

Formalization of the RM-ODP foundations based on the Triune Continuum Paradigm

Andrey Naumenko ^{a,*}, Alain Wegmann ^b

^a *Triune Continuum Enterprise, Switzerland*

^b *EPFL-I&C-LAMS, Switzerland*

Received 14 February 2005; received in revised form 20 October 2005; accepted 21 October 2005

Available online 21 February 2006

Abstract

The subject of research in this paper is RM-ODP, an ISO/ITU-T international standard. We investigate an important challenge that the standard faces, specifically the absence of a satisfactory formalization of its proposed conceptual framework. We report on the definition and implementation of a concrete resolution for this challenge: a single consistent formalization of the RM-ODP conceptual framework. Our solution formalizes denotational and Tarski's declarative semantics for RM-ODP. The formalization is based on the Triune Continuum Paradigm—a fundamental, logically rigorous object-oriented modeling paradigm that can be adapted by different modeling frameworks as a semantic foundation for their conceptual structures.

© 2005 Elsevier B.V. All rights reserved.

Keywords: RM-ODP; Object-oriented frameworks; Semantics; Triune Continuum Paradigm

1. Introduction

Reference Model of Open Distributed Processing (RM-ODP) [15] is a system modeling framework that was standardized by the International Organization for Standardization (ISO) and by the International Telecommunication Union (ITU). Current activities of the ISO/ITU committee that is responsible for RM-ODP standardization include the work towards the promotion of RM-ODP to practice, in order that RM-ODP would have a chance to constructively influence modern modeling practices. From our point of view, one of the approaches to this promotion is a rigorous presentation of the RM-ODP conceptual framework; in particular, a presentation of the framework in a computer-interpretable form.

The importance of such a presentation is emphasized by the fact that formalization of the RM-ODP conceptual framework was officially targeted by the ISO/ITU standardization committee (in fact, the last of the four parts of the RM-ODP standard is dedicated to the formalization). But neither the

standard itself nor the standard-related research had proposed a single consistent formalization of the framework before the definition of the formalization of RM-ODP foundations was discussed in [33], and formulated and implemented in [30].

Naumenko [30] defines the Triune Continuum Paradigm¹ (TCP), a paradigm for general system modeling, and applies the paradigm to interpret the RM-ODP standard conceptual framework. The result of this application allowed for the presentation in [30] of a single consistent formalization of the RM-ODP framework. The formalization is expressed in a computer-interpretable form.

In this paper we will describe the aforementioned formalization, represent a number of its strong points and thus contribute to the defined goal of the promotion of the RM-ODP conceptual framework into modeling practices.

This paper is organized as follows. In Section 2 we explain our motivation behind the formalization work that we present, and define the formalization scope. In Section 3 we introduce the readers to the Triune Continuum Paradigm and its main features, describing the advantages that it brings for the

* Corresponding author. Chemin du Mottey 16 B, 1020 Renens, Switzerland. Tel.: +41 21 624 1452; fax: +41 21 624 1452.

E-mail addresses: naumenko@triunecontinuum.com (A. Naumenko), alain.wegmann@epfl.ch (A. Wegmann).

¹ The readers interested in an introduction to the Triune Continuum Paradigm may refer to [28] and [29], while [30] presents more of the technical details related to the paradigm definition.

formalization of RM-ODP. In Section 4 we present the formalization of RM-ODP conceptual framework. And finally, in Conclusions we summarize the paper emphasizing the strong points of our result.

2. Motivation overview

RM-ODP² defines both a general framework of terms that apply “to any form of modeling activity” (RM-ODP Part 2) and five viewpoints, called “enterprise, information, computational, engineering and technology” viewpoints (RM-ODP Part 3). The general framework defines the fundamental concepts needed to model a system in an object-oriented way. The viewpoints are applications of these concepts for the specific views needed to describe an actual system. We focus our work on the framework definition that can be found in RM-ODP “Part 2: Foundations” and that presents the conceptual kernel of the standard. In particular, it presents several interrelated conceptual categories, each one containing a set of interrelated concepts. In this way RM-ODP “Part 2: Foundations” introduces its holistic ontology for modeling of ODP systems.

Thus, according to the role of RM-ODP Part 2 in the standard, our goal is to present the RM-ODP framework in a formal way and to keep its generic essence in relation to any particular applications. Such formalization applied in a particular modeling process should give easy-to-follow guidelines for building complete and consistent system specifications, and allow for the formal verification of resulting models. And the possibility of verification would significantly facilitate localization of specification faults.

2.1. Analysis of the RM-ODP standard

Let us position our paper with regard to the different parts of RM-ODP. There are four parts in the standard. Part 1 will not be considered for formalization since it contains a motivation overview of ODP and is not normative. Part 2 of the standard introduces ODP concepts and is the core of our formalization work. Part 3 is based on the concepts defined in Part 2. Part 3 defines viewpoints that are necessary to analyze and design ODP systems. Our presented work may be continued in future with Part 3 formalization.

Part 4 of RM-ODP describes the recommendations for approaching the standard formalization with LOTOS [14,25], ACT ONE [10], SDL-92 [17], Z [8,40] and ESTELLE [13] languages. Unfortunately, these are just recommendations or, in the best cases, small unrelated pieces of formalization presented with different formal techniques for the Part 2 concepts. As a matter of fact, even though the goal of the Part 4 is to present “a formalisation of the ODP modeling concepts defined in ITU-T Rec. X.902 | ISO/IEC 10746-2, clauses 8 and

9, and a formalisation of the viewpoint languages of ITU-T Rec. X.903 | ISO/IEC 10746-3” (see [15]), no single consistent formalization representing the clauses 8 and 9 of Part 2 can be found there.

As mentioned in the beginning of Section 2, RM-ODP concepts are introduced within a holistic framework of interrelated conceptual categories. Thus, in a formalization process, the concepts should not be singled out of the overall context of RM-ODP framework. RM-ODP concepts are interrelated; therefore their formalization should result in a single consistent solution and not in a number of unrelated pieces of formal structures corresponding to different isolated concepts.

Some of the ODP practitioners tend to consider that the formalization of the RM-ODP Part 2 is achieved simply because the ITU-T Rec. X.904 | ISO/IEC 10746-4 is standardized. This is unfortunate because the standard fails to provide a true consistent formalization of the concepts of Part 2. This lack is also noticed in [36], which says: “It is definitely the case that the degree of formalism in architecture specifications today varies, and none, including RM-ODP, have achieved a fully mathematical formal specification using an appropriate formal description technique”. Unfortunately the book [36] leaves out the Part 4 in its consideration.

Let us analyze the causes that prevented the standardization of a single consistent formalization of the clauses 8 and 9 of Part 2.

RM-ODP Part 4 contains suggestions for translations of the RM-ODP Part 2 concepts into concepts of the five aforementioned Formal Description Techniques (FDTs). Thus the latter concepts, having their operational semantics in the scope of their particular FDTs, give meaning for every translated RM-ODP concept, showing how an RM-ODP concept can be interpreted in the scope of a concrete FDT in the sense of operational semantics for the FDT concept.³

Such formalizations help people who know the specific FDTs to understand meanings of RM-ODP Part 2 concepts in terms of their familiar formal languages. However, the way proposed in the Part 4 to formalize RM-ODP has a number of limitations:

1. *FDT-dependence*. A proposed meaning of an RM-ODP concept will always be dependent on a particular Formal Description Technique. That is, the meaning will exist in the scope of a single FDT and it will not make sense outside of the operational constraints of this FDT.
2. *Lack of intrinsic RM-ODP generality*. RM-ODP Part 2 does not introduce any operational constraints for interpretation of its concepts. Thus, due to the existence of operational constraints for Formal Description Techniques,

² Each time we mention RM-ODP (or just ODP) in our paper, we refer to [15]. When we mention a part number (can be with name) and/or a clause number (or name) of RM-ODP, we refer to the corresponding part and/or clause in [15].

³ Note that expressive power of formal languages used in writing of RM-ODP Part 4 is limited and insufficient for the presentation of FDT-dependent operational semantics for RM-ODP Part 2 concepts. Part 4 explicitly refers to Part 2 concepts that were impossible to represent using the chosen FDTs. The choice of formal languages in Part 4 was defined within the scope of FDTs standardized by ISO at the time of RM-ODP standardization work.

the concepts of any particular FDT will always be less general than RM-ODP concepts. So within any of the explained FDT-dependent interpretations the intrinsic meaning of RM-ODP concepts will always be reduced.

3. *Absence of a single consistent formalization.* To perform the explained FDT-dependent interpretations, concrete RM-ODP Part 2 concepts are singled out of the complete RM-ODP Foundations framework. This produces a number of unrelated pieces of formalization presented with different formal techniques. A single consistent formalization of the complete conceptual framework from clauses 8 and 9 of Part 2 of RM-ODP is impossible to achieve with this formalization approach.

