Metamodeling Approach for the Design of Cyber-Physical Systems

Vitaliy Mezhuyev^{1*} and Refik Samet²

¹Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, Gambang, Malaysia; mejuev@ukr.net ²Department of Computer Engineering, Ankara University, Ankara, Turkey; samet@eng.ankara.edu.tr

Abstract

Background/Objectives: The design of Cyber-Physical Systems (CPS) needs an approach that allows us to model effectively their heterogeneous components, having both a computational and a physical nature. To contribute to this area, a new metamodeling approach for designing CPS is proposed in this paper. **Methods:** The metamodeling approach allows CPS designers the development of the metamodels in the different mathematical semantics (e.g., graph-based, vector-based, geometry-based). This is possible due to the introduction of the additional level of the metamodeling architecture (*M4*), which expresses the studied domain in terms of sets. To prove that *M4* is mathematically sufficient for producing all considered in the paper meta-metamodels (*M3s*), we give definitions of *M3s* in terms of *M4*, i.e. as having algebraic structure subsets. **Results:** To prove the concept, a geometrical meta-metamodel Ω , linking physical properties of multidimensional domains with their spatial structure, is proposed. Structural analysis of physical models showed the possible alphabet of Ω , which includes corresponding to the dimensions of the space metatypes; this has been a common result of abstraction from a geometrical structure of physical objects. Using the geometrical meta-metamodel Ω allows us to unify the representation of the spatial information at different levels of the CPS design. **Conclusion/Application:** The software tool that implements the proposed metamodeling approach is currently under development by the authors. The tool enables designers to express the syntax and semantics of metamodels and apply them for modelling CPS.

Keywords: Cyber-Physical System, Domain-Specific Modeling, Geometrical Meta-Metamodel, Metamodeling

1. Introduction

Along with the possibility of covering multiple domains, universal modelling languages, the most prominent of which is now Unified Modelling Language (UML), have a lot of drawbacks. The necessity of models development for a wide range of domains leads to the situation, that modelling concrete aspects of specific domains remains outside of possibilities of the universal languages¹. The idea of Domain-Specific Modeling (DSM) is the development of special languages that allow us to effectively capture the domain properties for use in software system design. Such Domain-Specific Languages (DSLs) with the help of a certain meta-metamodel are developed. There are several meta-metamodels, for example, the GOPPRR (Graph-Object-Property-Port-Role-Relationship) is used in the MetaEdit+ toolset². There are multiple examples of using graphs for metamodeling³, which provide a natural way for the decomposition of domains into hierarchical structures. Emphasizing the power of existing approaches, in this paper we propose a metamodeling architecture that allows us to adequately capture and represent the specific properties of multidimensional cyber-physical domains.

*Author for correspondence

The proposed approach is important due to its applicability to the design of Cyber-Physical Systems (CPS), i.e., systems that contain both computational and physical elements, with a close integration of physical processes and information processing⁴. The design of CPS should provide new types of abstractions that are applicable to modeling both physical domains and the corresponding computational domains. As multiple authors have noticed, standard abstractions do not work here⁵.

First, existing metamodeling approaches do not allow us to express the different mathematical semantics of the multiple abstraction levels of a CPS design. Second, there are difficulties in capturing spatial information, which are related to both the location of the actions and the use of location information in defining the actions⁶. This information is stored in various forms, while its unique representation at the different levels of a CPS design is needed.

To meet the challenges of CPS design, we propose a new metamodeling approach that is based on the following principles:

- allocation of an additional level of metamodeling architecture, which allows the development of metamodels in different mathematical semantics;
- unification of representing spatial information, both for the physical and computational levels of a CPS design;
- direct use of the physical models as basic objects (abstractions) of DSLs for CPS design;
- embedding physical semantics in the modeling languages by linking the metamodeling approach with the mathematical theory of modeling;
- definition of grammars, guaranteeing the development of syntactically correct constructs for both the physical and computational parts of the modeling languages.

