A Metamodel for the Unified Modeling Language

Andrey Naumenko and Alain Wegmann

Laboratory of Systemic Modeling
Swiss Federal Institute of Technology – Lausanne
EPFL-IC-LAMS, CH-1015 Lausanne, Switzerland
{Andrey.Naumenko,Alain.Wegmann}@epfl.ch

Abstract. Nowadays models, rather than code, become the key artifacts of software development. Consequently, this raises the level of requirements for modeling languages on which modeling practitioners should rely in their work. A minor inconsistency of a modeling language metamodel may cause major problems in the language applications; thus with the model driven systems development the solidness of modeling languages metamodels becomes particularly important. In its current state the UML metamodel leaves a significant area for improvement. We present an alternative metamodel that was inspired by the RM-ODP standard and that solves the problems of UML. RM-ODP was mentioned in UML specifications as a framework that has already influenced UML. Our metamodel was formalized, thus its resulting models can be simulated and checked for consistency. So, our proposed solution with constructive potential towards improvement of the UML metamodel, may have a significant practical impact on the UML specifications.

1 Introduction

Modern practices of software development should successfully manage the progressing complexity of modeling problems. Consequently, this raises the level of requirements for modeling languages on which practitioners, such as software designers and IT system architects, should rely in their everyday modeling work. A minor imperfection of a modeling language metamodel may cause major problems in the language applications. Thus with the model-driven systems development, the solidity of foundations of modeling languages becomes particularly important. These foundations include, for example:

- the overall internal consistency of semantics of a modeling language,
- the coherency and unambiguity in semantics definitions presenting relations between a model constructed using the modeling language from one side and the subject of modeling that is represented by the model from the other side,

J.-M. Jézéquel, H. Hussmann, S. Cook (Eds.): UML 2002, LNCS 2460, pp. 2-17, 2002.

- the theoretical justifications of the semantics relevance (e.g. the necessity and sufficiency of the semantic constructs for a representation of the modeling scope targeted by the language).

In its current state, the UML metamodel leaves a significant area for improvement with regard to the aforementioned criteria. In addition, the UML metamodel is considered to be quite sophisticated by the modelling community. Thus, in order to allow for more successful practical UML applications, as well as for their facilitation, the current state of the UML metamodel should be improved.

In this work, while analyzing the problems of the existing UML metamodel, we present an alternative metamodel that was inspired by the RM-ODP (Reference Model of Open Distributed Processing [1]) ISO/ITU standard. We show how our proposed metamodel successfully resolves some of the existing problems of UML and present literature references supporting our solution. The example of our metamodel that we present in this paper, implements a formalization of RM-ODP conceptual framework. UML specifications mention RM-ODP as a framework that has already influenced UML metamodel architectures ([7] in *Preface: Relationships to Other Models*). This increases the probability for the constructive potential of our metamodel to influence future evolution of UML specifications.

This paper is organized as following. Section 2 will present an analysis of the UML metamodel identifying three of its existing problems. Section 3 will introduce a detailed analysis for each of the three problems. Then Section 4 will define three respective solutions for the identified problems. These solutions will frame the definition of our alternative metamodel. Section 5 will conclude the paper making reference to a concrete example of our metamodel realization and highlighting the most important results of this paper.

2 Problems Identification

When developing any modeling language, the language designer needs to define a scope of the language applications and then to define a set of modeling concepts that would be necessary to represent the defined scope. For the language to be useful in modelers' community practices, the modeling concepts need to have clear, logically structured and consistent semantics. In other words, the better structured the semantics are, and the less internal inconsistencies they have – the more useful the language for the modelers that are interested in representing the identified modeling scope.

Unified Modeling Language (UML) was designed by the Object Management Group (OMG) as "*a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems*" ([7] section 1.1). This identifies the scope of UML applications.

The experience of modeling practices in modern industries shows that UML is found useful by modelers. The amount of modeling projects that use UML, the amount of books written about UML and the number of software tools that support UML are large in relation with the analogous practical achievements of other model-

ing languages. This proves that UML in its current state is more practical then other modeling solutions, although it doesn't mean that there are no problems with the current state of UML.

Consistently with the scenario explained in the first paragraph of this section, the UML specification [7] introduces a set of modeling concepts to represent the identified modeling scope. Section 2 of the specification defines UML semantics for these concepts.

The first problem we can identify is that these UML semantics are considered to be complex and difficult to understand by many modelers. OMG itself in its article "*Introduction to OMG's Unified Modeling Language (UML™)*" [8] confirms this, saying that the UML specification [7] is "*highly technical, terse, and very difficult for beginners to understand*". This situation can be improved by analyzing the current state of UML semantics, understanding the reasons that cause its complexity and by proposing a better organization of semantics for modeling concepts. In particular, we will show that our solution, by introducing a logically precise and internally consistent semantics structure that is based on Russell's theory of types [9], makes a positive difference in relation with the absence of such a structure in the current UML semantics. The explicit presence of such a structure helps to understand how the modeling concepts should be used in practice, whereas its absence creates numerous possibilities for confusions in practical applications of modeling concepts.