A detailed explanation of the formalization approach, which currently exists in RM-ODP Part 4, may be found in [38]. The approach never considers explicitly the relations that exist between different conceptual categories (such as between “Basic modelling concepts” (RM-ODP Part 2 clause 8) and “Specification concepts” (RM-ODP Part 2 clause 9)). This prevents a definition of the denotational semantics for the RM-ODP framework. In addition, existing formalization approach considers the concepts from RM-ODP Part 2 clause 8 and Part 2 clause 9 without considering “Basic interpretation concepts” (RM-ODP Part 2 clause 6); that is, it considers the concepts used within models without a relation to the concepts representing the universe of discourse being modeled. This means that Tarski’s declarative semantics⁴ are ignored in the current formalization approach.

In summary, the existing formalization approach favors *operational* semantics, and ignores *denotational* semantics and *Tarski’s declarative* semantics [32]. The approach abstracts from “Categorization of concepts” (RM-ODP Part 2 clause 5), which makes the single holistic framework, informally presented in RM-ODP Part 2, impossible to represent formally in Part 4.

In this paper we present an alternative approach to the formalization of RM-ODP “Part 2: Foundations”. Our approach is based on the Triune Continuum Paradigm [28–30]. It allows for formal definitions of denotational semantics and of Tarski’s declarative semantics for the conceptual structure of RM-ODP. Unlike operational semantics, these two kinds of semantics are dedicated to present relations of the concepts of a given modeling language independently of any operational context (see [32]), just expressing logical interrelations that exist between the concepts. So, unlike operational semantics (that depend on the operational constraints of particular FDTs used for their expression), denotational semantics and Tarski’s declarative semantics remain invariant regardless of any Formal Description Technique that can be applied to express them.

⁴ To see the relations between operational, denotational and Tarski’s declarative semantics the readers can refer to [32]. This paper also explains the importance of Tarski’s declarative semantics in the design of modeling languages.

Thus our approach is devoid of the analyzed limitations of the currently existing version of RM-ODP Part 4. Based on the Triune Continuum Paradigm, we are able to present a single consistent formalization of the RM-ODP “Part 2: Foundations”; the formalization which, being semantically independent from any Formal Description Technique, defines a formal meta-model of the RM-ODP standard conceptual framework.

The meta-model of RM-ODP Part 2 allows ODP models to be verified and checked for consistency. It stays on the same conceptual level as the UML meta-model [35] thus providing potential for UML and other modeling languages to be influenced by RM-ODP.⁵ Hence, we preserve the generic essence of the ODP framework with regard to the potential applications.

Positioning itself as the standard meta-model, our solution presents a significant advantage in comparison with those described in the Part 4 of RM-ODP. Specifically, we provide a possibility to formalize definitions and mutual relations not only for the modeling concepts (basic modelling concepts and specification concepts), as it is in Part 4, but for all the other relevant ODP concept categories. Hence, we benefit from the completeness of the scope definition within the RM-ODP standard and are able to formally express Tarski’s declarative semantics, i.e. to show clear relations between the universe of discourse being modeled and the model of it (including the basic modelling part and the specification part). We heavily emphasize the importance of RM-ODP Part 2 clause 5 (categorization of concepts) that is often ignored by ODP practitioners—perhaps due to its relatively implicit definition in the standard.

2.2. Analysis of previous research on RM-ODP formalization

A formal view on RM-ODP specifications insures their consistency within a frame of a particular project, and the ODP research community has produced several interesting results that are important for understanding the consistency and for implementing consistent specifications. Particularly, Bowman et al. [7] presented a nice discussion on the requirements that the RM-ODP framework imposes on different Formal Description Techniques (FDTs) for the formal interpretation of ODP systems specifications (note that ODP systems specifications are concrete applications of the RM-ODP conceptual framework for systems modeling—they are not the framework itself). Bowman et al. [7] considered a set of general ODP requirements and elaborated on the requirements for specific ODP viewpoints. Further research [5,6,24] defined a general

⁵ Note that as it is presented in [31], the existing official definition of the UML framework [35] lacks both internal consistency and a logical order in its organization. A number of efforts has been performed to improve this situation with the aid of constructive influence of RM-ODP. In particular, some of the UML profiles, such as Relationship Profile [34] have already been substantially influenced by RM-ODP. And it is also interesting to mention here the ongoing work for the new ISO/ITU standard “UML for ODP” [16] that is performed by ISO/JTC1-SC7 Working Group 19 (“Open Distributed Processing and Modelling Languages”) and that aims to describe a way to use UML for specifying ODP systems.

meaning of the specification consistency in the context of the RM-ODP viewpoints. Based on these papers, relating formalisms used for different viewpoints became a standard approach for ODP formalization work. This kind of problem positioning was previously considered independently from the context of the RM-ODP standard; for example, the analogous question of “multiperspective specifications” consistency was discussed in [12]. In the case of RM-ODP, the viewpoints are well-defined in the standard—this allowed for the publication of some successful case studies. For instance, studying interrelations of the viewpoints, Bernardeschi et al. [2] present mappings between the information viewpoint and the computational viewpoint languages; Boiten et al. [4] relate the computational viewpoint with the engineering viewpoint. At the same time another research thread concentrates only on formalizations for specific viewpoints. Durán and Vallecillo [9] and Steen and Derrick [41] propose approaches for the enterprise specifications and Najm and Stefani [26,27] and Sinnott and Turner [39] formalize the ODP computational model. Examples of approaches for computational, engineering and technology viewpoints can be found in [3]. Thus, historically the emphasis in ODP formalization research was put on the formalization of viewpoint languages and even more on the formalization of the specifications performed with the viewpoint languages.

The idea behind our work differs from those mentioned in the previous paragraph. Its originality is to consider formalization of the RM-ODP foundations presented in Part 2 of the standard, rather than formalization of ODP viewpoints introduced in Part 3 and related ODP systems specifications. This choice is justified by the standard, because the RM-ODP conceptual framework from Part 2 is defined to support ODP viewpoints. It presents a general vision on modeling, the vision that should further be applied in the context of a particular ODP viewpoint. Particularly, Sinnott and Turner [38], reporting on experiences out of the standard development, clarify: “The relationship between Part 2 and Part 3 of the RM-ODP may be seen as specialization. That is Part 2 gives a basic interpretation of a given concept and Part 3 gives a more specialized version.” The importance of a formal view on Part 2 of RM-ODP was noted not only by the standard itself but also in [20] and [21]. Specifically, Johnson and Kilov [20] are saying: “RM-ODP emphasizes common fundamental concepts encountered in any open distributed system, including distribution-independent concepts! It is used for descriptions of any system, not just software; and it includes both viewpoint-specific concepts, as well as—perhaps, more importantly—concepts common to all viewpoints—enterprise, information, computational, and so on.” Thus by formalizing Part 2 of RM-ODP, we provide a general, consistent and complete framework for modeling, suitable for further applications in the less general but more profound and precise contexts of the viewpoints.

2.3. RM-ODP Part 2: scope for formalization

As we have explained the motivation for our work, let us now describe the sections of RM-ODP Part 2 that were

formalized in [30]. In Part 2 there are 15 clauses. Clauses 1–4 are auxiliary to the rest of Part 2 and were not considered for the formalization. These clauses serve to introduce “RM-ODP Part 2: Foundations” in the overall context of the standard. Specifically, they:

- define the scope of “RM-ODP Part 2: Foundations” (Part 2 clause 1);
- refer to the general rules of ISO and ITU standardization (Part 2 clause 2);
- list “Background definitions” for the general terms used in the standard references (Part 2 clause 3);
- define the abbreviations used in the standard (Part 2 clause 4).

Further, in the scope definition (Part 2 clause 1) it is mentioned that: “This ITU-T Recommendation | Part of ISO/IEC 10746 covers the concepts which are needed to perform the modelling of ODP systems (clauses 5 to 14), and the principles of conformance to ODP systems (clause 15).” The conformance principles covered in clause 15 describe how an implementation of a specification performed with RM-ODP modeling framework should conform to the specification. So, clause 15 does not introduce concepts that are needed to perform the modeling of ODP systems. Therefore clause 15 was not considered for the formalization.

