine, i.e., its levels were separated by *instance of* relationships that were restricted to only occur between two adjacent levels.

Note that the so-called orthogonal classification architecture ([8]), essentially depicted in Figure 1(b), features two (individually strict) orthogonal classification layer structures (formed by linguistic and ontological *instance of* relationships respectively). Both of these hierarchies are therefore implied by the order (as in "classification power") of the elements they host. Elements representing particulars have order zero, elements representing types of particulars have order one, etc. As a result, these early level schemes can be referred to as *order-based*.

3   A Story of Levels

In general, the rationale for employing levels is to achieve a grouping of elements that share some commonality with each other. For instance, in the case of the OMG's four-layer architecture, all elements within a particular layer belong to one language definition or represent usages of one language. For an explicit description of a level-based organization scheme it is therefore helpful to explicitly state a *level cohesion principle* which characterizes why elements are grouped with other elements in the same level.

3.1   Level Cohesion Principle

Cohesion between elements in a level reflects a notion of semantic proximity. For instance, as mentioned before, Henderson-Sellers and Gonzalez-Perez proposed a level scheme whose level cohesion principle is stakerholdership. For schemes that are *order-synchronized* (e.g., MLT [13]), the cohesion principle is order value. For schemes that are *potency-synchronized* (e.g., Metadepth [23]), the cohesion principle is potency value.

A respective grouping of elements implies a segregation of elements (into their disjunct levels) which is considered helpful for

- maintaining an overview by systematically organizing elements, and
- detecting potentially problematic relationships by recognizing them as "level-crossing".

If a level scheme is constructed such that one would only expect one kind of relationship – e.g., *instance of* for order-based schemes – or only few kinds of relationships that are related in nature ([13]), then the occurrence of other kinds of level-crossing relationships could be deemed to flag problematic modeling scenarios that could be in discord with an intended underlying modeling paradigm or are known to create illogical scenarios (cf. Sect. 1).

From the above it follows that an alternative way to describe a level-based organization scheme is to explicitly state its *level segregation principle* which characterizes how elements in adjacent levels are related to each other. Note that while it may initially appear as if a segregation principle and a cohesion principle for the same level hierarchy were just dual formulations of each other, I will later show that this is not the case.

3.2 Level Segregation Principle

In a survey conducted ahead of Dagstuhl seminar 17492, 16 participants (out of 18 respondents) indicated in their response to the question "*What is multi-level modeling?*" that the segregation principle used in multi-level modeling is *abstraction* [21]. Fewer participants (10) phrased their definition in such a manner that allowed to infer the abstraction principle to be *classification*.

Within *order-based* level schemes there is still room for variability regarding the placement of elements in such level hierarchies. Even when a so-called *level-respecting* scheme ([19]) is used – i.e., when within instantiation chains the change in order must exactly correspond to the change in level (Δ order = Δ level) – the exact placement of elements is still not fully determined. The Dagstuhl seminar 17492 working group "Formal Foundations and Ontology Integration" recognized that organizing elements into levels always follows one of either two schemes:

LS_1: element.order = element.level, or
LS_2: element.order $\leq$ element.level [1].

I refer to LS_1 as *order-synchronized* and LS_2 as *order-aligned*.

Figure 2 shows the difference between elements that are synchronized with a level hierarchy based on order – see the elements on the left hand side whose order coincides with the level number – and elements whose order is only *aligned* or *compatible* with the level hierarchy.

The element subscripts in Fig. 2 denote the element order and the box on the right hand side with its associated up and down arrows illustrates that within an *order-aligned* scheme, it is possible to shift a classification ensemble of elements connected in a local instantiation chain up and down the level hierarchy (obeying LS_2).

