Comparing Multi-Level Modeling Approaches

Colin Atkinson[1], Ralph Gerbig[1] and Thomas Kühne[2]

[1] University of Mannheim,
B6, C2.11, Mannheim, Germany
(atkinson, gerbig)@informatik.uni-mannheim.de
[2] Victoria University of Wellington,
P. O. Box 600, Wellington 6140, New Zealand
Thomas.Kuehne@ecs.victoria.ac.nz

Abstract. As the range of modelling approaches that claim to be "multi-level" diversifies, there is growing debate in the literature about what multi-level modelling actually is and what form supporting languages and infrastructures should take. However, there is no consensus yet on how this debate should be framed and what objective criteria should be used to evaluate different approaches. It is clear from the literature that proponents of different approaches base their arguments on fundamentally different assumptions about what multi-level modelling is and what benefits it should aim to provide. In this position paper we identify some of the core issues that currently hinder progress towards the required consensus and identify some of the terminological differences that have amplified confusion. Referencing various work that represents diverging viewpoints, our goal is to initiate a meta-discussion on what the open questions in multi-level modelling are, how respective proposals to answer them could be evaluated, and which kinds of discussions are expedient in this context.

Keywords: multi-level modelling, deep modelling, metamodelling, level-agnostic

1 Introduction

Multi-level modelling is gaining resonance. Many groups have applied the approach or created variants, and a community focusing on multi-level modelling is emerging. However, at the present time, there is little consensus in the literature on fundamental multi-level modelling concepts, and if anything proponents of multi-level modelling appear to be diverging rather converging in their understanding of the approach. In particular, in a series of recent papers [9,12,13] some authors have presented a long list of fundamental criticisms of one of the first proposed multi-level modelling approaches [6], based on the notions of the orthogonal classification architecture and deep instantiation. Many other authors have also identified issues in this and other approaches to multi-level modelling and have suggested their own solutions (e.g., [10,18]).

An analysis of these papers reveals that the criticism ranges from challenges to the fundamental validity of the approach from a set-theoretic point of view to objections to the use of particular terminology and disagreements about the basic goals and motivations underlying multi-level modelling.

We believe that some of the disagreements can be resolved by adopting a different style of debate. Hence, we argue that the main challenge facing the fledgling multi-level modelling community at the present time is to clarify the boundary conditions and assumptions within which the debate about multi-level modelling should take place. Without consensus on these meta-issues, the chances of forging a common understanding about fundamental concepts in multi-level modelling will be small.

In order to foster greater clarity in the debate between multi-level modelling approaches, in this paper we attempt to identify some of the main open questions in multi-level modelling and suggest potential ways of converging towards broadly accepted answers. In particular, we aim to characterize what kinds of discussions are most likely to increase convergence and suggest certain principles to be used in evaluating different proposals.

The rest of the paper is organised as follows. To help set the context for the discussions, the next section briefly outlines some core features we believe could be used to characterized multi-level and deep modeling approaches. Section 3 continues by describing five key issues with the potential to cause controversy in the context of multi-level modeling. Section 4 then provides a deeper consideration of one of the most subtle and sensitive of these issues – "terminology" – with a specific focus on the term "level-agnostic" which is the subject of some debate at the present time. Finally, section 5 concludes with some final observations and closing remarks.

2 Multi-Level Modelling

Perhaps the most fundamental and important question is what qualities an approach needs to posses in order to be characterized as multi-level. Without a consensus on how to recognize multi-level modeling approaches, it will be impossible to conduct a meaningful debate about their relative strengths and weaknesses. We suggest that minimal requirements for a multi-level modelling approach include:

- some fundamental notion of abstracting a multitude of model elements to a common classifier.
- the ability for model elements to form anti-transitive instantiation chains.
- a concept of level, formed by elements belonging to the same classification level, the latter being defined by both classification depth and level membership of the root of the instantiation chain.