Clauses 10–14 represent the so-called “structuring concepts”. As they are defined in RM-ODP Part 1 clause 6.2.1: “structuring concepts - building on the basic modelling concepts and the specification concepts to address recurrent structures in distributed systems, and cover such concerns as policy, naming, behaviour, dependability and communication.” Essentially, these are the concepts defined by means of “Basic modelling concepts” and “Specification concepts” (Part 2 clauses 8 and 9). Consequently, they could be formalized as soon as the concepts from RM-ODP Part 2 clauses 8 and 9 are formalized. Leaving this subordinate formalization of clauses 10–14 outside of the scope of this paper, we emphasize, however, the generic nature of “structuring concepts” and their importance for system modeling applications.

Clause 7, “Basic linguistic concepts”, introduces two concepts: “Term” and “Sentence”. These are linguistic constructs that should be used for expressions in any language that could be employed for the description of ODP modeling. So, these are the concepts that are defined on the meta-level for the RM-ODP meta-model (meta-meta-level concepts) and they were not considered for the formalization.

All the remaining clauses of Part 2 (namely the clauses 5, 6, 8 and 9) were formalized in [30]. They represent the kernel of RM-ODP framework, including:

- RM-ODP Part 2 clause 5: “Categorization of concepts”;
- RM-ODP Part 2 clause 6: “Basic interpretation concepts”;
- RM-ODP Part 2 clause 8: “Basic modelling concepts”;
- RM-ODP Part 2 clause 9: “Specification concepts”.

3. Foundations of the formalization

The definitions of the conceptual framework provided in the RM-ODP standard are not self-sufficient; the standard does not make an explicit presentation of the assumptions that must be taken to interpret the definitions. This allows for the existence of different alternative interpretations of the conceptual framework, which obviously impedes a consensus about a single formalization of the framework. Thus to perform the formalization work on which we report in this paper it was necessary first, to introduce the interpretation constraints that were necessary to interpret and formalize the RM-ODP framework, and second, to explicitly present these constraints to people so that everyone would have a definitive idea about the assumptions behind the formalization.

We chose the Triune Continuum Paradigm (TCP) [28–30] as a paradigm to support our interpretation of the RM-ODP framework. This paradigm not only provided the necessary interpretation constraints, but also reinforced the standard conceptual framework with a number of solid theoretical foundations. In particular, the paradigm suggests three fundamental theoretical foundations to be used for the definition of a system modeling framework. The three theoretical foundations and their implications are:

1. Tarski's Theory of Truth, allowing for the definition of Tarski's declarative semantics for concepts in the framework;
2. Russell's Theory of Types, allowing for the categorization of concepts in the framework performed according to the theory;
3. Theory of Triune Continuum, allowing for the introduction of the basic modeling concepts in the framework according to the notion of Triune Continuum.

As defined in [30]:

- the first foundation assures unambiguity, adequacy and coherency in modeling interpretations, and makes explicit the representation capabilities of RM-ODP modeling framework;
- the second supports internal consistency of the framework and of the framework's applications;
- and the third assures the necessity and sufficiency of the introduced concepts for the general system modeling scope representation.

Let us make a brief presentation of these three foundations.

3.1. Tarski's Theory of Truth

Tarski's Theory of Truth, proposed by Alfred Tarski in 1935 [42], allows for the definition of Tarski's declarative semantics. Triune Continuum Paradigm uses the semantics in the context of general system modeling [32] to define formal relations between a subject that needs to be modeled and possible

models of this subject. The relation between a subject of modeling interest and its model is made by the modeler, who has this modeling interest with regard to the subject and who produces the model. Tarski's theory suggests to modelers to adopt an unambiguous way of defining this relation: an explicit one-to-one mapping between the modeler's perceived conceptualization of the subject of modeling and the representation of this conceptualization in the model. This is the way to define Tarski's declarative semantics for concepts used in the representation.

For example, in RM-ODP Part 2, clause 8.3 "Action" is defined as "Something which happens". "Something which happens" is the perceived conceptualization, agreed by ODP modelers (as we assume that the modelers agree on what "Something which happens" means). "Action" is the representation of this conceptualization in the model.

3.1.1. Adequacy of modeling interpretations

According to Tarski's theory it was shown [32] that:

- If the declarative semantics *are adopted* by modelers in a modeling community then:
 - ⇒ if different modelers *agree* on the conceptualization of a subject of modeling then:
 - the modelers can formally compare their respective models representing this subject.
 - ⇒ if different modelers *do not agree* on the conceptualization of a subject of modeling then:
 - the modelers cannot compare their models in a logically rigorous way.
- If the declarative semantics *are not adopted* by modelers in a modeling community then:
 - ⇒ the modelers cannot compare their models in a logically rigorous way.

Thus, as assured by Tarski's theory, because of the unambiguity of Tarski's declarative semantics, modelers have a possibility to argue formally about their models in their community. And this possibility exists only in the case when the modelers both adopt the declarative semantics and agree on the conceptualization of subjects of their modeling interests. This possibility of formal reasoning about modeling interpretations within a community of modelers favors adequate interpretations and discourages inadequate interpretations. For example, if Tarski's semantics formally define that:

"Action" in the model can exist if and only if there is "Something which happens" in the universe of modeling discourse,

then any interpretation (of "Something which happens") that is not "Action" becomes formally impossible in the model because of its inadequacy with regard to the semantics.

So unambiguity of Tarski's declarative semantics favors adequacy and discourages inadequacy of interpretations of subjects of modeling interest within applications of modeling frameworks that employ Tarski's semantics.

3.1.2. Coherency of modeling interpretations

Another important property that Tarski's semantics adds to modeling frameworks is coherency of modeling representations of subjects of modeling interest. This promotes coherent interpretations of the universe of modeling discourse in a given modeling project (and respectively hinders incoherent interpretations). Indeed, if Tarski's semantics are defined, then conceptualizations of the universe of discourse are necessary and sufficient for their respective modeling terms to be true within a particular modeling representation. Thus, should a conceptualization of the universe of discourse be incoherently interpreted in several individual cases within the representation (that is, should the conceptualization be mapped to several conflicting assertions in the representation), then the resulting incoherency formally described and immediately highlighted.

This property does not guarantee the adequacy of interpretations that we discussed in Section 3.1.1: a result of a coherent interpretation can be an inadequate representation.

3.1.3. Explicit representation capabilities for a modeling framework

By their definition Tarski's declarative semantics introduce explicit conceptualizations of the universe of discourse to be presented. This allows the users of a particular modeling framework to understand its representation capabilities. Such an understanding is very important because it removes the temptation to apply the framework in cases for which it is not designed. That is, such an understanding prevents the kind of failures that occur in modeling projects when modelers try to resolve modeling problems using an inadequate conceptual toolkit.

3.1.4. Compatibility of RM-ODP framework with Tarski's declarative semantics

In Sections 3.1.1–3.1.3 we have reviewed three valuable properties that Tarski's Theory of Truth guarantees for modeling frameworks that use Tarski's declarative semantics. The principles of Tarski's Theory of Truth are an integral part of the Triune Continuum Paradigm, thus in a concrete application of the paradigm (for example for the conceptual framework of RM-ODP) the paradigm:

- makes explicit the limits of representation capabilities for the chosen modeling framework,
- assures coherency and adequacy in different possible interpretations of subjects of modeling interest.

RM-ODP modeling framework [15] does not formally define Tarski's declarative semantics for its terms. However the framework is compatible with the principles of Tarski's Theory of Truth and allows for definitions of these semantics. Indeed, RM-ODP:

- introduces its universe of discourse,
- defines conceptualizations of the universe of discourse,
- in the definitions of its modeling terms refers to the defined conceptualizations.

The only missing part (that is required for the definition of Tarski's semantics) is to declare the truth of a modeling term in the model to be necessarily and sufficiently determined by the relevance of the conceptualization (respective to the term) in the universe of discourse. In other words, the relations between modeling terms and their corresponding conceptualizations of the universe of discourse should be defined in Tarskian "if and only if" form (see example for the term "Action" in Section 3.1.1).

Current RM-ODP definitions (which [15] introduces for RM-ODP modeling terms) do not contain Tarskian necessity and sufficiency. On the other hand, current definitions do not contradict with possible extensions that would define Tarski's semantics for their terms. And moreover, as we just mentioned, RM-ODP introduced all the necessary machinery for the definitions of Tarski's semantics. Thus for our formalization of RM-ODP framework it was possible to apply the Triune Continuum Paradigm as if the definitions of Tarski's semantics existed in the standard (regardless their actual absence). This application, of course, introduced some additional constraints for interpretation of RM-ODP framework. These additional constraints are not documented in [15], but they do not contradict to the current RM-ODP definitions, and, reinforcing RM-ODP with the strong logical foundations of Tarski's Theory of Truth, they provide the three valuable properties that we have discussed in Sections 3.1.1–3.1.3.

