Rearchitecting Meta-modeling Foundation: A Formal Approach

P. Jamshidi, F. Shams, S. Khoshnevis, R. Teimourzadegan, A.Nikravesh
Automated Software Engineering Research Group
Electrical and Computer Engineering Faculty
Shahid Beheshti University
{p_jamshidi, f_shams, s_khoshnevis, r_teimourzadegan, a.nikravesh}@sbu.ac.ir

Abstract

Model-Driven Development (MDD) is based on the premise of giving software engineers the most suitable infrastructure, based on which they can easily construct and transform their artifacts. Several paradigms like linear four-layer infrastructure have been proposed to facilitate MDD requirements. Although recently proposed paradigms have their own advantages and superiority to the traditional paradigms, there is no formal foundation, by which they could be described. A category can be seen as a structure that formalizes a mathematician’s description of a type of structure. Therefore, category theory is convenient for describing paradigm of the infrastructure. In this position paper, we provide a formal framework consisting categories of models to rearchitecting traditional linear meta-modeling paradigm to a category-based one. By adopting the paradigm, the dual classification problem arising from the need to capture both linguistic and ontological classification of model elements, could be formally clarified.

Keywords: model-driven development, meta-modeling foundation, category theory, multiple classification problem

1. Introduction

Abstract concepts, notation depicting them, representation of real world elements [1] are the most fundamental requirements of the MDD, which have been directly addressed by visual modeling. Concepts to facilitate dynamic user extension to the model concepts, interchanging of them, and transformation between them [1], are the complementary requirements, which should be addressed by abstraction and meta-modeling mechanisms. The traditional four-layer infrastructure as a linear paradigm has been proposed by OMG to underpin the first generation of MDD foundation. Although it has been a successful foundation [1], this traditional infrastructure does not scale up well to handle all technical requirements described earlier [1, 2, 3]. Using a single instance-of relationship to define the meta-level always seems to introduce inconsistencies. Atkinson [2] introduces a more sophisticated view of meta-modeling’s role in MDD to address the mentioned shortcoming though introducing two separate orthogonal relationships, namely linguistic and ontological instantiations. Bezivin [3] introduces a more formal view of meta-modeling to understand and clarify the shortcoming through introducing a framework based on conceptual graph. Therefore, an important first step towards a coherent modeling infrastructure, within which model elements could be precisely located, is to explicitly discriminate between instance-of relationships [1, 2, 3].

Category theory is a powerful and abstract theory about structures, and has been used in computer science for reasoning about specifications of programming languages [4, 5, 6]. The intense interest in category theory among researchers in computer science is due in part to the recognition that the different kinds of structures, which exist in this realm, look very much like a category [5]. The main difficulty with applying category theory to a new domain is that the theory is very abstract and formal proofs are required, which means category theory cannot be applied to a domain without a formal mathematical structure [4]. In this paper, the formal representation of MDD infrastructure is briefly proposed. Then, two classes of categories are introduced as a means to capture modeling infrastructure using algebraic formal notions. Finally, by means of constructions defined on the categories, the new paradigm of the infrastructure would be conceived. In summary, we have set the following objective for the research direction reported through this paper:

"To propose a formal framework called the Meta-modeling Infrastructure Formal Framework (MIF) to rearchitecting the traditional linear paradigm to a category-based paradigm in order to provide significant enhancement towards clarifying multiple classification problem"

In this paper, we trace briefly the evolution of the meta-modeling infrastructure paradigms. We propose a formal framework by means of category theory for rearchitecting the infrastructure. We then discuss the usage of the framework to rearchitecting the infrastructure. Finally, we
summarize the consequences of the resulting infrastructure and argue about further works.

2. Meta-modeling infrastructure paradigms

Figure 1 illustrates the traditional four-layer infrastructure that underpins the first generation of MDD technologies—namely the Unified Modeling Language and the Meta-Object Facility [1]. This infrastructure consists of a hierarchy of model levels, each (except the top) being characterized as an instance of the level above. As depicted in figure 1 the bottom layer, M0, is a real-world model instance that corresponds to software execution. The next layer, M1, consists of models that correspond to software definition. In the M2 layer, there are meta-models that essentially define syntax and semantic of the models which reside in M1 layer. In software terms, M2 corresponds to the definition of an implementation language. The top layer, M3, is self-defining, and contains languages that are used for the definition of meta-models, known as meta-metamodels.