The above minimal requirements are very inclusive and admit, for instance, traditional linguistic language definition stacks. We prefer to take a narrower view that furthermore:

- mirrors classification relationships in the domain with explicit relationships that are subject to well-formedness constraints [15].
- recognises type and instance facets of model elements and views them as inseparable [2].
- provides a mechanism for deep characterisation [5].

The approach containing the combination of all the above characteristics is often referred to as "deep (meta-)modelling". This is the particular flavour of multi-level modelling that has been the subject of the most debate in the recent literature [9,12,13] and we used some of this criticism to identify a number of potentially controversial issues that we discuss in the following section.

3 Potentially Controversial Issues

In order to move the field of multi-level modelling forward, it is of course desirable to have many alternative proposals and a healthy debate about their respective merits. There are a few pitfalls, however, that should be avoided in order to achieve progress as constructively as possible.

For example, we believe that subscribing to a particular "*school of thought*" and exclusively evaluating differing proposals from one subjective perspective can be problematic. Some examples of controversies that can emerge when issues have been evaluated with this mindset are discussed below.

3.1 Language Size

Always expecting a user to first define or choose their language [19] versus presenting a user with a rich library of modelling concepts to be adapted and used [6] represent the two different ends of a language engineering spectrum. No single point within this spectrum will be optimal for all types of users but we believe there is an interesting discussion to be conducted about the level of support tools aimed at the majority of modellers should provide with respect to language engineering.

Which modellers can be expected to be good language engineers and which library paradigms may turn out to be too narrow in the assumptions they make? A complete tool should recognise both language engineering and domain modelling as relevant tasks but there is certainly room for specialised tools that focus on one of these areas only. The potential pitfall to avoid is to assume that all user modelling is language engineering or that all language engineering can be subsumed under domain modelling.

3.2 Semantics

The clabject-based deep instantiation approach of Atkinson and Kühne has been criticised for lacking alignment with set theory and requiring the instantiation of elements that are not available for further instantiation [9]. While we do not dispute that there are formalisations and school of thoughts in which deep instantiation can be seen as "wrong", there is indeed a sound set-theoretic formalisation of deep instantiation [17] and in this framework – based on sets of sets – it is perfectly possible and natural to view elements as instances and types at the same time, with corresponding linear instantiation chains arising from this property.

We believe that instead of evaluating approaches according to whether or not they have compatible foundations, it is more helpful to examine whether approaches are internally consistent. If an approach is inconsistent to the effect of exhibiting contradictions or allowing unwanted paradoxes to occur then it could be viewed as "wrong". Non-conformance to a particular semantic foundation, however, should not be held against an approach per se. Thus, while conformance to established disciplines can be a potential advantage, it cannot be the ultimate criterion for determining the adequacy of a proposal.

It would of course be desirable to obtain a common sound formal foundation for all multi-level modelling approaches in order to move the technology forward, but until consensus on such a formalism has been reached it is unhelpful to assume one particular approach as being a benchmark to evaluate other approaches against. In fact, judging an approach using an inappropriate perspective and formalism can lead to claims of unsoundness and inconsistencies which are not in fact valid [7].

3.3 Intended Target Audience

Whether consciously or unconsciously particular approaches target different user groups. For instance, while Henderson-Sellers et al. appear to focus on making it easy to build tools [12], the original multi-level approach by Atkinson and Kühne [4] focused on reducing accidental complexity for the domain modeller. On the other hand, Vangheluwe et al.'s AtomPM tool appears to be geared towards language engineering [19], etc.

A considerable amount of debate can be avoided if one takes the intended target audience of a particular approach into consideration. For example, the apparent difference in attributing significance to domain-motivated (ontological) instantiation relationships between Henderson-Sellers et al. and Atkinson et al. can be understood as reflecting the different target audiences. When focusing on the internals of a tool, i.e., addressing a tool builder audience, it is natural to regard user-defined relationships and associated modelling patterns as secondary [12]. In contrast, work on the orthogonal classification architecture [4] or in particular the "Unified Modelling Library" [6], targets the modeller and aims to provide a richer environment in order to reduce accidental complexity.

