

Calling a Spade a Spade in the MDA Infrastructure

Colin Atkinson
University of Mannheim
68161 Mannheim, Germany
atkinson@informatik.uni-
mannheim.de

Thomas Kühne
Darmstadt University of Technology
64283 Darmstadt, Germany
kuehne@informatik.tu-
darmstadt.de

Introduction

Metamodeling has a pivotal role to play in the realization of the MDA. It is therefore essential that the MDA community establish a clear and sound view of what metamodels are, what purposes they serve, and what form they consequently should take. A lot of progress has been made in this direction in recent years, but there are still some fundamental issues that need to be sorted out.

In our view, one of the most fundamental problems is that the role of metamodeling in the MDA approach is generally looked at from one angle only. The “accepted wisdom” at the moment is that the role of metamodeling is to support language definition and/or extension. This “metamodeling is language definition” view is found in the preambles to many of the UML/MOF related OMG documents, and is even implicit in this workshop’s Call for Papers.

We do not challenge the fact that one of the most important functions of metamodeling in MDA is to support the definition of languages. However, we believe that characterizing language definition as the *only* role of metamodeling is overly simplistic and ultimately detrimental to the evolution of MDA technology. To do justice to metamodeling and properly characterize its role we need to go back to the foundations of “modeling” and “meta-ness” and analyze what their integration means.

Calling a Spade Something Else

At the outset of this discussion we need to point out, of course, that the meaning of a term is determined by a common understanding within the community using it. Thus, if the MDA community (and in particular the OMG) chooses to define “metamodeling” as language definition, and to characterize other modeling activities involving meta-ness as “not metamodeling”, then it is perfectly free to do so. And that would be the end of the debate. But before we go down this route we should be sure that

- a) such a definition makes sense, and serves to promote rather than hinder the development of MDA technology.
- b) such a definition is deliberately selected in preference to other possible definitions with a full awareness of the pros and cons.

At present we do not believe that either of these requirements holds. In particular, we do not think that characterizing metamodeling as just language definition does justice to it, and we do not think that this characterization is becoming popular because it is the best, but because the full spectrum of metamodeling is not being recognized.

In the spirit of calling a spade a spade, in the remainder of this position paper we try to present a complete characterization of metamodeling for MDA.

Form versus Content

Whenever human beings wish to communicate (whether in written or spoken form) they make statements in some kind of language. Usually communication concerns things and facts from reality but since language is part of that reality it is possible to make statements about (the use of) language as well. Consider the element “spade” in following sentence, for example.

“You should call a spade a spade”

Taking a grammatical or linguistic viewpoint, one would argue that “spade” is a “noun” and that the first occurrence of “spade” plays the role of a “subject”. Hence, “noun” and “subject” are *linguistic* classifiers for “spade”. Taking a semantic or ontological viewpoint, however, the term “spade” can be understood to refer to a certain type of tool or to an iconic symbol found on playing cards. Hence, “tool type” and “card identification mark” are *ontological* classifiers for “spade”.

This already gives us four classifiers for “spade”, and it is easy to come with a whole host of others. However, all the classifiers come in one of two fundamental flavors. The first kind of classification deals with the *form* of the statement element (what it is when we *mention* it), while the second deals with the *content* of the statement element (what it is when we *use* it). These two fundamental dimensions of classification [1] exists whatever kind of statement we are dealing with. In the case of a visual model (which is just a statement expressed in a graphical rather than a textual style), the linguistic (meta-) dimension deals with the classification of model elements according to their form (e.g. Class, Association, Attribute) and the ontological (meta-) dimension deals with the classification of model elements according to their content