Note that *strict metamodeling* only requires *order-alignment*, as opposed to *order-synchronisation*. It is a requirement that Δ order = Δ level, i.e., an *instance of* relationship cannot cross two or more level boundaries and the order of elements must change in lock-step with the level hierarchy, but there is no re-

Fig. 2. Order-Aligned Level Scheme

quirement that all order-zero elements must be placed at the bottom of the level hierarchy. Further note that *classic potency* (unlike *characterization potency* [22]) is also *order-locked* in the sense just described.
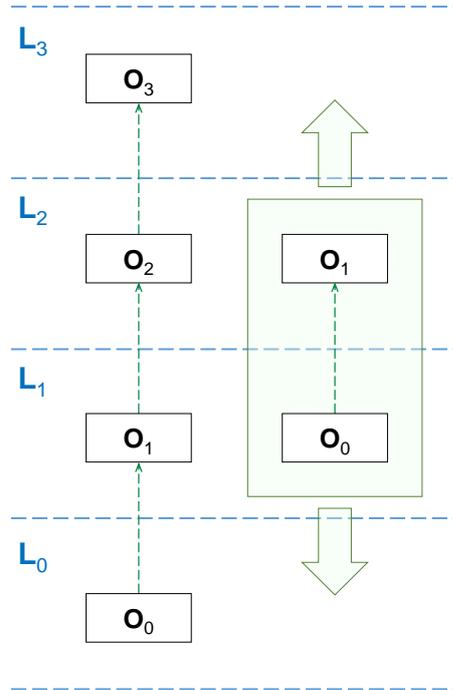
It is important to observe that all aforementioned level schemes imply the same relative level differences between elements. The relative level distance (Δlevel) between related elements – as measured by the difference in level values between elements in the same classification branch – is always the same, independently of the choice of LS_1 or LS_2. The latter, i.e., *order-aligned* schemes, simply enable elements to be optionally shifted up the level hierarchy to any desired height.

As a result, it is adequate to regard *order-synchronized* schemes as reflecting logical classification strata in an absolute sense, whereas the contents of *order-aligned* schemes are always locally equivalent to that of *order-synchronized* schemes but additionally support varying absolute localization.

Due to the fact that *order-synchronization* is simple, easy to formally capture, and references a tried and tested principle of organization, to which *order-alignment* represents a variation that has hitherto not been explicitly motivated, I will focus the following discussion on a comparison of these two *order-based* level schemes.

## 4   Comparing Order-Based Level Schemes

Since both *order-synchronization* and *order-alignment* use the same *level segregation principle* (classification), they both support the same level of sanity-checking, i.e., are equivalent with respect to the kinds of anti-pattern (as employed by Brasileiro et al. [12]) they can be used to detect.

However, *order-synchronization* and *order-alignment* differ in the *level cohesion principle* they use. In *order-synchronization* the cohesion and segregation principles are duals of each other. Any element that is segregated from a reference element is treated as being in cohesion with all other elements that are segregated from the reference element in the same way (here, that have the same level distance to the reference element). Likewise, any two elements that are viewed as being cohesive, are segregated to all other elements in the same manner respectively (here, by the same level distance).

In *order-alignment*, two cohesive elements need not have coincidental segregation properties. Figure 3 illustrates an advantage of the relaxed rules of the *order-alignment* scheme. In this scenario, which features two classification hierarchies – one for activities and one for products (cf. [7]) – it is possible for the product BobsClass to be associated to the activity enactment BobDesigns in a manner that does not require the association to cross a level boundary.

Note that an *order-synchronized* allocation of the elements in Fig. 3 would have required BobDesigns to reside at level zero, thus creating an association that would have crossed a level boundary. As discussed before, a potential advantage of level-based schemes is that they allow potentially problematic relationships to be easily recognized by the fact that they cross level boundaries even though they are not of the kind that gives rise to the *level segregation principle*. With respect to cases like BobDesigns and BobsObject – in which an alignment of separate classification hierarchies can be easily achieved by shifting them against each other – *order-aligned* schemes thus are able to reduce the number of false positives.