This paper is organized as follows. In Section 2, the principles of the proposed metamodeling architecture are discussed. In Section 3, the analyses of physical models to allocate the abstractions for the geometrical metametamodel for CPS design is given. In Section 4, the formal semantics of the geometrical meta-metamodel is discussed. Applicability of the approach to development of meta-metamodels in different mathematical semantics is shown. In Section 5, the implementation of the proposed approach is briefly considered. Conclusions and plans for future research finalize this paper.

2. Metamodeling Architecture

In general, the approaches for modeling domains can be divided into two parts: 1) using a so-called General Purpose Language (GPL) or 2) developing a DSL. Although existing GPLs are good for expressing computational domains, they are not suitable for modeling physical domains. At the same time, physical modeling languages do not allow us to express data structures and computational processes.

Thus, the design of CPS requires the development of an approach, that allows us to express heterogeneous semantics of interlinked physical and computational domains. The challenge here is to conjoin abstractions that have evolved over centuries for modeling physical processes (differential calculus, stochastic processes, etc.) with abstractions that have evolved over decades in computer science (data and algorithms). The former abstractions focus on the dynamics of a system via the evolution of its states over time, whereas the latter focus on the processes of transforming data⁴.

To develop a new metamodeling approach that is applicable to expressing the semantics of both the physical and computational domains, let us first briefly consider the existing approaches.

The methodology MOF (Meta Object Facility)⁷ was used by the OMG (Object Management Group) consortium for the development of the UML. MOF has four levels in its metamodeling architecture. The top level is the meta-metamodel (M3), which defines the language for the development of metamodels (having the level M2). The level M2 (here, UML) is used for the development of domain models of level M1 (the UML-models). The last is the level of data (M0), which describes the concrete instances of M1. The MOF meta-metamodel is based on the object-oriented methodology of software systems design.

Note, UML is applicable for modelling software intensive systems, but properties of physical domains remain outside of its possibilities. To support the modelling CPSs, OMG consortium have to develop a new metamodel. But its development will not solve the general problem of the MOF meta-metamodel adaptation for concrete domains.

The underlying reason is that different domains have quite different mathematical structure and properties of elements, and taken in this paper CPS is just an example of it. Thus the task arose to develop the method, which allows us to model heterogeneous domains, having different mathematical structure⁸. We cannot develop such the method being inside of the limited by M3 metamodelling architecture⁹. This why we need an additional level of the metamodelling architecture (M4), intended for expression of mathematical structures, with its domain adaptation at the level M3.

The meta-metamodel GOPPRR allows designers to produce metamodels inside graph-based notations by setting relationships between objects, the definitions of domain properties (attributes) and the roles². Each of the GOPPRR concepts is called a metatype. As MOF, the metamodeling architecture of GOPPRR can be shown in four levels (Figure 1).



Figure 1. The GOPPRR metamodeling architecture.

The proposed approach also has a multiple-level metamodeling architecture, but its semantics differ from existing methodologies. All of the metamodels are considered to be formal systems; mathematically, metamodel at each level of the proposed metamodelling architecture is a triple, which contains the alphabet A (the carrier of the formal system), the grammar G and the operations O

$$M2 = \langle A, G, O \rangle \tag{1}$$

We introduce the additional level of the metamodeling architecture - the meta-meta-metamodel (M4), as a formal system that is built on the basis of set theory. M4 includes the meta-metatype "element of a set", set operations and grammar rules, which (taken together) allow us to specify a set structure. This approach allows us to consider a domain as a set of heterogeneous entities linked by different mathematical structures (algebraic, logical, geometrical, etc.).

All of the levels of the proposed metamodeling architecture contain not only descriptive parts, such as in

MOF or GOPPRR, but also procedural parts (which are implemented with software functions).

The implementation of the procedural parts at all levels of the proposed metamodeling architecture forms the Application Program Interface (API) of the corresponding tool. The API of M4 contains the methods for manipulation with the elements of a set of composing domain entities. The API of M3 is the operations with subsets (e.g., with a node and an edge of a graph, and in the general case with any model objects of the considered domain). For M2, the API contains the metamodel processing routines (here, the metatypes of the level M3 become domain-specific types, i.e., to the mathematical subsets the semantics of the domain is assigned). M1 contains instances of the types and definitions of domainspecific methods, implemented with the APIs of all the previous levels. M0 is data values and processes in the computer memory (instances of the methods, defined at the level *M1*).