This venerable four-layer architecture has the advantage of easily accommodating new modeling standards (for example, the Common Warehouse Metamodel) as MOF instances at the M2 level [1]. Although it has been a successful foundation for first-generation MDD technologies, this traditional infrastructure does not scale up well to handle all technical requirements described earlier [1, 2, 3]. To address these problems, we must adopt a more sophisticated view of meta-modeling’s role in MDD and refine the simple one-size-fits-all view of instantiation. Regarding all instance-of relationships as being of essentially the same form and playing the same role does not scale up well for modeling requirements [1, 2, 3]. Thus, in general, it is necessary for model elements to have both a logical domain classifier, defining its content and a physical classifier defining its structure and presentation. A number of researchers have pointed out the existence of these two different forms of instance-of relationships. Bezivin and Gerbe [3] use the term meta-instance for physical classification and instance-of for logical instance-of relationships and Geisler et al. [7] use the terms inter-level instantiation and intra-level instantiation, respectively.

Although the evolution of the infrastructure to a non-linear paradigm, in which ontological relationships could be defined, covered the traditional paradigm’s shortcomings, in the most recent proposed version in research [1] as depicted in Figure 2, only in three levels the ontological concepts could be defined. Moreover, they have proposed the paradigms in an informal fashion by just visually representing and relating the layers like figure 2 instead of more formal representation. In this paper, we have aim to represent an infrastructure, which are based on formal framework and all the needed relationships could exist between concepts.

Figure 2. Two-dimensional classification infrastructure [1]

3. MIFF: framework for rearchitecting the infrastructure

An algebraic framework is a collection of consistent basic terms, their properties, lemmas and derived propositions based on them. Having appropriate notions, the meta-modeling infrastructure could be formally defined. By formally defining the meta-modeling infrastructure’s components, two classes of categories resides on the two levels of this infrastructure would be specified. Then by means of category theory notions, different relationships between the two classes of categories representing different instance-of relationships could be captured. In this section, the proposed framework for rearchitecting meta-modeling infrastructure, called MIFF, consisting the definition of model, model transformation, identity transformation, modification function, u-structure, homomorphism of u-structures, distinct modification operations, functors, functor composition, and natural transformation are proposed.

A category [6] is composed of objects (models) and morphisms (model transformation) relating them respecting some structural properties. In this work, we would shape the modeling infrastructure by means of categories of models and relationships between them. Therefore, to specify the categories, firstly, the notions of models for representing the objects, modification functions and model transformations for relating the objects, in addition u-structures and homomorphisms of them for describing the structural properties, are needed to formally define. Secondly, by proving the appropriate lemmas based on the notions, the structure of the categories [5] could be formally specified.

**Definition 1 (Model):** A model M1a_jk is a set consisting of model elements which conforms to meta-model M2a_j, which is a set of meta-model elements conforming to meta-metamodel (technical space) M3a,
which is a set of meta-metamodel elements conforming to itself for defining meta-models. Index i denotes the $i^{th}$ meta-metamodel, index j denotes $j^{th}$ meta-model, and index k denotes $k^{th}$ model in the four-layered meta-model architecture. Moreover, M3A denotes a set of meta-models like M2a and M2a' conforming to meta-metamodel M3a, and M2Ai denotes a set of models like M1aijk and M1aijk' conforming to the meta-model M2aij.

**Definition 2 (Modification function):** given a set M2Ai, modification function $\Phi_{a,i}^j$, as depicted in figure 1 with caption T1, can be defined as a sequence of modification operation $\phi_i$ which alter a model element $m_i$ of the M1aijk in M2Ai according to the following statements:

$$\Phi_{a,i}^j (M1aijk) = M1aijk'$$

Where

$$\Phi = \phi_1 \circ \phi_2 \circ \ldots \circ \phi_n$$

$$\phi_i \in \{Insert, Delete, Replace\}$$

$$M1aijk = \{m_1, m_2, \ldots, m_n\}$$

$$\Phi_{a,i}^j (\{\}) = \{\}$$

**Definition 3 (Model transformation):** given two sets M2Ai and M2Aij, the model transformation $f_{M1}^i$ as depicted in figure 1 with caption T2, is a partial function that takes a model M1aijk in M2Ai and produces a model M1aijk' in M2Aij.

**Definition 4 (Identity transformation):** given a model M1aijk, the identity transformation $id_{M1}^i$ makes no change in it according to the following statements:

$$id_{M1}^i (M1aijk) = M1aijk$$

**Definition 5 (U-structure):** A set with a unary operation is a simple structure, which is called a u-structure [5]. If the set is M2Ai and the operation is $\Phi_{a,i}^j$:

$$M2Ai \rightarrow M2Aij, (M2Aij, \Phi_{a,i}^j)$$

is a u-structure, whose underlying set is M2Ai, and whose unary operation is $\Phi_{a,i}^j$.