This desirable property of *order-aligned* schemes can be justified by elaborating their associated *level cohesion principle* as grouping elements that share semantic prox-
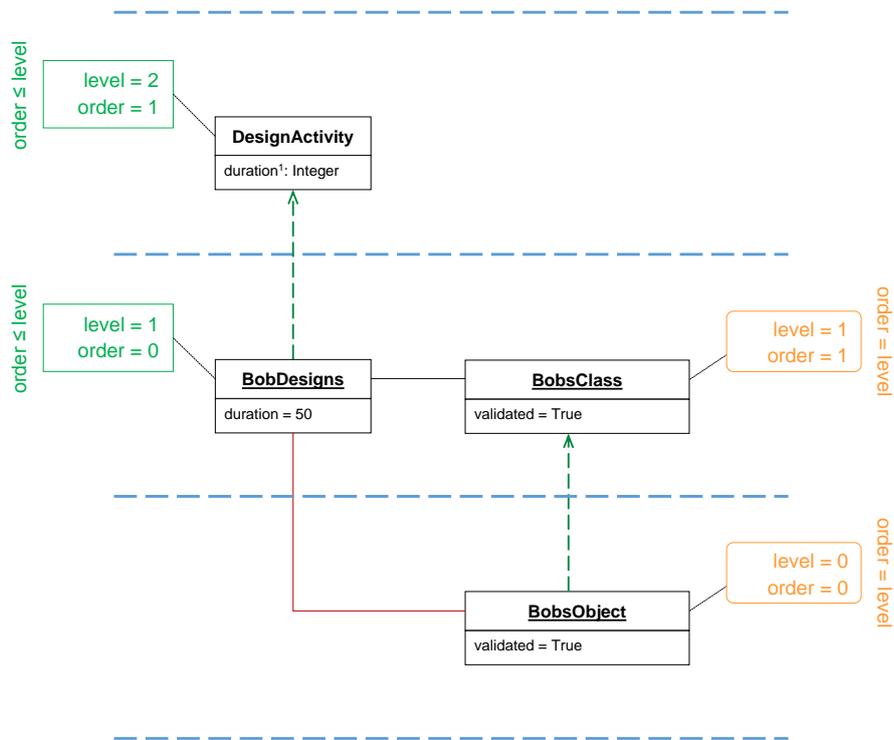
Fig. 3. Cohesion and Non-Alignment

imity – here in the sense of participating in relations as implied by associations – without causing a violation of a strict (cf. [5]) application of a classification *segregation principle*.

In summary, *order-aligned* schemes allow the same sanity-checking capabilities as *order-synchronized* schemes but support a more accommodating *level cohesion principle*. On the one hand, the implicit level allocation of elements in an *order-synchronized* scheme relieves modelers from making placement decisions. On the other hand, the level hierarchy only reflects the *segregation principle* at the expense of not accommodating a richer *cohesion principle*. While *order-alignment* may require manual vertical adjustments to local classification ensembles, the placement of elements could be largely automated by choosing placements that minimize the overall sum of level-crossing relationships.

The optimization principle has to be one of minimization as opposed to complete removal, since it is unfortunately not the case that *order-aligned* schemes can avoid all non-problematic relationships from crossing level boundaries. In Fig. 3, BobDesigns also entertains a relationship with BobsObject which has to reside on a different level in order to respect its local order-level-locking (Δorder = Δlevel) with respect to BobsClass. This means that the association between BobDesigns and BobsObject has to cross a level boundary, making it a "false positive" in the aforementioned sense.

Atkinson and Kühne have argued that diagrams exhibiting such "non-strict" scenarios can be regarded as the superimposition of two locally strict diagrams [7]. Figure 4 shows a different rendering of the elements in Fig. 3, using so-called *modeling spaces* to allow local classification hierarchies to be fully strict, and thus enable the notion that *every* level-crossing relationship within a modeling space does constitute a problematic relationship giving rise to the previously mentioned anti-pattern , i.e. enabling a local "no false positives" principle.

It seems plausible that the perimeters of modeling spaces can be straightforwardly defined based on connected classification ensembles, but the exact details and the empirical validation of the assumed advantages of *order-aligned* schemes over *order-synchronized* schemes remains future work.
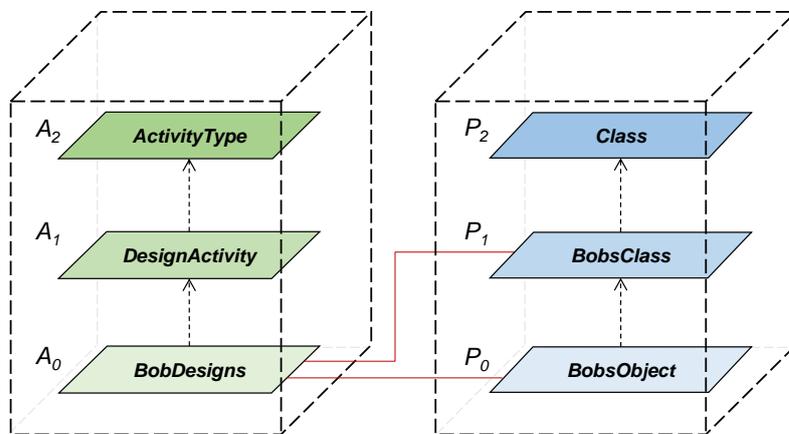

Fig. 4. Local Total Order-Alignment

5 Conclusion

Dagstuhl seminar 17492 working group "Formal Foundations and Ontology Integration" concluded that fundamental notions such as the organization of elements into levels, and the question as to which sanity-checking approaches are expedient, still require further investigation in order to support an explicit understanding and a solid platform for coherent multi-level modeling frameworks [1, p. 23, p. 33].