Following our proposal, the architecture for the development of the graph-based metamodel is shown in Figure 2. Here, a node and an edge of a graph serve as the metatypes for the development of the metamodel types (an attribute is the inherent part of a node and an edge). At the same time, the node and the edge are produced from the meta-meta-metamodel as having algebraic structure subsets. Note that while GOPPRR² and MGA³ also use the graphs for structuring domain properties, this is a partial case of the proposed approach, in which the development of the meta-metamodels in the different mathematical semantics is possible (e.g., in the following sections of this paper, we will consider an example of the development of a geometrical meta-metamodel).



Figure 2. The levels of the proposed metamodeling architecture.

The first stage of the metamodeling and, accordingly, the highest level of abstraction, is consideration of domain as a set of heterogeneous entities D. Analyses of mathematical structure and domain-specific properties of D are performed, respectively, at the levels M3 and M2. At the level M3 the elements of D are structured as the metatypes MT, which next are used to build domain specific types T at the level M2. The essence of the metamodel adaptation is linking defined at the level M3mathematical structures with domain attributes at the level M2.

The following relationships are between carriers of the previous (prototype) and the next (image) levels of the proposed architecture:

 $M_{43}: D \to MT, \qquad M_{43} \subseteq D \times MT,$ (2)

 $M_{32}: MT \to T, \qquad M_{32} \subseteq MT \times T,$ (3)

 $M_{21}: T \to I, \qquad \qquad M_{21} \subseteq T \times I, \tag{4}$

where *D* is the set of domain entities, MT - metatypes, *T* - domain specific types and *I* - instances of the types. M_{43} , M_{32} , M_{21} are the mappings, used respectively for domain structuring, metamodel adaptation and types instantiation (Figure 3).



Figure 3. Mappings of domain structuring, metamodel adaptation and types instantiation.

Mappings M_{43} , M_{32} , M_{21} are homomorphic, i.e. defined on *D* operations are applicable to the equivalent structures in the sets *MT*, *T* and *I*. The transition M_{43} from *M4* to *M3* is the mathematical structuring of domain *D*. In accordance with the M_{43} in the set *D* the subsets $\{d_1, d_2 \dots d_n\}$ are allocated, and form the elements of *MT* (metatypes):

$$M_{43}(\{d_1, d_2 \dots d_n\}) = MT,$$

$$\forall mt \in MT : \exists \{d_1, d_2, \dots d_n\} \in D, n = \overline{1 \dots |D|}$$
(5)

Depending on the mathematical structure of domain, the mathematical types MT can be the model objects of any mathematical apparatus, e.g. nodes and edges of a graph, geometrical figures, vectors etc.

3. Analyses of Physical Models and the Development of a Geometrical Meta-Metamodel

To allocate the abstractions of the meta-metamodel for CPS design, let us first analyze the well-known physical models, describing the properties and behavior of the physical domains. In this paper, we will consider the physical property to be the main element of physical knowledge to be acquired in the process of cyber-physical modeling.

Physical properties define the semantics of physical models, e.g., the mass m shows the gravitational and inertial properties of the bodies, the charge q - the electromagnetic properties, and so on. The physical properties exactly cause the possible forms of physical interactions (gravitational, electromagnetic, strong and weak) and thus define the behavioral semantics of a physical system. The values of the physical properties describe the state and the state transitions – the dynamics of a physical system. This arrangement allows us to link the physical modeling approach with the models of behavior, that are used for software systems design.

The principal point is the existence of physical domains in space and time, which is reflected by the spatial and time parameters in the corresponding physical formulas. The properties of space and time are always behind the physical models, which as the essential layer of metamodeling architecture for CPS design can be recognized.