While performing the analysis of the current UML semantics we can localize **the second problem**. Specifically that current UML semantics are very ambiguous in presenting relations between models constructed using the language on one side and the subject that is being modeled on the other side. This is an important problem, because even an internally consistent model will not have much of practical sense when its relations with the subject that it is supposed to represent are undefined. This situation with UML is improved in our solution by the introduction of a coherent and unambiguous set of modeling concepts definitions expressing a kind of Tarski's declarative semantics [10] for the mentioned relations between the model and the subject of modeling.

The third problem of the UML semantics, which we will consider in this paper, is the absence of any justifications in the UML specification that would explain why the presented set of UML modeling concepts is necessary and sufficient to represent the UML modeling scope. Without these justifications, the UML theoretical value is significantly diminished, since in this situation the language cannot prove the reasonableness of its ambitions to represent its modeling scope. In our solution, the introduction of the set of modeling concepts is supported by the solid philosophical and natural science foundations providing such kind of justifications.

3 Problems Analysis Based on the Foundations of UML Semantics

As we can see from the previous section, all three identified problems are related to the non-optimal semantics definition. Let us look at foundations of the UML seman-

tics in order to localize the chapters in specifications from where the mentioned problems originate. The UML specification [7] in section 2.4 introduces the semantics Foundation package: "*The Foundation package is the language infrastructure that specifies the static structure of models. The Foundation package is decomposed into the following subpackages: Core, Extension Mechanisms, and Data Types.*" Analyzing the specification further we see for these three packages:

- Core: "The Core package is the most fundamental of the subpackages that compose the UML Foundation package. It defines the basic abstract and concrete metamodel constructs needed for the development of object models." [7], section 2.5.1.
- Extension Mechanisms: "The Extension Mechanisms package is the subpackage that specifies how specific UML model elements are customized and extended with new semantics by using stereotypes, constraints, tag definitions, and tagged values. A coherent set of such extensions, defined for specific purposes, constitutes a UML profile." [7], section 2.6.1.
- Data Types: "The Data Types package is the subpackage that specifies the different data types that are used to define UML. This section has a simpler structure than the other packages, since it is assumed that the semantics of these basic concepts are well known." [7], section 2.7.1.

Thus we can conclude that the three identified problems originate from the Core package of the UML. Consequently it is on this package that we will focus our further consideration.

3.1 Problem 1: Structural Chaos of UML Semantics

Let us now concentrate on the first of the identified problems: the absence of a consistent structural organization of UML metamodel that leads to practical difficulties in understanding semantics for particular modeling concepts, as well as to the difficulties in understanding semantically allowed application contexts for a particular modeling concept.

As it is presented in Figure 2-5 from the Core package specification [7], the most general concept in the UML metamodel is called "Element". It is defined ([7], section 2.5.2.16) as following: "*An element is an atomic constituent of a model. In the metamodel, an Element is the top metaclass in the metaclass hierarchy. It has two subclasses: ModelElement and PresentationElement. Element is an abstract metaclass.*" Thus any atomic constituent of a UML model can be called as UML element.

As it is presented in the diagrams 2-5,6,7,8,9 of the UML specifications [7], all the other modeling concepts are specializations of "Element". This defines a flat structure for the UML metamodel, where any of the concepts can be used as UML elements. And even if the elements obviously belong to different semantic categories (for example, "Operation" and "Class"), there is no explicit categorization defined to help a modeler to understand which concepts should be used in which context.

We may notice an introduction of "abstract" and "concrete" constructs categories in section 2.5.1 of the UML specification: "Abstract constructs are not instantiable and are commonly used to reify key constructs, share structure, and organize the UML

metamodel. Concrete metamodel constructs are instantiable and reflect the modeling constructs used by object modelers (cf. metamodelers). Abstract constructs defined in the Core include ModelElement, GeneralizableElement, and Classifier. Concrete constructs specified in the Core include Class, Attribute, Operation, and Association." However, this categorization becomes quite confusing if it is compared with the actual terms' definitions presented in the UML specifications. For example, "Association" is defined ([7], section 2.5.2.3) relative to "Classifier", which means that "Association" can be considered as both the abstract and the concrete construct. To summarize, the categorization of concepts into the abstract and the concrete constructs does not have a consistent implementation in the current UML specifications and cannot help modelers who would like to understand the possible application context for a particular modeling concept.