As an analogy, it is simpler to write a compiler for an assembly language compared to a software engineering language like Eiffel [20], however it can be argued that the investment necessary to develop the far more complex Eiffel compiler will pay dividends in subsequent safe language usage. We believe that "*everything is an object*" [12] is mainly a tool builder's argument and is less helpful when aiming to support modellers.

3.4 Level of Modelling Discipline

Related to the above discussion regarding intended target audiences, there appear to be diverging views on what the intended target audience is. Some work appears to take a liberal approach, allowing concise and powerful solutions but also giving users rope to hang themselves [12], while other work attempts to enforce a discipline that is aimed at helping the user avoid inconsistent models [3].

We claim that is not ideal to point out perceived problems of disciplined approaches while not acknowledging the dangers that are implied by looser approaches. Strict metamodelling [3], for instance, has been criticised as being too restrictive and causing problems [10,12].

First, we believe that a lot of the criticism towards strict metamodelling is based on misunderstandings, e.g., the lack of awareness that new elements can always be introduced linguistically (i.e., without requiring an ontological type), that potency declarations are to be regarded as constraints that make certain guarantees but are not an enabler for deep instantiation, and that potency constraints do not cross instantiation dimensions, e.g., from the linguistic dimension to the ontological dimension, etc.

Second, we suggest that the debate is comparable to the debate about the value of static typing in programming languages. In the same way that some research in programming languages strives to make types and type checking available when needed [8] based on ideas of optional type inference, it seems promising to adopt a similar approach to modelling. In particular, since there is no extra effort involved in adhering to strict metamodelling (since levels can be inferred [15]) we suggest to support optional sanity checking on the basis of strict metamodelling principles.

3.5 Terminology

Terminology, i.e., the choice of words and their meaning, is always a potential source of confusion and unnecessary (or undiscovered) disagreement. As an example, the fact that the name of a particular, domain-oriented, instantiation type includes "ontological" should not be construed to mean that respective models in so-called ontological levels need to exhibit "*all the trappings of an ontological (meta)model such as the UFO*" [12].

The particular naming choice involving "ontological" was made because the original meaning of "ontology" relates to the things that exist (in the sense of a universe of discourse), in contrast to (linguistically classified) notation elements. There was no intention to invoke any of the features associated with contemporary ontology research.

We believe terminology can have various degrees of intuitiveness but effectively only the explicitly given definition of concepts and mechanisms should be decisive when it comes to criticising an approach.

A good example of an increasingly used term that can be interpreted in diametrically opposed ways is "level-agnostic". In the following, we clarify these different interpretations.

4 Level-Agnostic Languages

The adjective "level-agnostic" implies that an approach does not make the treatment of an element dependent on its level in the ontological classification hierarchy. In this sense, the UML is definitely not a level-agnostic language since it uses a different notation for objects compared to classes, even though arguably all objects and classes are just modelling elements with the only difference being that classes have a type facet whereas object do not.

The idea of level-agnosticism for a modelling language as discussed in this context can be traced back to the proposal of using a uniform notation for UML elements, independently of which level they belong to, e.g., whether they are objects or classes [2]. This idea has later been expanded to include further notational elements [1].

A language can achieve level-agnosticism in one of two ways. One way is to essentially only recognise a single level [12] which internally can support arbitrary constellations of ontological classification relationships. We refer to such languages as "level-blind". The other way to support level-agnosticism is to use levels as a structuring and soundness-enforcing mechanism, without letting level membership impact on such things as element representation and rendering. We refer to such languages as "level-adjuvant".

4.1 Level-Blind Languages

Since the term "level-agnostic" implies the existence of levels, a "level-agnostic language" has to acknowledge them in some sense, even the approach put forward in [12]. Otherwise the described "search for a level-agnostic language" should have been characterised as a search for a "*level-less*" language. The characteristic feature of a level-blind language is that it essentially ignores the fact that there are levels.

An example for level boundaries that are considered insignificant in the level-blind approach of [12] are the ontological classification level boundaries implied by elements in a universe of discourse, such as "Lassie", "Collie", and "Breed" (c.f. section 3.3). In contrast, Atkinson and Kühne give the respective ontological classification hierarchy the same significance as the time-honoured linguistic classification hierarchy formed by language metamodelling [4].