The important aspect here is to define how the physical properties with the spatial and temporal structure of a cyber-physical domain are linked. We propose to learn the properties of the physical models from the point of view of their spatial (geometrical) structure and to allocate the methods of distribution of the physical properties among it. Such analyses will show the invariant spatial structures of the physical models, which we plan to use as the basic abstractions of the meta-metamodel for the CPS design.

Note that the assumption about the spatial localization and distribution of physical properties is one of the most often used abstractions in physical modeling. For example, the model of a material point is well-known in physics. The material point is the model of a physical object in an abstraction that is taken from its shape, size and spatial structure. In other words, we consider the properties of a physical object to be located at a geometrical point.

There are other well-known models and abstractions in physics that are based on the concept of the spatial distribution of physical properties. For example, the linear distribution of a physical property in the model of a thin conductor with a current has been used; the model of the surface distribution of a charge and different types of the volumetric distributions of scalar and vector properties are also well known in physics (e.g., the scalar field of the potential of a system of charges or the vector field of magnetic induction that is produced by a conductor with a current).

Thus, abstractions with respect to the spatial distributions of physical properties are widely used when developing models from different physical domains. This approach allows us to allocate the next invariants, which are localized in point, linear, superficial, and volumetric spatial distributions of the physical properties. The set of the corresponding geometrical objects, i.e., the point, the line, the surface and the region in 3D space, we define as the metatypes of the geometrical meta-metamodel. Table 1 shows examples of physical models that can be derived from abstractions.

From Table 1, it follows that operating the geometrical meta-metamodel allows designers to develop metamodels for modeling different physical domains. The idea of the definition of the physical metamodel is the concretization of the abstract geometrical concepts by attributing to them physical properties that have concrete values at the level of a physical model development.

It is very important that the generality of the abstractions of the meta-metamodel results in the applicability of a wide spectrum of mathematical methods. For example, the solutions of the one-, two- and threedimensional physical problems are based on modeling the relevant distribution of the properties (linear, surface and volumetric). Here, integration and differentiation of the first, second and third orders can be correspondingly applied. Thus, the development of a meta-metamodel in which designers are allowed to manipulate the basic spatial distributions of the physical properties facilitates the process of mathematical modeling. For example, in the case of an application of differential calculus, the use of the "point" as a carrier of an elementary physical property allows us to perform the analysis of infinitesimal changes in values, characterizing the dynamics of a physical system.

Table 1.	Analyses of physical models by the spatial				
distribution of properties					

The concept of the meta- metamodel	The physical abstraction or model (used as the type of metamodel)		
	Material point, point charge, source of oscil- lations, etc.		
Point(s)	System of material points; dipole, model of a polar and non-polar molecule; model of a substance (an ideal gas, an ideal liquid, model of an absolutely solid and elastic body, etc.); electronic gas, model of an electric current, etc.		
Line	The model of a thread, of a string, of a chain; contour with a current; abstractions such as a trajectory, force line of a field, etc.		
Surface	Surface distributions of a charge (model of a condenser), of mass (e.g., model of inertia moment) and other physical properties.		
Region	Vector (of force, of intensity, of speed, etc.) and scalar (of potential, a distribution of the intensity of light waves, etc.) fields.		

CPS link cyberspace with the physical world through a network of interrelated elements, such as sensors and actuators, robotics, and computational engines¹⁰. The network of heterogeneous distributed sensors is an essential part of CPS - it is used to control machines, regulate and verify the quality of physical, chemical and biological processes, measure and transmit environmental parameters, check the status of buildings, etc.

The model of sensor networks includes elements deployed in the model of the physical world for the purpose of monitoring certain phenomena of interest. The sensors perform certain measurements, which result in data, and they transmit it to a server over some physical channels. Data acquisition always begins with a physical property (e.g., the temperature, acceleration, pressure) to be measured. The physical property must be transformed into a unified form that can be sampled by a data acquisition system.