This paper aimed at taking steps towards addressing these important questions. In particular, Almeida's question: "What does it mean for an entity to be in a 'level'?" [1, Sec. 3.2] can now be answered in two ways based on the framework developed in this paper. One way to answer the question is to state how an entity is *segregated* from other entities in adjacent levels. The second way to answer the question is to state the *level cohesion principle* which groups the entity with other entities in the same level.

I maintain that multi-level modeling research would benefit from each approach having both its *level segregation principle* and its *level cohesion principle* explicitly formulated. For instance, such an explication would avoid criticizing sanity-checking strictness schemes for their apparent inflexibility when their rules are being applied to level schemes whose *level segregation principle* is simply not amenable to strictness enforcing rules. Any multi-level modeling approach with a *level segregation principle* that is akin to generalization will certainly not benefit from enforced strictness. However, it is undoubtedly the case that *order-based* level schemes can significantly benefit from strictness schemes by effectively trivializing the detection of a whole class of anti-patterns.

This paper showed that there is merit in separating a *level segregation principle* from a *level cohesion principle* because there are useful examples where these principles are not simply duals of each other. Instead, a *level cohesion principle* may enjoy a constructive cohabitation with a *level segregation principle*, enriching the latter. I discussed the particular case of *order-alignment*, which has some interesting advantages over *order-synchronization* that deserve further investigation.

I maintain there is promise in identifying more anti-patterns whose detection would reveal serious inconsistencies and illogical constellations in multi-level models. Such anti-patterns could then inform further constraints to be used for tightening level membership principles. However, until such anti-patterns are identified within scenarios enabled by *order-alignment* schemes (compared to *order-synchronized schemes*) it appears to be justified to not prematurely tighten level membership rules.

Acknowledgments

References

1. Almeida, J.P.A., Frank, U., Kühne, T.: Multi-Level Modelling (Dagstuhl Seminar 17492). Dagstuhl Reports 7(12), 18–49 (2018), http://drops.dagstuhl.de/opus/volltexte/2018/8675
2. Aschauer, T., Dauenhauer, G., Pree, W.: Representation and traversal of large clabject models. In: Schürr, A., Selic, B. (eds.) Proceedings of the $12th$ International Conference on Model Driven Engineering Languages and Systems, MODELS 2009, Denver, CO, USA, October 4-9, 2009. pp. 17–31. Springer Verlag (2009)
3. Atkinson, C.: Meta-modeling for distributed object environments. In: Enterprise Distributed Object Computing. pp. 90–101. IEEE Computer Society (Oct 1997)
4. Atkinson, C., Gerbig, R., Kühne, T.: Comparing multi-level modeling approaches. In: Proceedings of the 1st International Workshop on Multi-Level Modelling co-located with the 17th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2014). CEUR Workshop Proceedings, vol. Vol-1286, pp. 43–52 (2014)
5. Atkinson, C., Kühne, T.: Separation of concerns through stratified architectures. International Workshop on Aspects and Dimensions of Concerns, ECOOP 2000, Cannes, France (June 2000)