An approximate sketch of another possible categorization can be found in section 2.5.2 of UML specifications. The section introduces the figures 2-5,6,7,8,9 as following: "*Figure 2-5 on page 2-13 shows the model elements that form the structural backbone of the metamodel. Figure 2-6 on page 2-14 shows the model elements that define relationships. Figure 2-7 on page 2-15 shows the model elements that define dependencies. Figure 2-8 on page 2-16 shows the various kinds of classifiers. Figure 2-9 on page 2-17 shows auxiliary elements for template parameters, presentation elements, and comments.*"

So a reader could guess that "Backbone", "Relationships", "Dependencies", "Classifiers" and "Auxiliary Elements" are probably different categories of the modeling concepts. Unfortunately these pseudo-categories are neither defined in the relations between each other, nor in some other theoretical or practical application context. In addition, if we check the described figures, we see that the same modeling concepts (e.g. "Classifier" or "Relationship") are present at the same time in several of the diagrams. Thus a potential differentiation between the pseudo-categories is particularly difficult to understand.

We can conclude that the current UML specification of the Core fails to introduce a practically useful categorization of concepts that would define different application contexts for different conceptual categories. Unfortunately this problem cannot be solved by a simple adoption of some categorization for the currently existing UML concepts. This is due to the absence of any explicitly mentioned consistent strategy of concepts introduction by UML. In fact, judging from the specification, for us the strategy for the introduction of particular concepts remains obscure even on an implicit level. Surprisingly some concepts seem to appear without a significant justification whereas other conventional object-oriented terms are omitted.

For example, let us look at definitions of "ModelElement" and "PresentationElement", which are the two subclasses of UML element. We see that "PresentationElement" is defined ([7], section 2.5.2.33) as "*a textual or graphical presentation of one or more model elements.*" Thus essentially a "PresentationElement" is a "ModelElement" presented in a textual or a graphical form. Here we may mention that, in general, a "ModelElement" from inside a model doesn't make sense to anybody or to anything if it is not presented in some form to somebody or to something who perceives the model in this form of presentation. Thus we may affirm that, in general, a "ModelElement", as soon as it is of interest to somebody or to something, is necessar-

ily a "ModelElement" presented in some form. Thus, in fact, "PresentationElement" is a specialization of "ModelElement" where the forms of a possible presentation are known concretely (namely a textual and a graphical form). This specialization is the only value that is added to the semantics of "ModelElement" to obtain the semantics of "PresentationElement". Because of this minor significance of the added value, we may consider "PresentationElement" as not important enough (not essential) to be a separate concept inside the UML metamodel. The elimination of "PresentationElement" accompanied by the addition of the descriptions of possible ways of presentation inside the definition of "ModelElement" would simplify the metamodel without diminishing its value.

3.2 Problem 2: Absence of Declarative Semantics in UML

After having studied the complete UML metamodel, we can note that the UML specifications define explicitly only two concepts whose definitions are made by referring (relating) to the subject (system) that is being modeled. The first concept is "ModelElement", it is defined ([7], section 2.5.2.27) as "*an element that is an abstraction drawn from the system being modeled.*" The second of the two concepts is "Component", it is defined ([7], section 2.5.2.12) as following: "*A component represents a modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces*". All the other concepts that constitute the metamodel are defined as parts of a UML model, **only** in the relations with each other and with the two mentioned concepts. That is, the definitions of all the UML metamodel concepts, with exception of the two mentioned, do not make reference to the subject that is being modeled. This semantics definition is not optimal.

Indeed as we said, only two concepts used in UML models are defined by a reference to the subject being modeled. More than that, the UML metamodel doesn't define why these two concepts and why only these two (and not some other) were designated for this purpose. This means:

a. that this choice of these two concepts does not have a tenable reason defined in the UML specification;
b. that UML specification does not define a tenable relation between a subject that needs to be modeled and its model.

The conclusion 'b.' is particularly important, because it means that for the UML concepts the specification does not define any kind of formal declarative semantics that were introduced by Alfred Tarski [10]. Indeed, Tarski's declarative semantics for concepts used inside models are supposed to introduce mappings between the agreed conceptualizations of a subject that is being modeled and the concepts inside its model. The UML metamodel never presents an agreed conceptualization of the subject of modeling. Thus the specification has no choice but to define modeling concepts exclusively in their interrelations inside the model. In the general case, this approach is not an optimal one for the following two main reasons:

1. The overall complexity of the relations between concepts in the UML metamodel is greater than it would have been if part of the concepts were defined in the rela-

tions with the subject of modeling. Indeed, the quantity of concepts is the same in both cases, but in the latter case some concepts would be defined in a self-sufficient way, whereas in the former, the corresponding concepts for their definitions will need relations to other concepts from the metamodel.

2. Since there is no tenable relation defined between a subject that needs to be modeled and its model and since there is not any agreed conceptualization of the subject, there cannot be a formal proof (such as in the case of Tarski's declarative semantics) that a given modeler's interpretation (that is a model) represents the subject of modeling in a logically consistent way[1]. In other words, in this case several mutually contradicting models can represent the same subject of modeling and all of them may be confirmed as adequate; or, from the other side, one single model may be related with the same degree of adequateness to different mutually incompatible subjects of modeling.