**Definition 6 (Homomorphism of u-structures):** A homomorphism of u-structures is a function, which preserves the structure [5]. So if (S, u) and (T, v) are u-structures, f: S $\rightarrow$ T is a homomorphism of u-structures if f(u(s)) = v(f(s)) for all s $\in$ S.

**Lemma 1:** The model transformation $f_{M1}^i$ is a homomorphism of two u-structures.

**Proof:**

$$\forall M1aijk \in M2Ai, \Phi_{a,i}^j \exists \Phi_{a,i}^{j'}$$

$$f_{M1}^i (\Phi_{a,i}^j (M1aijk)) = \Phi_{a,i}^{j'} (f_{M1}^i (M1aijk))$$

**Lemma 2:** Any composition of two model transformations ($f_{M1}^i \circ g_{M1}^i$) is a homomorphism of two u-structures.

**Proof:**

$$\forall M1aijk \in M2Ai, \Phi_{a,i}^j \exists \Phi_{a,i}^{j'}$$

$$(f_{M1}^i \circ g_{M1}^i) (\Phi_{a,i}^j (M1aijk)) = \Phi_{a,i}^{j'} (f_{M1}^i (g_{M1}^i (M1aijk))) = \Phi_{a,i}^{j'} (f_{M1}^i (g_{M1}^i (M1aijk))) = \Phi_{a,i}^{j'} (f_{M1}^i \circ g_{M1}^i (M1aijk))$$

**Lemma 3:** Identity transformation $id_{M1}^i$ is a homomorphism of two u-structures.

**Proof:**

$$(id_{M1}^i (M1aijk)) = \Phi_{a,i}^{j'} (id_{M1}^i (M1aijk))$$

**Proposition 1 (M1-category):** a quadruple ([M2Ai, $\Phi_{a,i}^j$], [f_{M1}^i, id_{M1}, $\oplus$]) consisting of a set of u-structures (M2Ai, $\Phi_{a,i}^j$) and a set of homomorphisms f_{M1}^i and an identity transformation id_{M1}^i and a composition operation $\oplus$ is a class of category, which is called “M1-category” for each k=i',...u' subject to the following conditions:

1. composition is associative:

$$f_{M1}^i \circ (g_{M1}^i \oplus h_{M1}^i) = (f_{M1}^i \circ g_{M1}^i) \oplus h_{M1}^i$$

2. identity transformation acts as a identity with respect to the composition:

$$f_{M1}^i \circ id_{M1}^i = id_{M1}^i$$

**Proof:**

Lemma 1, 2, 3 $\Rightarrow ([M2Ai, $\Phi_{a,i}^j$], [f_{M1}^i, id_{M1}, $\oplus$]) is a class of categories with u'-l' members. The quadruple is called “M1-category” which is the class of specified categories reside in the M1 layer of modeling.

To this point, all the notions have been referred to the M1 layer of modeling, but the following definitions are pertained to M2 layer of modeling accordingly.

**Definition 7:** given a set M3A, modification function $\Psi_{a,i}^j$ can be defined as a sequence of modification operation $\psi_i$ which alters a meta-model element mm of the M2aij in M3A according to the following statements:

$$\Psi_{a,i}^j (M2aij) = M2aij$$
Where

\[ \Psi = \psi_1 \bullet \psi_2 \bullet \ldots \bullet \psi_n \]
\[ \psi_i \in \{ \text{Insert, Delete, Replace} \} \]
\[ M2a_{ij} = \{ mm_1, mm_2, \ldots, mm_n \} \]
\[ \Psi_{a_i}(M2a_{ij}) = \psi_1(mm_1) \bullet \psi_2(mm_2) \bullet \ldots \psi_n(mm_n) \]

**Definition 8:** given two sets M3A_i and M3A_i', the model transformation \( \Gamma^{M2}_i \) is a partial function that takes a meta-model M2a_{ij} in M3A_i and produces a meta-model M2a_{ij'} in M3A_i'.

**Definition 9:** given a meta-model M2a_{ij}, the identity transformation \( id_{M2}^i \) makes no change in it according to the following statements:

\[ id_{M2}^i( M2a_{ij}) = M2a_{ij} \]

**Definition 10:** According to the definition 5, a set with a unary operation is a u-structure. If the set is M3A_i and the operation is \( \Psi_{a_i} : M3A_i \rightarrow M3A_i \) (M3A_i, \( \Psi_{a_i} \)) is a u-structure, whose underlying set is M3A_i, and whose unary operation is \( \Psi_{a_i} \).