Thus, modeling sensor networks is the essential part of CPS design. In the analyses given above, we emphasized the physical part of CPS, in which objects of geometrical meta-metamodel as carriers of physical properties were considered. However, the generality of the proposed approach allows us to use the geometrical meta-metamodel for modeling sensor networks, in which geometrical objects are used for the acquisition of corresponding physical properties. For example, the types of the metamodel for modeling sensor networks in CPS are the points of interest and the regions of interest, which are linked by connectors.

To show the applicability of the proposed approach for producing the graph-based metamodels, we can refer to the metamodel of interacting entities¹¹, which was used for the development of a real-time operation system^{12,13} and for modeling distributed parallel real-time software^{14,15}. This metamodel includes the alphabet, which contains the standard parallel programming types of synchronization objects (critical section, mutex, semaphore, resource, FIFO, etc.) and types of software tasks (drivers, applications, etc.); grammar rules, which specify the valid interactions of software tasks via synchronization objects; and operations, which are used for the definitions of the code-generation functions.

At the level of the metamodel development to the nodes and the edges of the meta-metamodel, the semantics of the domain is assigned. For example, the node is the metatype for the definition of the types of software tasks and synchronization objects, and the edge is the metatype for the definition of the types of channels (communication protocols) between the tasks and synchronization objects. Note, that an application of graph-based metametamodels for software system design is a topic that is highly developed in the literature¹⁶. In the next sections, we will emphasize the possibility of using the proposed architecture for the development of the meta-metamodel for CPS design.

A comparison of the graph-based software metamodel and the geometry-based sensor network metamodel is given in Table 2.

Thus, the proposed approach allows us to link in the same metamodeling notation the different abstraction layers of the CPS design. The unique representation of the models of physical domains, software components, sensor nodes, etc. allows us to effectively express and manipulate spatiotemporal information of the cyber-physical world. The advantage is the mapping properties of the CPS into spatial structures, defining the location, geometrical forms and topology. This approach also provides the formal base for metamodeling, allowing us to use mathematical

Domain Level	Distributed concur- rent software system		Sensor network	
М3	Alphabet	Operations	Alphabet	Operations
	Node, Edge	Add/ remove a node/edge, find a path, select a	Point, Line, Surface, Region	Add/remove a figure, move, resize, compose, set/get a
		subgraph, etc.		property, etc.
M2	Software task, synchro- nization object, channel of com- munica- tion, etc.	Send/ receive data, syn- chronize software tasks, allocate resource, etc.	Point of interest, region of interest, connec- tor, etc.	Define a spa- tial location, calculate sensing radius, build topology, etc.

Table 2. Metamodels for software and the sensor partsof a CPS

methods for the analyses of physical and computational domains and leading to the minimization of development errors. Let us consider the formal base of our approach in more detail.

4. Formal Semantics of the Geometrical Meta-Metamodel

The previous section's analysis shows the invariant geometrical structures, which we take as basic abstractions for the development of the metamodels for modeling cyber-physical domains.

The idea of the proposed approach is to take a combination of the geometrical information { Γ_k }, defining the spatial structure of a cyber-physical domain and the specific information { F_k }, which is given on the geometry { Γ_k }, where k = 1,...,K, and K is the number of objects that compose the model of the cyber-physical domain.

We define the set of geometrical objects as

$$\Gamma_k^n, k = 1, ..., K, n = 0, 1, 2, 3$$
 (6)

In each point of (6), the domain-specific information F(x,y,z,t) about the properties of the cyber-physical domain is given and described in the form of $F_k(x,y,z,t)$, i.e., as the constriction (narrowing) of the function F on the respective object { Γ_k }

$$F|_{\Gamma_k^n} = F_k|_{\Gamma_k^n}, k = \overline{1, K},$$
(7)

where *n* is the parameter that shows the dimension of the geometrical object $\{\Gamma_k^n\}$ and also defines the metatype symbols of the metamodel:

- if n = 0, then {Γ_kⁿ} is a 0-dimensional (0D) object or the point P;
- if n = 1, then $\{\Gamma_k^n\}$ is a 1D object or the line *L*;
- if n = 2, then $\{\Gamma_k^n\}$ is a 2D object or the surface *S*;
- if n = 3, then {Γ_kⁿ} is the region D in the three-dimensional space D³.