As explained in [30], it was particularly important to define Tarski's declarative semantics for Basic modeling concepts of RM-ODP. This was done with the aid of the Theory of Triune Continuum (see Section 3.3 further in this paper; more details can be found in [28] and in [30]). While Specification concepts of RM-ODP did not require Tarski's semantics (denotational semantics were sufficient here).

The readers may refer to [32] to discover more details about the usefulness of Tarski's Theory of Truth for modeling frameworks.

3.2. Russell's Theory of Types

Russell's Theory of Types [37] was defined by Bertrand Russell in 1908. This theory is used in the Triune Continuum Paradigm to ensure internal consistency in the metamodeling structure proposed by the paradigm.

The metamodeling structure is one of the important features proposed by the paradigm; its importance is explained by the fact that this structure should shape the metamodels of the concrete system modeling frameworks that would adopt the Triune Continuum Paradigm for their concepts (e.g. the conceptual framework of RM-ODP). Because of this importance the metamodeling structure needs a solid theoretical support that is assured by Russell's theory of types. Two important features of the metamodel proposed by the paradigm with the aid of Russell's theory are:

- the metamodeling structure is rigorous — to thoroughly define precise application contexts for the different concepts that could potentially make use of this structure;

- at the same time, the structure is flexible — to make it possible to be adopted by the diverse range of already existing system modeling frameworks.

Russell’s theory of types defines a structure of propositions that can be used in the logically rigorous constructions in a language to avoid the famous Russell’s paradox [11]. Meta-modeling structure of the Triune Continuum Paradigm was defined in [30] adhering to the structure of propositions introduced by Russell. In particular for the construction of logically rigorous statements in a language Russell introduced (see [37]):

- individuals: “We may define an individual as something destitute of complexity; it is then obviously not a proposition, since propositions are essentially complex.”
- first-order propositions: “Elementary propositions together with such as contain only individuals as apparent variables we will call first-order propositions.”
- higher-order propositions (the second-order propositions, the third-order propositions, etc.): “We can thus form new propositions in which first-order propositions occur as apparent variables. These we will call second-order propositions.”

For the metamodeling organization of concepts used in a model, in correspondence with Russell’s theory, Naumenko [30] defines:

- Model Elements (MEs)—direct analogs of Russell’s individuals. Model Element is the most general term referring to any element of the model. As are Russell’s individuals, MEs are “destitute of complexity”. This means that Model Elements, considered without the propositions associated to them, do not exhibit any particular information.
- Basic Modeling Concepts (BMCs)—direct analogs of Russell’s first-order propositions. In a model, BMCs characterize Model Elements in the same way as the first-order propositions characterize the individuals in Russell’s theory of types.
- Specification Concepts (SCs)—direct analogs of Russell’s higher-order propositions. In a model, SCs characterize BMCs in the same way as the higher-order propositions characterize the first-order propositions in Russell’s theory of types.

Thus Russell’s theory assures the aforementioned necessary rigor of the paradigm’s metamodeling structure. Also this structure is potentially flexible, because no particular constraint is given for the definitions of concrete BMCs and SCs.

The names of “Basic Modeling Concepts” and of “Specification Concepts” in the Triune Continuum Paradigm are analogous to the names of two conceptual categories introduced in Foundations of RM-ODP but, as it is explained in [30] and discussed in Section 4 of this paper, in some cases the RM-ODP concepts partitioning does not correspond to the one proposed by the paradigm.

TCP metamodeling structure is not the only solution that may exist for the aforementioned requirements (rigor and flexibility). Other solutions may also be considered. For example, Yaglom [45] talks about generic approach: “model . . . described in accordance with Aristotle’s general scheme of construction of “deductive sciences,” i.e. by a list of fundamental (undefineable) objects and relations and a collection of axioms characterizing those objects and relations.” And one may try to explore other approaches. However, TCP metamodeling structure provides a concrete advantage.

Thanks to the Russell’s theory, TCP supports internal consistency of relations between concepts within specifications by defining the way in which the relations should be constructed. Russell’s first-order propositions that can be applied on individuals are differentiated from the higher-order propositions that can be applied on top of the other propositions but cannot be applied on individuals. Analogously, in the Triune Continuum Paradigm BMCs can be applied on MEs. And SCs are differentiated from BMCs: SCs cannot be applied on MEs, they can only be applied on top of BMCs and on top of other SCs. And for the construction of conceptual structures the higher-order SCs can be applied only on the SCs or BMCs that are one order below, but cannot be applied on the SCs or BMCs that are more than one order below them.

By constructing the hierarchies of concepts within specifications in accordance with Russell’s theory, potential internal contradictions will be detected and disallowed, and thus internal consistency of specifications will be preserved.

Let us consider an example of Russell’s paradox. Specification (S1) of “a hairdresser who cuts the hair of all the people who do not cut their own hair and only of those people” introduces internal contradiction. Let us consider the question: “Can the hairdresser cut her own hair?” in the context of specification S1. On one hand, if the hairdresser can cut her own hair, then she is not able to cut the hair *only* of those that do not cut their own hair (C1). On the other hand, if the hairdresser cannot cut her own hair, then she is not able to cut the hair of *all* the people who do not cut their own hair (C2).

Here we have a case of differentiation of people into two categories: those who cut their hair and those who do not cut their hair. This differentiation corresponds to two disjoint sets and, since they are disjoint, for any member of the sets it is impossible to belong to two of them at the same time. In S1 predicate “cuts the hair of all the people who do not cut their own hair” applied to the hairdresser puts her to the set of those who cut their hair (because of C2). And at the same time predicate “cuts the hair only of those people who do not cut their own hair” applied to the hairdresser puts her to the set of those who do not cut their hair (because of C1). Thus the two predicates applied to the hairdresser contradict with each other.

Russell’s theory makes it possible to resolve this contradiction. And TCP, making use of Russell’s theory, disallows

specification S1 within an application of any modeling framework that would use TCP. An attempt to construct S1 will manifest the conflict of types within the hierarchy of SCs as following:

1. Introduction of a model element:
ME1;
2. Assignment of Basic Modeling Concept BMC1 = “object” to ME1:
BMC1(ME1): “ME1 is an object”;
3. Assignment of Specification Concept SC1 = “type: hair-dresser” to BMC1(ME1):
SC1(BMC1(ME1)): “ME1 is an object, type: hairdresser”;
4. Assignment of Specification Concept SC2 = “subtype: cuts the hair of all the people who do not cut their own hair” to SC1(BMC1(ME1)):
SC2(SC1(BMC1(ME1))): “ME1 is an object, type: hair-dresser, subtype: cuts the hair of all the people who do not cut their own hair”;
5. Assignment of Specification Concept SC3 = “subtype: cuts the hair only of the people who do not cut their own hair” to SC2(SC1(BMC1(ME1))):
SC3(SC2(SC1(BMC1(ME1)))): impossible, SC3 is in conflict with SC2 for any SC2(SC1(BMC1(ME1))).

This example shows how the use of Russell’s theory eliminates internal contradictions in applications of modeling frameworks that use TCP. On the other hand, ignoring Russell’s structure of propositions favors appearance of such contradictions. For instance if we ignore Russell’s theory in the example above, then application scope of SC3 remains unrestricted, and it is allowed to apply SC3 directly on SC1(BMC1(ME1)). Thus any check of SC3 application will show that SC3 is applicable (indeed SC3(SC1(BMC1(ME1))) is a valid construction, thus SC3 is applicable). And so, based on such a check, one may mistakenly think that SC3 is applicable everywhere, and that SC3(SC2(SC1(BMC1(ME1)))) is a valid construction.

3.3. Theory of Triune Continuum

To realize potential flexibility of the metamodeling structure that we discussed in the previous section, the paradigm defines a minimal set of BMCs; the set which is necessary and sufficient for a complete representation of the general system modeling scope on the most abstract level. Such a solution allows different existing system modeling frameworks to place their modeling concepts as specializations of these BMCs (that is, as the concrete SCs applied on the BMCs), and hence to adopt the paradigm with its solid theoretical foundations.

Naumenko [30] proposes the Theory of Triune Continuum as a solution to define and justify this minimal necessary and sufficient set of Basic Modeling Concepts. The solution provides to modelers a special observer-relational frame of reference that is defined as a philosophically supported

generalization of fundamental frameworks of natural science. In particular:

- in classical (Newtonian) mechanics, observer-relational reference frames exhibit the relational nature in space, whereas time and material objects remain invariant for different observers;
- in relativistic mechanics observer-relational reference frames exhibit the relational nature in space and in time, whereas material objects remain invariant for different observers;
- in the Theory of Triune Continuum, an observer-relational reference frame exhibits the relational nature in space, in time and in the constitution of models that represents different subjects of modeling (including material objects) in the models. So, representations of material objects are observer-relational here.