To illustrate the second point let us take Tarski's original example [10] for declarative semantics definition: 'It snows' is true (in the model) if and only if it snows (in the subject of modeling). So if we decide to take the subject of modeling where it snows, without the declarative semantics, then both 'It snows' and 'It does not snow' can be considered true in the model if it snows in the subject of modeling. From the other side, without the declarative semantics the model where "'It snows' is true" may represent equally well both the subject of modeling where it snows and the subject of modeling where it does not snow.

In the case of UML specification, as we said, there are only two concepts that make the direct relation to a subject of modeling: "ModelElement" and "Component". Parts of their definitions can be considered as introducing the declarative semantics. For example, according with [7], sections 2.5.2.27 and 2.5.2.1 we can write for "ModelElement": 'A ModelElement exists' is true in the model if and only if a subject of modeling ("system being modeled") is. And for "Component" according with [7], section 2.5.2.12 we can write: 'A Component exists' is true in the model if and only if there is a modular, deployable, and replaceable part of the subject of modeling ("of system"). But as we see, these definitions are too abstract: they do not give a possibility for a differentiation of modeling concepts, thus the choice of only these two concepts to be defined using the declarative semantics is not practical.

3.3 Problem 3: Absence of Theoretical Justifications for UML Metamodel to Represent the Targeted Modeling Scope

As we said, the third problem of UML semantics is the absence of any justifications in the UML specification that would explain why the presented set of UML modeling concepts is necessary and sufficient to represent the UML modeling scope. This problem can be considered as natural for the current state of UML because, from its

[1] Here we mean the logical consistency in an interpretation of subject of modeling. The internal consistency of a model (the model being a result of the interpretation) is a different subject. The internal consistency of a model can be ensured by the consistency of the UML metamodel, and to ensure the logical consistency of the interpretation, a kind of consistent Tarski's declarative semantics is necessary.

outset, the language was constructed by OMG as a result of the integration of the best existing industrial modeling practices, but these practices were never really linked with the existing scientific theories. Although the "best practices" strategy can be considered as an attempt at practical justification of UML, the theoretical justification was never defined in the language specifications and still needs to be provided.

Thus as we can see, this third problem of the UML metamodel is a theoretical problem, compared to the first two identified problems that are practical. So the third problem is important as soon as UML would pretend to be standardized as a modeling technique by some international standardization committee (such as ISO), which would normally assume solid scientific foundations rather then just results of practical experience to support the language.

4 Solutions for the Identified Problems of the UML Metamodel

4.1 Solution to Problem 1: Categorization of Concepts Based on Russell's Theory of Types

As we explained in Section 3.1, UML doesn't define any explicit logically consistent strategy for the introduction of modeling concepts. To solve the problem of the structural chaos of UML semantics we needed to define such kind of strategy ourselves.

To define the structure of our metamodel, we took the basic conceptual structure of the RM-ODP [1] standard (part 2: Foundations) and reinforced it by means of the strong theoretical foundations of Russell's theory of types [9], as well as by means of the structural principles of Tarski's declarative semantics [10].

As it was proposed in RM-ODP part 2 clause 6 that defines "Basic Interpretation Concepts" conceptual category, we call the subject of modeling (which is the subject that has some modeling interest to a modeler) as "Universe of Discourse". In RM-ODP, "Universe of Discourse" was constituted by entities (defined in [1] 2-6.1 as "*any concrete or abstract thing of interest*") and propositions that can be asserted or denied as being held for entities (defined [1] 2-6.2).

This notion of the "Universe of Discourse" organization is compatible with Russell's theory of types [9] defined by Bertrand Russell in 1908, that introduces individuals and propositions over individuals. Particularly, [9] explains:

"We may define an individual as something destitute of complexity; it is then obviously not a proposition, since propositions are essentially complex. Hence in applying the process of generalization to individuals we run no risk of incurring reflexive fallacies.

Elementary propositions together with such as contain only individuals as apparent variables we will call first-order propositions. We can thus form new propositions in which first-order propositions occur as apparent variables. These we will call second-order propositions; these form the third logical type. Thus, for example, if Epimenides asserts "all first-order propositions affirmed by me are false," he asserts a second-order proposition; he may assert this truly, without asserting truly any first-order proposition, and thus no contradiction arises.

The above process can be continued indefinitely. The (n + 1)th logical type will consist of propositions of order n, which will be such as contain propositions of order n - 1, but of no higher order, as apparent variables. "

Analogously, in our case we have "entity" corresponding to Russell's "*something destitute of complexity*". Indeed, the only intrinsic meaning of an entity in the RM-ODP definition is to be "something" that can be qualified by means of propositions; an entity has no other meaning without the propositions associated with it. Thus, by mapping Russell's "individual" and "proposition" to our "entity" and "proposition", respectively, we can use Russell's suggestion in the context of our universe of discourse. This allows us to differentiate the propositions based on their subject of application:

- if a proposition is applied to an entity it is considered as the first-order proposition;
- if a proposition is applied to a proposition it is considered as the higher-order proposition.