The advantage of an approach that is blind to the ontological level boundaries is that it allows all elements to be treated the same, since membership to ontological levels is abstracted away from. The disadvantage, however, is that element membership to ontological levels can no longer be exploited to uncover unsound scenarios. Paradoxical situations like being one's own baby [14, p. 247] are hallmarks of single level, flat domain approaches where unification has been taken to the extreme. If, for instance, everything is a set, as in naïve set theory, then even the set of all sets that do not contain themselves is a set. Yet this construction establishes Russel's famous paradox, as it is not possible to find a single consistent answer to the question whether the said set is a member of itself.

An analogous construction is obviously possible with the "*everything is an object*"-approach put forward in [12]. Consider an object O whose instances are all those objects that are not instances of themselves. This leads to the paradoxical situation that O must be an instance of itself when it is not an instance of itself and must not be an instance of itself when it is an instance of itself. Such paradoxes are impossible in a language that uses levels to avoid self-reference. Stratification was one of the proposals to fix naïve set theory and works just as well when used in the form of ontological levels forming a domain-classification hierarchy.

4.2 Level-Adjuvant Languages

A level-adjuvant language is a level-agnostic language that recognises the utility of levels, without implying accidental differences in treatment. For instance, the orthogonal classification architecture [4] uses ontological level boundaries to avoid paradoxical modelling scenarios but does not require different treatments with respect to representation or rendering (c.f. [5, Fig. 16, p. 307]). The orthogonal classification architecture hence establishes level-agnosticism with respect to representation and notation but is not level-blind with respect to enforcing soundness for models of the universe of discourse.

The difference between using level-blind versus level-adjuvant languages is like the difference between untyped programming languages versus strongly typed programming languages. While the "anything goes" approach of untyped languages supports small but powerful solutions, the discipline attained by typing rules pays big dividends when unsound scenarios are rejected straight away as opposed to being discovered through testing, or worse, not uncovered at all (c.f., section 3.4).

For instance, the "everything (even a class) is an object" approach [12, Fig. 10] that was first used as the backbone of the Smalltalk metaclass hierarchy [11], allows an object o to be its own class or to be an instance of another object c and a subclass of c at the same time. Unless such scenarios are excluded by suitable well-formedness rules, it is possible to construct paradoxical situations and "sillygisms" [16]. The discipline afforded by requiring an ontological classification relationship to be acyclic and level-respecting [15] is exactly designed to exclude such problems.

It is ironic that the proponents of a single-level, flat domain approach [12] that takes unification to the extreme, criticise level-aware approaches for enabling paradoxes even though level-awareness is the key to avoiding a whole class of paradoxes that naturally occur in single-level, flat domain approaches.

5 Conclusion

As multi-level modelling grows as a research discipline, there is hope that all groups involved will be able to converge on some universally acceptable concepts and ideas. We are by no means suggesting that all current and future approaches should be brought into line with a single view, though. On the contrary, variety and lively discussions are obviously important to further any new discipline. Nevertheless, unless agreement can be found on a number of fundamental questions, the discipline and its community will struggle to grow cohesively.

In this paper we therefore suggested that future debates should be not be held from subjective standpoints using a particular school of thought as a frame of reference. Observing incompatibilities between different approaches can be a starting point for fruitful discussions, but cannot be used as a basis for claims of other approaches being "wrong".

Moreover, we propose that in the vast majority if cases different approaches should not be judged as "right" or "wrong" but as "better" or "worse". The intention behind this proposal is not just to use relative terminology versus absolute terminology, it more crucially also involves different evaluation criteria. In our view, the merits of an approach should be judged on what benefits and/or disadvantages it brings to the intended target audience. In other words, instead of using subjective "schools of thought" as a reference, we propose to use pragmatics as a deciding criterion. Who exactly is the target audience? How does the target audience benefit from a certain philosophical underpinning in a concrete manner? What are the extra tools and/or opportunities given to a user? How do different approaches compare in terms of model maintenance? We believe that these are some of the crucial questions that should be asked when debating divergent viewpoints.