The defined frame of reference is based on the original notion of Triune Continuum, which owes its name to the triune essence of three continuums. These three continuums, as shown in [30], are necessary and sufficient to represent the general system modeling scope. The three defined continuums are:

1. *spatiotemporal continuum* (introducing space–time in models and defining subjective space–time metrics to be used in the models);
2. *constitution (non-spatiotemporal) continuum* (introducing constitution of models, and defining subjective constitutional metrics to be used in the models, e.g. objects defined in relation with their environments);
These two continuums are introduced as complements to each other within the universal general system modeling scope. In other words, everything in the scope that is not space–time is constitution, and everything in the scope that is not constitution is space–time.
3. *information continuum* (emerging from the mutual relations of the first two continuums and containing information about these relations, e.g. information about objects and their environments being related to the spatiotemporal intervals or to the points in space–time).

Thus the three continuums are triune: none of them exist without the others; either the three exist altogether, or they do not exist at all. Indeed, as soon as the first (spatiotemporal) continuum is introduced, everything in the universal scope that does not belong to the first continuum immediately shapes the second (constitution) continuum; and the third (information) continuum immediately emerges as the information about the mutual relations of the first two continuums (e.g. as spatiotemporal information about the constitution).

The readers can find more of the general information about the Theory of Triune Continuum in [28] and in [30]. Let us present here two of examples demonstrating how applications of this theory can be relevant in relation to the RM-ODP conceptual framework.

3.3.1. Example 1: reference frame of the Triune Continuum Theory

Let us review an example of the reference frame constructed in accordance with the Theory of Triune Continuum. In this example let us opt for the separately considered space and time dimensions (see Fig. 1).

On this example we see that to cover the modeling scope in the case of separately considered space and time dimensions we need to have the following concepts:

For the spatiotemporal continuum:

- (1) *space interval*, (2) *space outside the space interval* and (3) *point in space* as the boundary between the first two spatial concepts;
- (4) *time interval*, (5) *time outside the time interval* and (6) *point in time* as the boundary between the first two temporal concepts.

For the constitution continuum:

- (7) constitutional interval, usually called *object*;
- (8) constitutional space outside the constitutional interval, usually called *environment* of an object;
- (9) boundary between the first two constitutional concepts, usually called *interface* of an object to the environment.

For the information continuum:

- (10) information about constitution related to a point in time, that is a static information element usually called *state*;
- (11) information about constitution related to an interval in time, that is a dynamic information element usually called *action*;
- (12) information about constitution related to a point in space, can be called *spatial state*;
- (13) information about constitution related to an interval in space, can be called *spatial trace*.

These 13 concepts are necessary and sufficient to cover the general system modeling scope on the abstract level in the case of separately considered space and time dimensions. In the more general case of a single spatiotemporal dimension we will have 8 concepts.

3.3.2. Example 2: conformance of RM-ODP basic modelling concepts to the Theory of Triune Continuum

Being applied within the Triune Continuum Paradigm, the Theory of Triune Continuum allows to define the different sets of Basic Modeling Concepts that are *necessary and sufficient* for different modeling frameworks to represent their respective modeling scopes.

For the particular case of RM-ODP conceptual framework we can localize the following essential BMCs that are introduced in RM-ODP Part 2 clause 8 and are in accordance with the Theory of Triune Continuum: location in space and location in time (belonging to the spatiotemporal continuum), object and its environment (belonging to the model constitution continuum), state and action (belonging to the information continuum and representing respectively static and dynamic information about objects and their environments in space–time), interaction point and interface (belonging to the information continuum and representing respectively static and dynamic information about the boundary between objects and their environments in space–time). This gives us 8 concepts. Let us review the difference between these 8 concepts and 13 concepts from Example 1 (Section 3.3.1):

- RM-ODP *location in space* and *location in time* correspond to the *space* and *time intervals* from Example 1 in the cases of a nonzero value for “arbitrary size” mentioned in the RM-ODP definitions 2–8.9 and 2–8.10.
- RM-ODP *location in space* and *location in time* correspond to the *point in space* and *point in time* from Example 1 in the cases when “arbitrary size” mentioned in the RM-ODP definitions 2–8.9 and 2–8.10 is equal to nil.

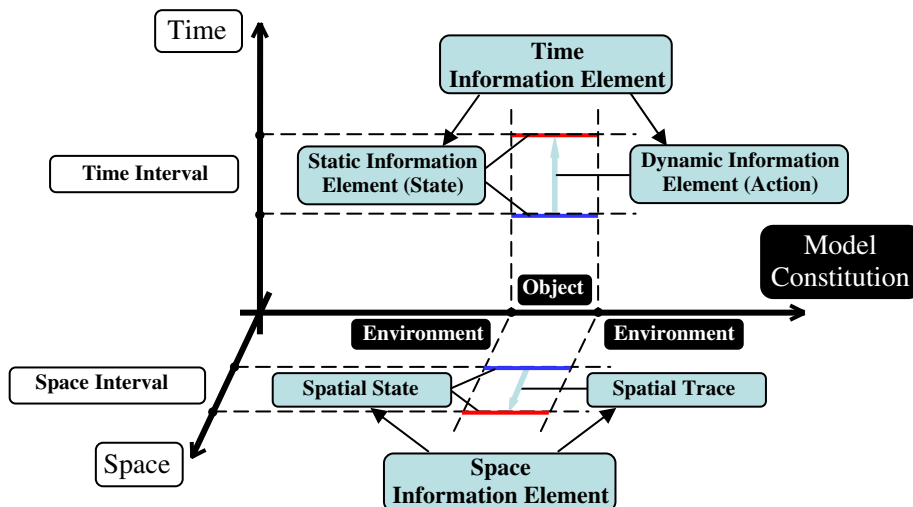


Fig. 1. An example of reference frame of the Triune Continuum Theory.

- RM-ODP does not introduce explicit basic modelling concepts corresponding to the *space and time outside space and time intervals* from Example 1. However, even if analogs to these concepts do not appear in RM-ODP definitions, such concepts (or at least their respective notions) are always implicitly present in the RM-ODP modeling activities.
- RM-ODP *object* corresponds to the *object* from Example 1.
- RM-ODP *environment (of an object)* corresponds to the *environment (of an object)* from Example 1. In this case “part of the model” in RM-ODP definition 2–8.2 will correspond to “part of the model constitution” in Example 1.
- RM-ODP *interface* as it is defined in 2–8.4 does not correspond directly to the *interface* from Example 1. For some reason in RM-ODP *interface* is considered as purely behavioral concept. In Example 1 this will correspond to the *dynamic information about the interface*. But RM-ODP also introduces a related concept of interaction point (2–8.11) that according to clarifications from 2–8.1 may be considered as correspondence to the *static information about the interface* in Example 1. Thus we may say that RM-ODP *interface* and *interaction point* correspond to the *information about the interface* in Example 1.
- RM-ODP *action* corresponds to the *action* from Example 1.
- RM-ODP *state* corresponds to the *state* from Example 1.
- RM-ODP *location in space* corresponds to the *spatial state* from Example 1 in the cases when both:
 - a) “arbitrary size” mentioned in the RM-ODP definitions 2–8.9 is equal to nil
 - b) and when in terms of Example 1 it contains information about constitution related to a point in space.
- RM-ODP *location in space* corresponds to the *spatial trace* from Example 1 in the cases when both:
 - a) “arbitrary size” mentioned in the RM-ODP definitions 2–8.9 has a nonzero value
 - b) and when in terms of Example 1 it contains information about constitution related to an interval in space.

Having reviewed this comparison we see that:

- in several cases RM-ODP Basic Modelling Concepts (introduced in RM-ODP Part 2 clause 8) are not exactly equivalent to the concepts defined within the presented application of the Triune Continuum Theory;
- regardless the absence of exact equivalences, for the RM-ODP concepts it is possible to specify the conditions in which they will correspond to the concepts from the application of the Triune Continuum Theory.

Thus we can conclude that RM-ODP Basic Modelling Concepts are conformant (to the high degree, although not absolutely) with the Theory of Triune Continuum. And so, RM-ODP Basic Modelling Concepts can be considered as necessary and sufficient (to the high degree, but not

formally) for the purposes of the RM-ODP scope representation.

Formal necessity and sufficiency can be shown in the case of an unconditional conformance of a conceptual framework to the Theory of Triune Continuum. RM-ODP conceptual framework is close to this unconditional conformance but does not formally satisfy this criterion (as we have seen in the presented Example 2).