Of course, in an application of these propositions there may be a situation when a higher-order proposition is applied on another higher-order proposition, which in its turn is applied on yet another higher-order proposition and so on, until the overall structure of the higher-order propositions is finally applied on the first-order proposition. Hence for simplification, we will refer to the combination of several higher-order propositions, which is applied on a first-order proposition, as a single higher-order proposition.

So we ordered the entities and propositions that constitute a universe of discourse in agreement with the Russell's theory of types. Now we can look at models that should represent an arbitrary universe of discourse. A model is the place where modeling language constructs should be applied. Thus it is for the model part of our metamodel that we should provide a useful structure of the categorization of concepts, which would explain the different contexts of practical applications for the concepts from different categories.

We suggest organizing the modeling concepts structure in such a way that there would be a straightforward correspondence between the model and the corresponding represented universe of discourse. That is, we suggest constructing a structure of concepts in the model in agreement with Russell's theory of types, which would correspond directly to the universe of discourse organization we presented earlier.

According to our suggestion, within the model we will be able to identify "Model Elements" that will be analogous to the Russell's "*individuals*" defined "*as something destitute of complexity*". Also, under this assumption, in the model we will have some concepts that are analogous to the Russell's "*first-order propositions*" (we will call them "Basic Modeling Concepts"), and some concepts – analogs of the "*higher-order propositions*" (we will call them "Specification Concepts"). With this approach to the construction of a model it would be necessary to qualify "Model Elements" with the aid of "Basic Modeling Concepts", which in their turn could be qualified by means of "Specification Concepts".

Thus we are able to define the correspondence between the conceptual categories from within the model and the entities and propositions that form the universe of discourse that should be modeled. The correspondence was defined as follows:

- *Entities* of the Universe of Discourse are modeled by *Model Elements* in the Model.
- *First-order Propositions* from the Universe of Discourse are modeled by *Basic Modeling Concepts* in the Model.
- *Higher-order Propositions* from the Universe of Discourse are modeled by *Specification Concepts* in the Model.

So, model elements are defined in the model as one to one counterparts to entities from the universe of discourse. Let us consider more closely the two other conceptual categories from within the model. As we showed, in correspondence with Russell's definitions, basic modeling concepts (essentially the first-order propositions) contain model elements as "*apparent variables*"; and specification concepts (the higher-order propositions) contain the basic modeling concepts as "*apparent variables*".

In fact, these two conceptual categories were introduced by RM-ODP specifications ([1] part 2, clauses 8 and 9); up to this point in our presentation we only reinforced logical justifications for this categorization with the support of Russell's theory of types and with explicit definitions of the application contexts for concepts from the two categories. For further explanation of the difference between concepts from the two conceptual categories we will use the principal structure of relations between a universe of discourse from one side and its model from the other side; this structure was defined by Alfred Tarski in 1935 for the introduction of his formal declarative semantics [10].

The basic modeling concepts set, as it aims to model the first-order propositions from the universe of discourse, should contain the concepts expressing the qualities that are considered as primary and intrinsic for the universe of discourse entities. This fundamental nature of the primary qualities belonging to the universe of discourse doesn't allow their modeling representations to be defined exclusively within the model. Hence the only possibility for a definition of the basic modeling concepts is to define them using Tarski's declarative semantics [10]: the semantics that define equivalence of an agreed conceptualization of the universe of discourse to a concrete concept in the model. The set of basic modeling concepts constructed in this way is the necessary, sufficient and limited set representing a limited amount of intrinsic qualities from the universe of discourse.

The set of specification concepts contains all the other concepts that can be found in models. These concepts aim to model the higher-order propositions from the universe of discourse; thus they do not represent the primary qualities of the universe of discourse entities and hence they do not need to have Tarski's declarative semantics for their definitions. So these concepts will be defined only in the relations between themselves and in the relations with the basic modeling concepts, but not in the relations with the universe of discourse. In a general case, the set of specification concepts is not limited because of the same quality of the higher-order propositions set. As new higher-order propositions can be constructed by applying one higher-order proposition

on another, new specification concepts can similarly be constructed by applying one specification concept on another.

So, it becomes clear that there is a significant semantic difference between the two conceptual categories. Basic modeling concepts are defined using Tarski's declarative semantics, but specification concepts are not. This is the consequence of the differences in their design purposes, which explains the clear difference in their corresponding applications within a model.

Additional details on this categorization can be found in [4]. Here let us present a UML diagram explaining the structure of the introduced categorization (see Figure 1).