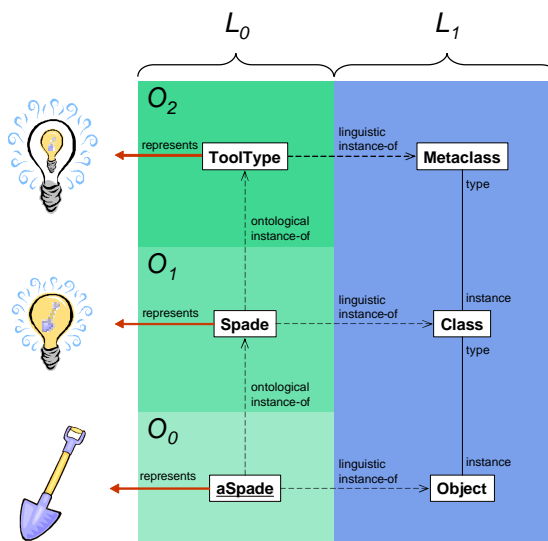


Fig. 1 Two metamodeling dimensions

the linguistic (meta-) dimension deals with the classification of model elements according to their form (e.g. Class, Association, Attribute) and the ontological (meta-) dimension deals with the classification of model elements according to their content

(e.g. ToolType). In Fig. 1 the real world is shown in the leftmost column, a model of it in the middle column (L_0), and a language definition for the model element vocabulary in the rightmost column (L_1). Thus, the linguistic meta-boundary runs vertically between L_0 and L_1 , whereas the ontological meta-boundaries run horizontally (within L_0).

Perspective and Perception

Clearly the ontological classification of types has just as much “metaness” about it as the linguistic classification of types, yet the term “metamodeling” is typically reserved for “linguistic metamodeling” only. One could argue that the O_n level hierarchy is just another flavor of linguistic metamodeling by viewing the O_n level as defining a language to be used in O_{n-1} . But this would imply that adding class “Spade” to level O_1 amounts to “extending the language for level O_0 ”. The central question therefore becomes:

“When an object-oriented programmer defines a new class, is he/she extending the language?”

The usual answer is “no.” When programmers write classes they usually don’t think of themselves as extending the programming language, but rather as *using* the language. In fact, one of the key ideas that (especially Smalltalk) programmers must learn is that not all fundamental concepts are captured in the core language definition, but are provided as part of a standard library. Likewise any addition to one of the O -levels in Fig. 1 is not a language extension, but a use of the language defined in L_1 .

Still, even though adding new types to level O_1 does not correspond to language extension, it is clear that metamodeling is taking place. Just as “Class” corresponds to the set of all classes (e.g., “Spade”, “Person”, etc.), “ToolType” corresponds to the set of all tool kinds (e.g., “Spade”, “Hammer”, etc.). It is, thus, unjustified to characterize any extension of L_1 as “metamodeling” and any extension of O_2 as “not metamodeling”.

It appears that perspective plays an important role in characterizing model extensions as “meta-modeling” or not. Obviously, tool builders and members of standard consortiums take it for granted that L_1 extensions constitute “meta-modeling” exclusively. Yet from the perspective of a modeling language’s user, the type hierarchy formed by the O -levels is much more relevant. In other words, ontological metamodeling is “user (content) metamodeling” and linguistic metamodeling is “standard (form) metamodeling”.

Conclusion

Ontological metamodeling does not depend on the existence of an explicit O_2 level, but actually is alive and well in the UML today. However, rather than being supported by a natural extension of the existing O_0 — O_1 levels, it is implicitly supported via stereotypes and profiles. The effect is the same, but the metamodeling character of

profile creation is either suppressed or misleadingly cast as a “language extension activity”. In effect, metamodeling other than for language definition goes on all the time, but the predominant “standard definition” perspective reserves the term “metamodeling” exclusively for its own purposes.

However, at the end of the day “standard definition” is a means to an end and not an end to itself, so it is the user’s perspective which should be predominant. Therefore, user metamodeling should be recognized as such and be cleanly supported instead of forcing all the baggage associated with stereotypes, tagged values and the rest of profile paraphernalia on users. Important MDA techniques such as type level transformations in both “framework based” and “marking mechanism” versions [2], actually call for ontological metamodeling support.

We believe that if the current unbalanced view of metamodeling continues, the evolution of MDA technology will be stifled and the full potential of metamodeling will not be fulfilled.

REFERENCES

1. Colin Atkinson and Thomas Kühne, Rearchitecting the UML Infrastructure, ACM journal "Transactions on Modeling and Computer Simulation", Vol. 12, No. 4, 2002
2. David S. Frankel, Model Driven Architecture: Applying MDA to Enterprise Computing, OMG Press, 2003