4. Results of the formalization

Preliminary results of our discussed formalization were previously presented to the RM-ODP research community [33]. The final version of the formalization, which is too large to be presented in this paper, can be found in [30]. We formalized the part of RM-ODP standard that was identified in Section 2.3 of this paper. To perform the formalization we interpreted the definitions of the corresponding RM-ODP concepts using interpretation constraints provided by the Triune Continuum Paradigm, which are explained in Section 3. The formalization was expressed in Alloy [19], a language for the description of structural properties of a model. Alloy was chosen because of the public availability of the corresponding software tool, “Alloy Constraint Analyzer”, which allows simulation of instances of conceptual structures formalized with Alloy and representation of these instances in a graphical form. Our formalization provides denotational and Tarski’s declarative semantics for the RM-ODP conceptual framework. Note that as we explained in Section 2.1, due to the nature of denotational and Tarski’s declarative semantics [32], exactly the same formalization can be expressed using other formal description techniques.⁶

The formalization has two parts: declarations of the concepts (in terms of elements belonging to sets and possible relations between sets) and definitions of the concepts (defining logical constraints that members of the declared sets should obey). Section 4.1 presents the declarations and Section 4.2 presents the definitions.

To facilitate the readers’ interpretation of Alloy structures, let us mention here that in Alloy “markings at the ends of relation arrows denote multiplicity constraints: ! for exactly one, ? for zero or one, * for zero or more and + for one or more. Omission of a marking is equivalent to *.” [18]. Logical operators and words reserved for Alloy quantifiers and relations are [19]:

```

logic-op ::= && / || / -> / <->
negate  ::= not | !
comp-op ::= in / = / negate in / negate = / /= /
in
quantifier ::= all | some | no | sole | one
expr-op  ::= + / - / &

```

⁶ Of course, analogously to Alloy, such formal description techniques should have sufficient expressive power to express the mentioned two kinds of RM-ODP semantics.

Now let us present here the most important parts of the formalization. By presenting these parts we would like to provide readers with a basic understanding of the formalization.

4.1. Declarations

```
model RM-ODP {
  domain { ODP_Concepts }
  state {
    /* declaration of ODP concept categories
    (RM-ODP 2.5) */
    partition BasicInterpretationConcepts,
    BasicModellingConcepts,
    SpecificationConcepts : static ODP_Concepts
```

Here the set of RM-ODP concepts is partitioned into categories, corresponding to “Basic interpretation concepts”, “Basic modelling concepts” and “Specification concepts”.

```
/* declaration of "Basic interpretation
concepts" (RM-ODP 2.6) */
partition UniverseOfDiscourse,
InterpretationPossibilities : static
BasicInterpretationConcepts
partition Entity, Proposition :
UniverseOfDiscourse
partition FirstOrderProposition,
HigherOrderProposition : Proposition
holds : Proposition-> UniverseOfDiscourse+
System : Entity
Syssystem : System
```

Here we show further partitioning of the “Basic interpretation concepts” set into subsets, which correspond to the RM-ODP concepts defined in [15], Part 2 clause 6, and to some of the additional subsets, such as FirstOrderProposition and HigherOrderProposition sets.

The InterpretationPossibilities part of BasicInterpretationConcepts contains concepts of:

- Abstraction (2–6.3), which allows for different levels of details to exist when modeling the universe of discourse;
- Atomicity (2–6.4), which allows the definition of granularity for a given level of abstraction;
- Architecture (2–6.6), which introduces a set of rules that define the structure of a system in the universe of discourse.

As we see, these three concepts are defined on the meta-level for the RM-ODP metamodel framework (meta-meta-level concepts). As mentioned in Section 2.3, on the same meta-meta-level we find RM-ODP 2–7: “Basic linguistic concepts”. Since our goal was to formalize RM-ODP metamodel and not the meta-meta-view of RM-ODP modeling framework, the

InterpretationPossibilities concepts were not considered for the formalization.⁷

The partitioning into sets of FirstOrderProposition and HigherOrderProposition appears from the foundations of Triune Continuum Paradigm, it cannot be found in the RM-ODP standard [15]. This introduction of additional partitioning of the RM-ODP concepts was inevitable. It is done to apply Russell’s theory of types and, as we explained in Section 3.2 of this paper, it is necessary to perform our RM-ODP formalization.

```
/* introduction of relations between RM-ODP
concept categories */
modeledByBMC : FirstOrderProposition->
BasicModellingConcepts
modeledBySC : HigherOrderProposition->
SpecificationConcepts
```

Here we show two important Alloy declarations. They declare relations between the propositions that exist in the universe of discourse of an RM-ODP model on one side and the concepts that are used inside the RM-ODP model to represent these propositions on the other side. The former (FirstOrderProposition and HigherOrderProposition sets) belong to the “Basic interpretation concepts” set, whereas the latter concepts are “Basic modelling concepts” set and “Specification concepts” set. These two Alloy declarations define Tarski’s declarative semantics as it is suggested by the Triune Continuum Paradigm (see Section 3.1 of this paper):

- concepts from the BasicModellingConcepts set are relevant within an RM-ODP model *if and only if* there exist propositions corresponding to them in the FirstOrderProposition set in the universe of discourse of the RM-ODP model;
- concepts from the SpecificationConcepts set are relevant within an RM-ODP model *if and only if* there exist propositions corresponding to them in the HigherOrderProposition set in the universe of discourse of the RM-ODP model.

Let us now introduce two other important Alloy declarations from the formalization:

```
/* introduction of relations between RM-ODP
concept categories */
mappedToBMC : SpecificationConcepts->
BasicModellingConcepts
mappedToSC : BasicModellingConcepts->
SpecificationConcepts
```

⁷ We would like, however, to emphasize the importance of these concepts. The concepts of abstraction, atomicity and architecture provide modelers with indispensable possibilities for interpreting a Universe of Discourse within a modeling activity. A discussion about these concepts is beyond the scope of this paper; they were numerously discussed in other publications. In particular, the readers can refer to [22] that explicitly presents the wide usage of abstraction in business modeling.

These lines declare relations between the concepts from BasicModellingConcepts set and the concepts from SpecificationConcepts set. The relations are defined in agreement with the Triune Continuum Paradigm’s suggestion to use Russell’s Theory of Types for conceptual constructions within RM-ODP models: basic modelling concepts are used as the first-order propositions applied on Model Elements within RM-ODP models and specifications concepts are used as higher-order propositions applied on basic modelling concepts (see Section 3.2). Thus in an RM-ODP model containing concrete model elements there will always be a mapping between basic modelling concepts and specifications concepts.

The meaning of these two relations can be seen on the example at the end of Section 3.2 in this paper. The example shows how within a valid specification it is possible to establish the correspondence between a BMC that defines its characteristic context of applicability for SCs and an SC that is applicable within this context. Such a correspondence will not be possible within invalid specifications where a concept from SpecificationConcepts set will be applied inappropriately to characterize a concept from BasicModellingConcepts set. Thus these two relations are important for the detection of inconsistencies within specifications. The readers can also refer to [30] for additional explanations.

Let us present now partial declarations of different RM-ODP concepts.

```
/* declaration of ``Basic modelling
concepts`` (RM-ODP 2.8) */
partition Constitution, SpaceTime,
Information : static BasicModellingConcepts
partition Object, Environment : static
Constitution
environment (~object) : Object! ->
Environment!
partition StructuralInfo, BehavioralInfo :
static Information
Behavior : BehavioralInfo
State_ : StructuralInfo
partition Action, BehavioralConstraint :
static Behavior
c o r r e s p o n d i n g _ c o n s t r a i n t
(~constrained_action) : Action ->
BehavioralConstraint
partition InternalAction, Interaction :
static Action
partition InteractionPoint, Space, Time :
static SpaceTime
Interface: Behavior
Activity: Behavior
LocationInSpace : Space
LocationInTime : Time
```

Here, in correspondence with the Triune Continuum Paradigm, we first declare partitioning of BasicModelling-

Concepts set into three subsets: Constitution, SpaceTime and Information. The three subsets correspond to the three continuums defined by the paradigm as explained in Section 3.3. Another partitioning defined by the paradigm concerns Information set of concepts; we divide this set into StructuralInfo and BehavioralInfo subsets. These two subsets will contain RM-ODP basic modelling concepts that feature, respectively, *static* and *dynamic* information about objects and their environments within space–time). Then we declare a group of concepts from the “Basic modelling concepts” category of RM-ODP, with their corresponding interrelations that can be found in the standard.

```
/* declaration of "Specification concepts"
(RM-ODP 2.9) */
partitionType, Class, Instance, Composition,
Decomposition: SpecificationConcepts
TemplateType: Type
Template, InstantiationRules : TemplateType
Instantiation: Instance
partition Creation, Introduction :
Instantiation
associated_type: Class! -> Type!
member_of_class(~set_of): Instance+
-> Class!
satisfies_type(~valid_for): Instance+
-> Type!
TemplateClass: Class
associated_template_type: TemplateClass!
-> TemplateType!
member_of_template_class(~set_of_
instantiations): Instantiation+
-> TemplateClass!
subtype(~supertype): Type -> Type
subclass(~superclass): Class -> Class
specification (~instantiation):
Instantiation -> Template!
derived_class(~base_class): TemplateClass
-> TemplateClass
incremental_modification: Template ->
Template
refinement: SpecificationConcepts ->
SpecificationConcepts
```

Here we declare a group of concepts from the RM-ODP “Specification concepts” category, with their corresponding interrelations that can be found in the standard.