**Proposition 2 (M2-category):** A quadruple \( \langle \{M3A_i, \Psi_{a_i}\}, \{f_{M2}^i\}, id_k, \circ \rangle \) consisting of a set of u-structures (M3A_i, \( \Psi_{a_i} \)) and set of homomorphisms \( f_{M2}^i \) and identity transformation \( id_k \) and composition operation \( \circ \) is a category which is called “M2-category” for each \( k=a' \ldots u' \), subject to the following conditions:

1. composition is associative:
\[ f_{M2}^i \circ (g_{M2}^i \circ h_{M2}^i) = (f_{M2}^i \circ g_{M2}^i) \circ h_{M2}^i \]

2. identity transformation acts as an identity with respect to the composition:
\[ f_{M2}^i \circ id_k = id_k \circ f_{M2}^i = f_{M2}^i \]

After specification of categories, the needs for capturing the relationship between them are interested. Therefore, the notion of functor in category theory is adopted. In metamodeling infrastructure, functors can be referred as vertical (in comparison with model transformation and modification functions which are horizontal transformation) relationship that relate two levels of abstraction.

**Definition 11 (Functor Type 1):** given M1_i-category and M1_i'-category as two categories, functor \( F_{M1}^{M1} \) as type 1 (between same level of abstraction), which is depicted in figure 1 with caption T3, is a function that assigns to each M1a_{ijk} a \( F_{M1}^{M1}(M1a_{ijk}) \) and to each morphism \( M1a_{ijk} \rightarrow M1a_{ijk'} \) a morphism \( F_{M1}^{M1}(M1a_{ijk}) \rightarrow F_{M1}^{M1}(M1a_{ijk'}) \) in such a way that:

\[ F_{M1}^{M1} : M1_i\text{-category} \rightarrow M1_i'\text{-category} \]

\[ F_{M1}^{M1}(M1a_{ijk}) \rightarrow F_{M1}^{M1}(M1a_{ijk'}) \]

1. \( F_{M1}^{M1} \) preserves composition:
\[ F_{M1}^{M1}(f_{M1}^i \circ g_{M1}^i) = F_{M1}^{M1}(f_{M1}^i) \circ F_{M1}^{M1}(g_{M1}^i) \]

2. \( F_{M1}^{M1} \) preserves identity morphisms:
\[ F_{M1}^{M1}(id_k) = id_{k'} \]

**Definition 12 (Functor Type 2):** given M1_i-category and M2-category as two categories, functor \( F_{M2}^{M2} \) as type 2 (between two level of abstraction), which is depicted in figure 1 with caption T4, is a function that assigns to each M1a_{ijk} a \( F_{M2}^{M2}(M1a_{ijk}) \), and to each morphism \( M1a_{ijk} \rightarrow M1a_{ijk'} \) a morphism \( F_{M2}^{M2}(M1a_{ijk}) \rightarrow F_{M2}^{M2}(M1a_{ijk'}) \) in such a way that:

\[ F_{M2}^{M2} : M1_i\text{-category} \rightarrow M2\text{-category} \]

\[ F_{M2}^{M2}(M1a_{ijk}) \rightarrow F_{M2}^{M2}(M1a_{ijk'}) \]

1. \( F_{M2}^{M2} \) preserves composition:
\[ F_{M2}^{M2}(f_{M2}^i \circ g_{M2}^i) = F_{M2}^{M2}(f_{M2}^i) \circ F_{M2}^{M2}(g_{M2}^i) \]

2. \( F_{M2}^{M2} \) preserves identity morphisms:
\[ F_{M2}^{M2}(id_k) = id_{k'} \]

By introducing several relationships such as functors between categories, there is a need to compare them by means of a formal notion. In category theory, comparison between two relationships is possible through natural transformation.

**Definition 14 (Natural transformation):** given two functors \( F_{M1}^{M1} : M1_i\text{-category} \rightarrow M1_i'\text{-category} \) and \( G_{M1}^{M1} : M1_i\text{-category} \rightarrow M1_i'\text{-category} \), a natural transformation \( \alpha \) from \( F \) to \( G \) is a function that assigns to each object \( M1a_{ijk} \) of \( M1_i\text{-category} \) a morphism \( \alpha_{M1a_{ijk}} : F_{M1}^{M1}(M1a_{ijk}) \rightarrow G_{M1}^{M1}(M1a_{ijk}) \) of \( M1_i'\text{-category} \) such that,
for every morphism $M1a_{ijk} \xrightarrow{f^{M1}_{ijk}} M1a_{ijk'}$ of $M1_r$-category the following square commutes:

$$
\begin{array}{ccc}
F_{\alpha M1_{ij}}(M1a_{ij}) & \xrightarrow{\alpha M1_{ij}} & G_{\alpha M1_{ij}}(M1a_{ij}) \\
F_{\alpha M1_{ij}}(M1a_{ij}) & \xrightarrow{\alpha M1_{ij}} & G_{\alpha M1_{ij}}(M1a_{ij}) \\
F_{\alpha M1_{ij}}(M1a_{ij}) & \xrightarrow{\alpha M1_{ij}} & G_{\alpha M1_{ij}}(M1a_{ij})
\end{array}
$$

Most of the notions introduced in this section have been existed in the category theory literature, but in this paper, we adopt an algebraic definition of the meta-modeling concepts in order to define a categorical paradigm of the infrastructure on which different relationships would be captured. By introducing two classes of categories that one of them resides on the M1 layer of the infrastructure and the other on the M2 layer, also a set of meta-metamodel elements on the M3 layer, all possible relationships could be formally defined with no restriction.

4. Discussion: rearchitecting the infrastructure

Through the proposed algebraic framework, the four-layered meta-model paradigm depicted in figure 1 would be rearchitected to the category-based paradigm depicted in figure 3. As depicted in figure 3, according to proposition 1, each package in M1 layer is the representative of models residing in the M1 layer of the traditional paradigm that have same meta-metamodel. In addition, the package in M2 layer, according to proposition 2, is the representative of all meta-models residing in M2 layer of traditional paradigm that each conforms to a meta-metamodel existing in M3 layer. Moreover, the package in M3 layer is a set of all meta-metamodels residing in M3 layer of traditional paradigm. If there are N technical spaces (meta-metamodels in M3 layer), the number of packages in the infrastructure would be N+2. Each package, which is a category introduced in proposition 1 and 2, has its own transformation and modification functions. By utilizing the new paradigm, all the relationships could be defined formally with no restriction by mathematical notions. It means that based on this infrastructure, we could explicitly relate model elements that are based on same meta-model (by means of a modification function, $\Phi_{a_{ij}}$), different meta-models (by means of a model transformation function, $f_{i M1}$), different meta-metamodels (by means of a functor function type 1, $F_{M1}$) and different levels of abstraction (by means of a functor type 2, $F_{M2}$). Moreover, comparison between two relationships is possible through natural transformation according to the definition 14.

Within this new paradigm of the infrastructure, all kinds of needed relationships between model elements could be captured and serve to precisely locate a model elements with the modeling infrastructure. Therefore, by adopting the new paradigm, the dual classification problem arising from the need to capture both linguistic and ontological classification of model elements, could be formally clarified. This separation has a number of important benefits. In particular, it

- clarifies the distinction between linguistic meta-modeling and ontological meta-modeling (by means of the functions and the functors), and thus opens the way for a sound and optimal balance between the definition of the UML in terms of language features and model library content
- provides a natural approach for problem-oriented meta-modeling and allows an arbitrary number (by means of the categories) of logical meta-levels appropriate for the modeling problem in hand
- provides a sound and natural interpretation (by means of concepts such as the natural transformation) of stereotypes and tagged values in terms of logical meta-modeling
- restores the doctrine (by means of the precise algebraic framework) of logical meta-levels [2] as a natural way of constructing meta-levels according to instance-of relationships in each of the meta-dimensions (linguistic and ontological)

The main contribution of this proposal is that it offers a clearer understanding of how the concepts in the existing meta-modeling architecture can best be evolved, and rearchitected by means of algebraic framework to a more appropriate infrastructure, which enables meta-modeling concepts to be placed on a sound footing.

5. Conclusion and further work

In this paper, we have proposed a formal framework based on which we have introduced the rearchitected paradigm of meta-modeling infrastructure. This categorical-based paradigm has the capabilities to explicitly capture all the relationships between concepts, which we might encounter in modeling endeavor. The related concepts might pertain to a category, to different but same abstraction level categories, or to different abstraction level categories. By means of constructions on the framework in terms of functions, we could precisely locate a model element within the proposed infrastructure.

The applicability and soundness of the proposed framework have not fully proved in this paper. Nevertheless, we believe that the framework we have developed lead the way to a sound basis on which a modeling language could be placed.
6. References

![Figure 1. Layers of modeling (Traditional infrastructure)](image)

![Figure 3. Categories of modeling (Categorical infrastructure)](image)