6. Atkinson, C., Kühne, T.: The essence of multilevel metamodeling. In: Gogolla, M., Kobryn, C. (eds.) Proceedings of the 4th International Conference on the UML 2000, Toronto, Canada. pp. 19–33. LNCS 2185, Springer Verlag (Oct 2001)
7. Atkinson, C., Kühne, T.: Processes and products in a multi-level metamodeling architecture. International Journal of Software Engineering and Knowledge Engineering 11(6), 761–783 (2001)
8. Atkinson, C., Kühne, T.: Model-driven development: A metamodeling foundation. IEEE Software 20(5), 36–41 (Sep 2003)
9. Atkinson, C., Kühne, T.: Rearchitecting the UML infrastructure. ACM Transactions on Modeling and Computer Simulation 12(4), 290–321 (Oct 2003)
10. Atkinson, C., Kühne, T.: Concepts for comparing modeling tool architectures. In: Briand, L. (ed.) Proceedings of the ACM/IEEE 8th MODELS. pp. 398–413. Springer Verlag (2005)
11. Atkinson, C., Kühne, T.: Reducing accidental complexity in domain models. Software and Systems Modeling 7(3), 345–359 (Springer Verlag, 2008)
12. Brasileiro, F., Almeida, J.a.P.A., Carvalho, V.A., Guizzardi, G.: Applying a multi-level modeling theory to assess taxonomic hierarchies in wikidata. In: Proceedings of the 25th International Conference Companion on World Wide Web. pp. 975–980. WWW '16 Companion, International World Wide Web Conferences Steering Committee (2016), https://doi.org/10.1145/2872518.2891117
13. Carvalho, V.A., Almeida, J.P.A.: Toward a well-founded theory for multi-level conceptual modeling. Software & Systems Modeling pp. 1–27 (2016)
14. Fonseca, C.M., Almeida, J.P.A., Guizzardi, G., Carvalho, V.A.: Multi-level conceptual modeling: From a formal theory to a well-founded language. In: Proceedings of the 37th International Conference on Conceptual Modeling (ER 2018). LNCS, Springer Verlag (10 2018)
15. Frank, U.: Multilevel modeling - toward a new paradigm of conceptual modeling and information systems toward. Business & Information Systems Engineering 6(6), 319–337 (2014)
16. Henderson-Sellers, B., Clark, T., Gonzalez-Perez, C.: On the search for a level-agnostic modelling language. In: Proceedings of the 25th International Conference on Advanced Information Systems Engineering. pp. 240–255. CAiSE'13, Springer-Verlag (2013)
17. Henderson-Sellers, B., Gonzalez-Perez, C.: The rationale of powertype-based metamodelling to underpin software development methodologies. In: Proceedings of the 2Nd Asia-Pacific Conference on Conceptual Modelling - Volume 43. pp. 7–16. APCCM '05, Australian Computer Society, Inc. (2005), http://dl.acm.org/citation.cfm?id=1082276.1082278
18. Igamberdiev, M., Grossmann, G., Selway, M., Stumptner, M.: An integrated multi-level modeling approach for industrial-scale data interoperability. Software & Systems Modeling pp. 1–26 (2016)
19. Kühne, T.: Matters of (meta-) modeling. Software and System Modeling 5(4), 369–385 (2006)
20. Kühne, T.: Contrasting classification with generalisation. In: Sixth Asia-Pacific Conference on Conceptual Modelling, APCCM. pp. 71–78 (2009)
21. Kühne, T.: What is multi-level modeling? Dagstuhl 17492 Presentation, Shared Documents of Dagstuhl Seminar 17492 (December 2017), https://materials.dagstuhl.de/files/17/17492/17492.ThomasK%C3%BChne4.Slides.pdf
22. Kühne, T.: Exploring potency. In: ACM/IEEE 21st International Conference on Model Driven Engineering Languages and Systems (MODELS '18). ACM (2018)
23. Lara, J., Guerra, E., Cuadrado, J.S.: Model-driven engineering with domain-specific metamodelling languages. Softw. Syst. Model. 14(1), 429–459 (Feb 2015)
24. Lara, J.D., Guerra, E., Cuadrado, J.S.: When and how to use multilevel modelling. ACM Transactions on Software Engineering and Methodology 24(2), 12:1–12:46 (2014), https://doi.org/10.1145/2685615

25. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: Representing knowledge about information systems. Information Systems 8(4), 325–362 (1990)
26. Neumayr, B., Schuetz, C.G., Jeusfeld, M.A., Schrefl, M.: Dual deep modeling: multi-level modeling with dual potencies and its formalization in f-logic. Software & Systems Modeling pp. 1–36 (2016)
27. Rossini, A., de Lara, J., Guerra, E., Nikolov, N.: A comparison of two-level and multi-level modelling for cloud-based applications. In: Proceedings of ECMFA 2015. pp. 18–32. LNCS 9153 (2015)