Note that in the last two groups of declarations, we kept the original RM-ODP concepts distribution between the “Basic modelling concepts” set and “Specification concepts” set. This was done for convenience of regular RM-ODP users, who are accustomed to the standard RM-ODP structure. This distribution of concepts can be improved to correspond to the Triune Continuum Paradigm’s distribution between BMCs and SCs. With this improvement “Basic modelling concepts” will contain only seven concepts: *Space*

and *Time intervals* (or “Location in time” and “Location in space” as defined by RM-ODP), *Object* and its *Environment*, *Interface* (or “Interface” and “Interaction point” as defined by RM-ODP), *State* and *Action* instead of the currently existing 11 concepts (see explanations in the example in Section 3.3.2). The other concepts currently belonging to the RM-ODP “Basic modelling concepts” set (Activity, Behaviour, Communication) will belong to the “Specification concepts” set. As it is shown in [30], with the improved distribution all BMCs are essentially the first-order propositions on Model Elements, and all SCs are essentially the higher-order propositions. While with the current RM-ODP distribution not all basic modelling concepts are first-order propositions on Model Elements (some of them are higher-order propositions).

After completing the Alloy declarations that essentially define the distribution of concepts in sets, we pass to the Alloy definitions that define the logical constraints that members of the sets should obey.

4.2. Definitions

```
/* definitions for "Basic interpretation
concepts" (RM-ODP 2.6) */
def FirstOrderProposition {
  all p: FirstOrderProposition | (p.holds:
Entity) }
def HigherOrderProposition {
  all p: HigherOrderProposition | (p.holds:
Proposition) }
```

Here we defined the nature of elements of Entity, FirstOrderProposition, and HigherOrderProposition sets in agreement with Russell’s Theory of Types application in the Triune Continuum Paradigm.

The other definitions that we formalized (e.g. Action, Internal Action, Interaction, Behaviour, Interface, Location in space, Location in time, Class, Instance, Type, etc.) correspond to the definitions of concepts expressed in English in the RM-ODP standard Part 2 clauses 8 and 9.

For example, let us consider here the concepts of Action, InternalAction and Interaction. For their formalized definitions we first defined an auxiliary concept of participant:

```
/* definitions for "Basic modelling
concepts" (RM-ODP 2.8) */
def participant {
  all a: Action, b: Constitution | b in a.
participant <-> (a.instant_begin.
state_existence in b.constitution_state)
&& (a.instant_end.state_existence in
b.constitution_state) }
```

Here we expressed that something from Constitution is defined as a participant of an Action if and only if the pre- and post-states of the Action are in the allowed states of the constitutional element under consideration.

Then we were able to define Action:

```
def Action{
  all a: Action | (a.instant_begin
!=a.instant_end) && (a.instant_begin.
state_existence != a.instant_end.
state_existence) && (a.participating_
object in a.participant) }
```

We can notice two parts in this Action definition. The first is the state difference in the beginning of the action (a.instant_begin) and in the end of it (a.instant_end), which reflects RM-ODP 2–8.3 definition statement that something should happen to be an action. And the second part of the definition is the fact that there should be an object among action participants, this reflects the RM-ODP definition associating action with at least one object.

To define InternalAction and Interaction we need to say that in the first case the environment of the participating object does not participate in the action, and in the second case it does participate:

```
def InternalAction {
  all a: InternalAction | a.
participating_object in a.participant
-> a.participating_object.environment not
in a.participant }
def Interaction {
  all a: Interaction | a.participating_object
in a.participant-> a.participating_object.
environment in a.participant }
```

As another example, here are the definitions for the concepts of Class and Instance:

```
/* definitions for "Specification
concepts" (RM-ODP 2.9) */
def Class {
  all c: Class | some i | one t | i.
satisfies_type = t && i in c.set_of &&
i.member_of_class = c && c.associated_type
= t }
def Instance {
  all a: Instance | some t | a.satisfies_type = t
}
```

The remaining formalized definitions, as well as more detailed explanations of the formalization, can be found in [30].

This concludes the presentation of the most important parts of our formalization of RM-ODP.

5. Conclusion

In this paper we have presented our contribution to the RM-ODP standard-related research. This contribution resolves an important problem of the RM-ODP standard: the absence of a single consistent formalization of the RM-ODP conceptual framework. A realization of such formalization was officially

targeted by the ISO/ITU standardization committee as a goal of the last (the fourth) part of RM-ODP. But neither the standard itself nor the standard-related research had proposed a single consistent formalization. We have defined and implemented such formalization. Our formalized structures correspond to RM-ODP Part 2 clauses 5 (Categorization of concepts), 6 (Basic interpretation concepts), 8 (Basic modelling concepts) and 9 (Specification concepts).

The definition of our formalization is based on the Triune Continuum Paradigm, an object-oriented paradigm for general system modeling, which provides a logically rigorous theoretical base for organization of conceptual frameworks that are used for system modeling in different contexts (e.g. in software development, in enterprise architecture, in the architecture of financial services, in jurisprudence, etc.). With the aid of the Triune Continuum Paradigm, our formalization provided formal denotational and Tarski's declarative semantics for the RM-ODP concepts. Thanks to the paradigm, the formalization is internally consistent and it features logically coherent means of interpretation of a subject of modeling interest.

We implemented our formalization of RM-ODP in a computer-interpretable form. This was done with the aid of Alloy, a language for the description of structural properties of a model. The resulting models were verified with the associated software tool, Alloy Constraint Analyzer.

The goal of our work is to help promote the practical applications of RM-ODP. The formal model of RM-ODP Part 2 that we presented in this paper can indeed serve for the promotion of RM-ODP towards a wider use in the modern modeling practices. Some of the applications of our results have already justified this claim.

Firstly, our formalization can serve as a foundation for the development of RM-ODP-based computer-aided design tools. An example of such a tool employing our formalization can be found at [23].

Secondly, the formal model of RM-ODP foundations can help define semantics of other existing modeling languages such as UML. Indeed, as it is presented in [31], the existing official definition of the UML framework [35] lacks both internal consistency and a logical order in its organization. Whereas our formalization of RM-ODP framework is devoid of these flaws and offers a rigorously structured meta-model.

Thirdly, our formalization can serve in numerous practical cases of system modeling applications. For example, in [1] the formal meta-model of RM-ODP is used to specify some of the behavioral constraints relevant for system modeling. Another example of application of our work on RM-ODP was presented at [44]: using the RM-ODP standard, some of the results from systems sciences and some of the foundational ideas of the Triune Continuum Paradigm [44] introduced an ontology for system modeling. With these results, our formalization of RM-ODP helped to define Systemic Enterprise Architecture Methodology [43], a method developed to address business and IT alignment through the systematic modeling of business and IT systems.

Thus, the formalization that we have introduced in this paper adds practical value to the RM-ODP standard.