Empirical evaluations are notoriously hard to conduct, but debates on the merits of a certain approach can also be resolved theoretically. As long as arguments and comparisons are made with an explicitly stated target audience in mind, we believe that the discipline of multi-level modelling can converge and future users can benefit from this new, emerging sub-discipline of model-driven development.

References

1. Atkinson, C., Kennel, B., Go, B.: The level-agnostic modeling language. In: Malloy, B., Staab, S., Brand, M. (eds.) Software Language Engineering, Lecture Notes in Computer Science, vol. 6563, pp. 266–275. Springer Berlin Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-19440-5_16
2. Atkinson, C., Kühne, T.: Meta-level independent modeling. In: International Workshop Model Engineering (in Conjunction with ECOOP'2000). Springer Verlag, Cannes, France (Jun 2000)
3. Atkinson, C., Kühne, T.: Profiles in a strict metamodeling framework. Journal of the Science of Computer Programming 44(1), 5–22 (Jul 2002)
4. Atkinson, C., Kühne, T.: Model-driven development: A metamodeling foundation. IEEE Software 20(5), 36–41 (Sep 2003)
5. Atkinson, C., Kühne, T.: Rearchitecting the UML infrastructure. ACM Transactions on Modeling and Computer Simulation 12(4), 290–321 (Oct 2003)
6. Atkinson, C., Kühne, T.: A tour of language customization concepts. In: Zelkowitz, M. (ed.) Advances in Computers, vol. 70, chap. 3, pp. 105–161. Academic Press, Elsevier (June 2007)
7. Atkinson, C., Kühne, T.: In defence of deep modelling. submitted for publication (2014)
8. Bracha, G., Griswold, D.: Strongtalk: Typechecking smalltalk in a production environment. In: Proc. of the ACM Conf. on Object-Oriented Programming, Systems, Languages and Applications (1993)
9. Eriksson, O., Henderson-Sellers, B., Ågerfalk, P.J.: Ontological and linguistic metamodelling revisited: A language use approach. Information & Software Technology 55(12), 2099–2124 (2013)
10. Gitzel, R., Merz, M.: How a relaxation of the strictness definition can benefit MDD approaches with meta model hierarchies. In: Proceedings of the 8^{th} World Multi-Conference on Systemics, Cybernetics and Informatics. vol. IV, pp. 62–67 (July 2004)
11. Goldberg, A., Robson, D.: Smalltalk-80: The Language and its Implementation. Addison-Wesley, Reading, MA (1983)
12. 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, Berlin, Heidelberg (2013)
13. Henderson-Sellers, B., Eriksson, O., Gonzalez-Perez, C., Ågerfalk, P.J.: Ptolemaic Metamodelling?: The Need for a Paradigm Shift, chap. 4, pp. 90–146. IGI Global (2013)
14. Jackson, D.: Software Abstractions: Logic, Language, and Analysis. The MIT Press, Cambridge, Mass. (Apr 2006)
15. Kühne, T.: Matters of (meta-) modeling. Software and Systems Modeling 5(4), 369–385 (2006)

16. Kühne, T.: Contrasting classification with generalisation. In: Sixth Asia-Pacific Conference on Conceptual Modelling, APCCM. pp. 71–78 (2009)
17. Kühne, T., Steimann, F.: Tiefe charakterisierung. In: Rumpe, B., Hesse, W. (eds.) Proceedings of Modellierung 2004. pp. 121–133. LNI (45), GI (Mar 2004)
18. de Lara, J., Guerra, E.: Deep meta-modelling with metadepth. In: TOOLS (48). pp. 1–20 (2010)
19. Mannadiar, R.: A Multi-Paradigm Modelling Approach to the Foundations of Domain-Specific Modelling. Ph.D. thesis, McGill University (2012)
20. Meyer, B.: EIFFEL the language. Prentice Hall, Object-Oriented Series (1992)