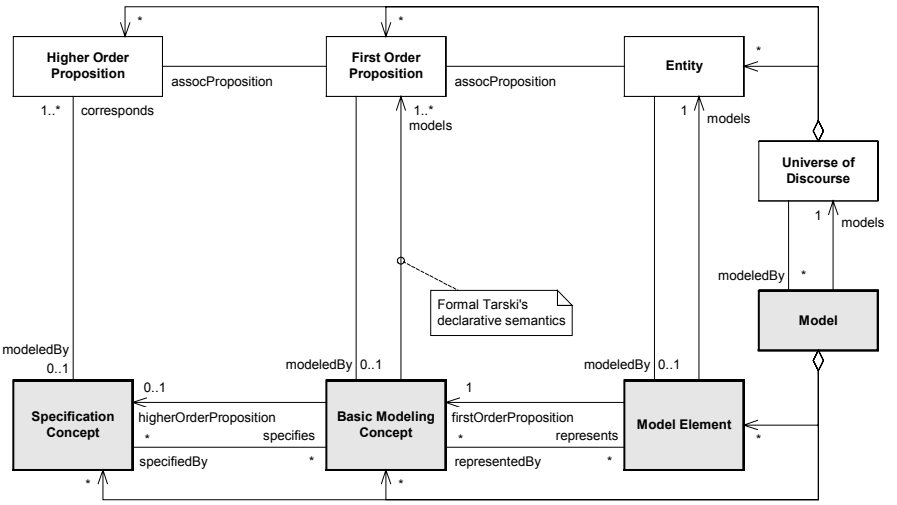

Fig. 1. Categorization of concepts for the proposed metamodel (UML diagram)[2]

4.2 Solution to Problem 2: Tarski's Declarative Semantics Definitions for Basic Modeling Concepts

The complete analysis of definitions for concepts from the basic modeling concepts category that was introduced in the previous section can be found in [3]. Here we will just briefly explain the overall structure of basic modeling concepts, and present this structure in the form of UML diagram.

Figure 2 presents the idea of general organization for the basic modeling concepts category. Essentially the set of concepts is determined by the consideration of the spatiotemporal conceptual continuum and the non-spatiotemporal conceptual continuum. The former represents in the model a space-time from the universe of discourse, and the latter represents in the model the non-spatiotemporal conceptual entities that

[2] In the diagram in Figure 1, in addition to all the explained particularities of the categorization structure, we also showed that a specification concept can specify any of the basic modeling concepts, and a basic modeling concept can be specified by any of the specification concepts. In fact this is true only for the generic specification concepts – the subcategory of specification concepts whose definition is beyond the scope of this paper and can be found in [3,4].

constitute the universe of discourse. In correspondence with their ancestors from the universe of discourse, the former is presented by the Space and the Time dimensions on Figure 2, while the latter by the Model Constitution dimension. Being considered in the same context of a model, the two introduced conceptual continuums necessarily give birth to the third one that is essentially the Information about the Model Constitution within Space-Time. Detailed analysis of this approach can be found in [3].

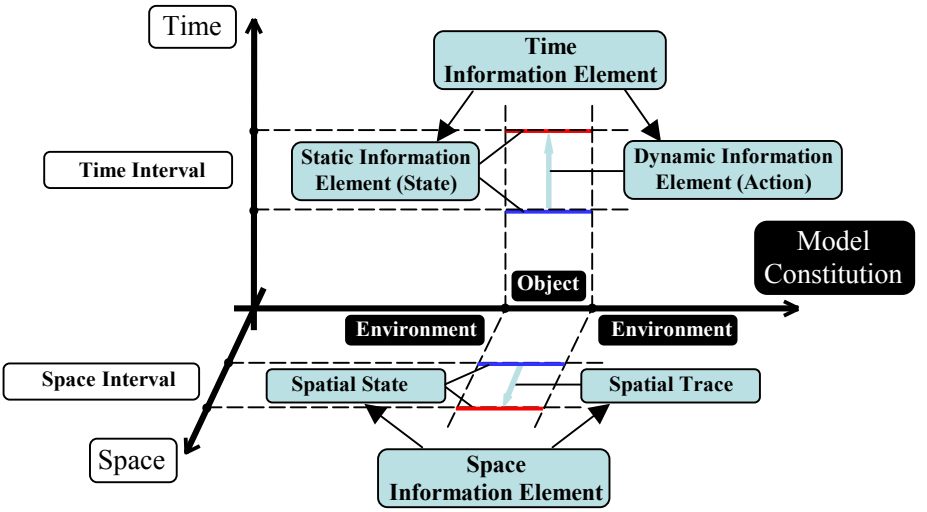

Fig. 2. Three-dimensional framework with the dimensions of "Space Continuum", "Time Continuum" and "Model Constitution Continuum", which allows for the emergent "Information" continuum

By defining limiting points within Space-Time and Model Constitution dimensions we obtain concepts of Space Interval, Time Interval for the Space and the Time and concept of Object with the concept of its Environment for the Model Constitution. Also, with the definition of these limiting points we are able to consider Object and its Environment:

- at a single moment in Time, and thus to define the concept of Static Information Element (State) within the Information continuum;
- at an interval between two moments in Time, and thus to define the concept of Dynamic Information Element (Action) within the Information continuum.