References

- [1] P. Balabko, A. Wegmann, From RM-ODP to the formal behavior representation, in: H. Kilov, K. Baclawski (Eds.), *Practical Foundations of Business and System Specifications*, Kluwer Academic Publishers, September 2003, pp. 41–66.
- [2] C. Bernardeschi, J. Dustzadeh, A. Fantechi, E. Najm, A. Nimour, F. Olsen, Transformations and Consistent Semantics for ODP Viewpoints, in: H. Bowman, J. Derrick (Eds.), *Proceedings of Second IFIP conference on Formal Methods for Open Object-based Distributed Systems—FMOODS'97*, Chapman and Hall, July 1997.
- [3] G.S. Blair, J.-B. Stefani, *Open Distributed Processing and Multimedia*, Addison Wesley Longman Ltd., 1998.
- [4] E.A. Boiten, H. Bowman, J. Derrick, P.F. Linington, M.W.A. Steen, Viewpoint consistency in ODP, *Computer Networks* 34 (3) (August 2000) 503–537.
- [5] H. Bowman, E.A. Boiten, J. Derrick, M.W.A. Steen, Viewpoint consistency in ODP, a general interpretation, in: E. Najm, J.-B. Stefani (Eds.), *Proceedings of the First IFIP International Workshop on Formal Methods for Open Object-Based Distributed Systems*, Chapman and Hall, March 1996, pp. 189–204.
- [6] H. Bowman, J. Derrick, P. Linington, M. Steen, Cross-viewpoint consistency in open distributed processing, *IEE Software Engineering Journal* 11 (1) (January 1996).
- [7] H. Bowman, J. Derrick, P. Linington, M. Steen, FDTs for ODP, *Computer Standards and Interfaces* 17 (September 1995) 457–479.
- [8] S.M. Brien, J.E. Nicholls, *Z Base Standard version 1.0*, Technical Monograph PRG-107, Programming Research Group, Oxford University, November 1992.
- [9] F. Durán, A. Vallecillo, Writing ODP Enterprise Specifications in Maude, in: J. Cordeiro, H. Kilov (Eds.), *Proceedings of ICEIS 2001, Workshop On Open Distributed Processing—WOODPECKER'2001*, Setúbal, Portugal, July 2001.
- [10] H. Ehrig, B. Mahr, *Fundamentals of algebraic specification*, EATCS Monographs on Theoretical Computer Science, vol. 6, Springer-Verlag, 1985.
- [11] G. Frege, The Russell Paradox, in: G. Frege (Ed.), *The Basic Laws of Arithmetic*, University of California Press, Berkeley, 1964, pp. 127–143.
- [12] A. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, B. Nuseibeh, Inconsistency handling in multiperspective specifications, *IEEE Transactions on Software Engineering* 20 (8) (August 1994) 569–578.
- [13] ISO 9074, Estelle, a Formal Description Technique Based on an Extended State Transition Model, 1997.
- [14] ISO/IEC 880, LOTOS—A Formal Description Technique Based on the Temporal Ordering of Observational Behavior, 1989.
- [15] ISO, ITU-T, *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. http://isotc.iso.ch/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm.
- [16] ISO, ITU-T, *ISO/IEC 19793 | ITU-T Recommendation X.906. "UML for ODP"*, draft 2003–2005.
- [17] ITU-T, Recommendation Z.100, "CCITT Specification and Description Language (SDL)", 1993.
- [18] D. Jackson, *A Comparison of Object Modelling Notations: Alloy, UML and Z*. MIT Lab for Computer Science, August 1999.
- [19] D. Jackson, *Alloy: A Lightweight Object Modelling Notation*, *ACM Transactions on Software Engineering and Methodology* 11 (2) (2002) 256–290.
- [20] D. Johnson, H. Kilov, An Approach to an RM-ODP Toolkit in Zeta, *Proceedings of the 1st Workshop on Component-Based Systems*, in conjunction with European Software Engineering Conference (ESEC) and ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE), Zurich, Switzerland, 1997.
- [21] D.R. Johnson, H. Kilov, Can a flat notation be used to specify an OO system: using Z to describe RM-ODP constructs, in: E. Najm, J.-B. Stephani (Eds.), *Proceedings of FMOODS96: IFIP WG 6.1 Conference on Formal Methods in Object-oriented Distributed Systems*, Chapman and Hall, Paris, March 1996, pp. 407–418.

- [22] H. Kilov, *Business Models*, Prentice Hall, 2002.
- [23] L.S. Lê, A. Wegmann, Definition of an Object-Oriented Modeling Language for Enterprise Architecture, Proceedings of HICSS-38, Hawaii USA, January 2005.
- [24] P.F. Linington, J. Derrick, H. Bowman, The specification and conformance of ODP systems, Proceedings of 9th International Workshop on Testing of Communicating Systems, IFIP TC6/WG6.1, Chapman and Hall, Darmstadt, September 1996, pp. 93–114.
- [25] L. Logrippo, M. Faci, M. Haj-Hussein, An introduction to lotos: learning by examples, *Computer Networks and ISDN Systems* 23 (1992) 325–342.
- [26] E. Najm, J.-B. Stefani, A formal semantics for the ODP computational model, *Computer Networks and ISDN Systems* 27 (1995) 1305–1329.
- [27] E. Najm, J.-B. Stefani, Computational models for open distributed systems, in: H. Bowman, J. Derrick (Eds.), *Proc. of FMOODS'97*, Chapman and Hall, Canterbury, 1997.
- [28] A. Naumenko, A report on the Triune Continuum Paradigm and on its foundational theory of Triune Continuum, in: J. Castro, E. Teniente (Eds.), *PHISE'05, the 1st International Workshop on Philosophical Foundations of Information Systems Engineering*, Proceedings of the CAISE'05 Workshops, vol. 2, FEUP edições, Porto, Portugal, June 2005, pp. 439–450.
- [29] A. Naumenko, Basics of the Triune Continuum Paradigm, in: M. Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology*, Idea Group Inc., January 2005.
- [30] A. Naumenko, Triune Continuum Paradigm: a paradigm for General System Modeling and its applications for UML and RM-ODP, PhD Thesis number 2581, Swiss Federal Institute of Technology—Lausanne (EPFL), June 2002.
- [31] A. Naumenko, A. Wegmann, A Metamodel for the Unified Modeling Language, in: J.-M. Jézéquel, H. Hussmann, S. Cook (Eds.), *Proceedings of UML 2002, the 5th International Conference on the Unified Modeling Language, "UML" 2002—The Unified Modeling Language: Model Engineering, Concepts, and Tools*, LNCS 2460, Springer-Verlag, Dresden, September 2002, pp. 2–17.
- [32] A. Naumenko, A. Wegmann, C. Atkinson, The Role of Tarski's Declarative Semantics in the Design of Modeling Languages, Technical report, vol. IC/2003/43, Swiss Federal Institute of Technology, Lausanne, April 2003, EPFL.
- [33] A. Naumenko, A. Wegmann, G. Genilloud, F. Frank, Proposal for a formal foundation of RM-ODP concepts, in: J. Cordeiro, H. Kilov (Eds.), *Proceedings of ICEIS 2001, Workshop On Open Distributed Processing—WOODPECKER'2001*, Setúbal, Portugal, July 2001.
- [34] OMG, Profile for Relationships, Version 1.0, <http://www.omg.org/cgi-bin/doc?formal/2004-02-07>, February 2004.
- [35] OMG, Unified Modeling Language Specification, Version 1.5, <http://www.omg.org/uml>, March 2003.
- [36] J.R. Putman, *Architecting with RM-ODP*, Prentice Hall, 2001.
- [37] B. Russell, Mathematical logic as based on the theory of types, *American Journal of Mathematics* 30 (1908) 222–262.
- [38] R.O. Sinnott, K.J. Turner, Applying Formal Methods to Standard Development: The Open Distributed Processing Experience, *Computer Standards and Interfaces* 17 (1995) 615–630.
- [39] R.O. Sinnott, K.J. Turner, Specifying ODP Computational Objects in Z, Proceedings of 1st International Workshop on Formal Methods for Open Object-Based Distributed Systems, Paris, France, March 1996, pp. 375–390.
- [40] J.M. Spivey, *The Z Notation, A Reference Manual*, International Series in Computer Science, 2nd edition, Prentice-Hall International, 1992.
- [41] M.W. Steen, J. Derrick, ODP Enterprise Viewpoint Specification, *Computer Standards and Interfaces* 22 (3) (August 2000) 165–189.
- [42] A. Tarski, *Logic, Semantics, Meta-mathematics*, Oxford University Press, 1956.
- [43] A. Wegmann, On the Systemic Enterprise Architecture Methodology (SEAM), Proceedings of ICEIS 2003, Anger, France, April 2003.
- [44] A. Wegmann, A. Naumenko, Conceptual Modeling of Complex Systems Using an RM-ODP Based Ontology, Proceedings of the 5th IEEE International Enterprise Distributed Object Computing Conference—EDOC 2001, Seattle, USA, September 2001.
- [45] I.M. Yaglom, *Mathematical structures and mathematical modeling*, Gordon and Breach Science Publishers, 1986.



Andrey Naumenko was born in Sevastopol, USSR in 1974. He received Bachelor of Natural Sciences (1995) and Master of Science in Physics (1996) degrees from MIPT (Phystech). Andrey performed his doctoral research specializing on the analysis of Information Systems. He earned his Doctor of Sciences degree at EPFL in 2002 after having spent several years working for the projects of IT and telecom companies. His research interests are in the area of Information Systems engineering and information management for multifarious contexts of mankind's activity.



Alain Wegmann earned his EE degree at EPFL in 1981. He worked for 14 years with Logitech (Switzerland, Taiwan, US) in positions ranging from software developer, IS manager, manufacturing engineering to VP engineering and VP OEM marketing. He joined the EPFL university in 1997 as professor. His teaching and research interests are in enterprise architecture and business/IT alignment.