Thus we obtain the structure of basic modeling concepts presented in the UML diagram from Figure 3.

In correspondence with the explanations from the previous section all the basic modeling concepts have formal definitions in the form of Tarski's declarative semantics. We recommend to check [3] for all these concrete definitions.

The definitions of basic modeling concepts, as well as the definitions for all the other concepts proposed in our metamodel, have much in common (and are even

identical in many cases) with the definitions of corresponding concepts given by RM-ODP. We formalized the overall RM-ODP foundations framework ([4], [5], [6]), including the basic modeling concepts part, using Alloy [2] formal description technique. Thus the basic modeling concepts semantics introducing a coherent set of Tarski's declarative semantics for relations between the concepts and the subject that is being modeled (universe of discourse) present a formally justified logical structure.

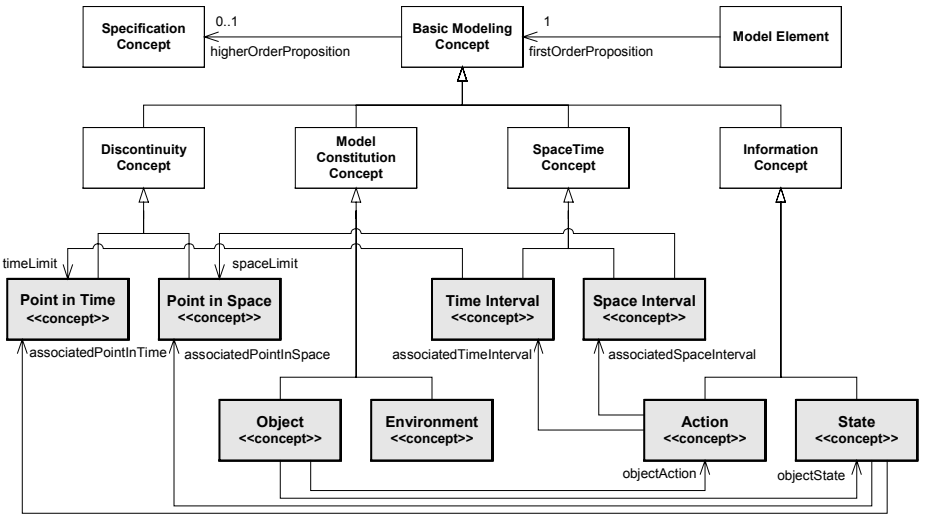

Fig. 3. Basic Modeling Concepts: Conceptual Specialization (UML diagram)

4.3 Solution to Problem 3: Philosophical and Natural Science Foundations of our Proposed Metamodel

As we explained in the analysis of Problem 2, even for the choice of two modeling concepts that were linked in their definitions with the subject of modeling, UML specification does not define any tenable reason. And the set of modeling concepts that are defined using declarative semantics could be the very source of justifications for the ambitions to represent a given modeling scope with the modeling concepts of the language. Indeed, if the declarative semantics concepts cover all the possibilities of the agreed conceptualizations of the modeling scope then, the set of concepts can be considered as sufficient for the modeling purposes. And from the other side, the very set of declarative semantics concepts that would cover all the agreed conceptualizations of the modeling scope can be considered as necessary due to the necessity of the scope representation.

As the previous paragraph shows, the approach to the solution of the third indicated problem of UML metamodel is in the scientific justification of an agreed conceptualization of the modeling scope and in a formally defined unambiguous and logically consistent correspondence of the conceptualization to the modeling concepts that are designated to represent the conceptualization in the model.

The complete theoretical justifications of the universe of discourse conceptualization (that was introduced in the previous section to support the introduction of basic modeling concepts) can be found in [3]. Here we will just mention that:

- the possibility to define limiting points and thus discrete concepts within a conceptual continuum is justified by mereology that is a branch of philosophy studying whole-part relationships;
- the possibility to consider the constitution of models as a conceptual continuum that is independent with regard to the spatiotemporal continuum is an original idea. This idea generalizes fundamental foundations of both classical and relativistic mechanics that study spatiotemporal characteristics of material objects. In our case the scope is generalized to include imaginary conceptual entities. The resulting space-time-constitution framework can be considered as an extension of the traditional Minkowski's space-time framework;
- the vision of defining information as a continuum emerging out from the space-time and the model constitution continuums being considered in the same context is an original idea that however has an analogy found in Taoist philosophy.

The important result was to demonstrate that our conceptualization of the universe of discourse is in agreement with fundamental philosophical and natural science foundations. This demonstration (presented in [3]) allows us to rely on the introduced conceptualization and thus to define the set of Tarski's declarative semantics for the basic modeling concepts of our metamodel as not only having the logical consistency in the interpretation, but also being justified as a generalization of scientific experience. And as we explained in the beginning of this section, the definition of this Tarski's declarative semantics set for the limited modeling scope introduced by the conceptualization provided a straightforward logical proof that the resulting limited set of basic modeling concepts is necessary and sufficient for the modeling scope representation.

5 Conclusions

Our metamodeling solution, described in this paper, is an original proposition that doesn't have direct analogs in any of the modeling frameworks known to us. However, different modeling frameworks can benefit from the presented advantages of our metamodel. This is possible because our metamodel reserves only two things: the structure of its organization and the basic modeling concepts set that consists of six concepts (the concepts presented as gray rectangles in the diagram from Figure 3: Point in Time, Point in Space, Time Interval, Space Interval, Object, Environment, Action and State). At the same time the metamodel allows for an arbitrary construction of the specification concepts set. So, different object-oriented frameworks, as soon as they are destitute of self-contradictions, could fit their terminology in our defined metamodel structure making use of its internal consistency, logical coherency of interpretation, formalized semantics and theoretically justified foundations.

For example, in [3] we show how the RM-ODP conceptual framework (defined in [1]) fits successfully our presented metamodeling structure. In [3] we demonstrate that realizing the appropriate categorization that is implied by the RM-ODP definitions, we can construct the specification concepts structure for our metamodel containing all the defined RM-ODP concepts. This example allowed us to formalize the RM-ODP conceptual structure ([4], [5], [6]), thus now it is possible to verify the RM-ODP models with the aid of computer tools.

Let us summarize the most important results of this paper. In the paper we identified and analyzed three important problems of UML metamodel:

- Absence of an explicit structural organization defined for UML metamodel;
- Absence of a formal declarative semantics in UML metamodel;
- Absence of theoretical justifications for UML metamodel to represent the modeling scope that is targeted by UML.

We solved these problems by defining an alternative metamodel that:

- has an internally consistent structure supported by Russell's theory of types [9];
- defines a kind of Tarski's declarative semantics [15] for the basic modeling concepts, thus it is coherent and unambiguous in the interpretations of subjects of modeling;
- is applied on a concrete example of the RM-ODP conceptual framework that is formalized in a computer-interpretable form [5];
- provides philosophical and natural science foundations to justify that its proposed modeling concepts set is necessary and sufficient to represent its identified modeling scope.

Our metamodel defines concrete improvements for the current state of UML, and it can have a constructive influence on the evolution of UML specifications by providing the language designers with its logical rigor, its formal presentation and its solid theoretical foundations. The concreteness of our solutions and the fact that we implemented them formally on the example of RM-ODP (the framework that was mentioned by UML specifications as influential for UML metamodel architectures) are two strong points that may attract OMG attention to the results of our research.

References

1. ISO, ITU.: ISO/IEC 10746-1, 2, 3, 4 | ITU-T Recommendation X.901, X.902, X.903, X.904. "Open Distributed Processing - Reference Model". 1995-98.
2. Jackson D.: "Alloy: A Lightweight Object Modelling Notation". Technical Report 797, MIT Laboratory for Computer Science, Cambridge, MA, February 2000.
3. Naumenko, A.: "Triune Continuum Paradigm: a paradigm for General System Modeling and its applications for UML and RM-ODP". Ph.D thesis 2581, Swiss Federal Institute of Technology - Lausanne, June 2002. http://stella.epfl.ch/tcp/Naumenko-PhD-Thesis.pdf .

4. Naumenko, A., Wegmann, A.: "A Formal Foundation of the RM-ODP Conceptual Framework" EPFL-DSC technical report No. DSC/2001/040, EPFL, Switzerland, July 2001.
5. Naumenko, A., Wegmann, A.: “RM-ODP part 2: Foundations in Alloy”. EPFL-DSC Technical report No. DSC/2001/041, EPFL, Switzerland, August 2001.
6. Naumenko, A., Wegmann, A., Genilloud, G., Frank, W. F.: "Proposal for a formal foundation of RM-ODP concepts". Proceedings of ICEIS 2001, WOODPECKER`2001, J. Cordeiro, H. Kilov (Eds.), Setúbal, Portugal, July 2001.
7. OMG: Unified Modeling Language Specification. Version 1.4, September 2001.
1. OMG: Introduction to OMG's Unified Modeling Language (UML™). January 2002, http://www.omg.org/gettingstarted/what_is_uml.htm .
9. Russell, B.: “Mathematical logic as based on the theory of types”. American Journal of Mathematics, 30, 1908, pp. 222-262.
10. Tarski, A.: “Logic, Semantics, Meta-mathematics.” Oxford University Press 1956.