In the case of n = 1 and n = 2, the narrowing of F_k are the traces of the functions F(x, y, z, t) on the corresponding lines Γ_k^1 or surfaces Γ_k^2 .

The formulas stated above constitute the definition of the alphabet of the metatypes of the geometrical metametamodel Ω .

The full definition of the geometrical meta-metamodel (which corresponds to the level *M3* of the metamodeling architecture)

$$\Omega = \{\{MT\}, \{R_{\Omega}\}, \{O_{\Omega}\}\}$$
(8)

where $MT = \{P, L, S, D\}$ is the alphabet of the metatypes. The metatypes *P*, *L*, *S*, and *D* are produced from the level *M4* in such a way that they are having mathematical structure subsets (of geometrical points, in this case). For this task, *M4* includes the operations on sets: union $A \cup B$, intersection $A \cap B$, complement $A \setminus B$, and supplement \overline{A} , where *A*, *B* are two sets. These operations are used to formulate the system of relationships R_{Ω} , which are needed for the specification of the metamodel syntax (Table 3).

The mathematical operators O_{Ω} are the significant part of the geometrical meta-metamodel. They include differential and integral operators, the methods for data approximation, etc. As an example, the operators of interpolation, interlination, and interflatation $\{O_P, O_L, O_F\}$ are shown in Table 3. Such the metamodel for the mathematical modeling surface of a body on the basis of radar data was used. For a mathematical description of these methods, we refer to¹⁴.

The meta-metamodel Ω , used for the development of the cyber-physical metamodel Φ , is given as follows

$$\Phi = \{\{T\}, \{G_{\Phi}\}, \{O_{\Phi}\}\}$$
(9)

where *T* is the type of the objects produced from *MT* inside the grammar R_{Ω} .

 G_{Φ} is the set of rules (the grammar) that defines the syntax of the model M of the cyber-physical domain, and

 O_{Φ} is the set of methods for processing the cyber-physical model.

Development of the metamodel Φ inside the metametamodel Ω has the following steps:

- composition of the elements of *MT* to create the geometrical types *T* of the metamodel Φ (by resizing, parallel movement, rotation, distribution in space, and changing other geometric attributes);
- formulation of the rules of the grammar G_{Φ} by putting restrictions on the instances of the types *T* (e.g., by producing a line segment, by the limitation of a spatial area by planes);
- attributing *T*, i.e., adding to the geometrical objects physical properties, measured by physical values;
- defining the mathematical methods O_{Φ} applicable to the cyber-physical model M.
- Inside Φ , the model M of a cyber-physical system is

$$M = \{\{\Gamma\}, \{F\}\}$$
(10)

where Γ is the set of geometrical objects used to define the spatial structure of the physical domain, and *F* is the set of functions that define the distributions of the physical properties on Γ .

Specified in the points of the geometrical object Γ_k^n , the function F depends on m parameters $S_0, S_1, ..., S_m$: $F_k(S_0, S_1, ..., S_m)$, which can be time, some operators or functionals from F (e.g., velocity, acceleration), or some integral characteristics (e.g., surface, volume, total charge). More details are given in¹⁷.

The process of the model M development inside the metamodel Φ has the following steps:

- specification of the parameters of space and time (number of dimensions, time interval, etc.);
- creation of instances of the types *T*;
- changing geometrical attributes of the instances of *T*;
- specification of the distributions of *F* on Γ (as sets of numerical values);
- generation of the data structures from the traces of *F* on Γ;
- application of the methods O_{Φ} of calculation;
- interpretation of the results of the modeling and improvement of M.

While the purpose of the paper is to define geometrical meta-metamodel, to prove the concept let's summarize several *M3s*, useful for CPS design. Table 3 shows comparison of graph-based, vector-based and geometry-based meta-metamodels.

Level	Alphabet	Grammar	Operations
M4	Elements <i>d</i> of the set <i>D</i>	The rules of grammar, based on the relations $d \in D$, $\{d\} \subseteq D$	Create / delete element <i>d</i> and subset { <i>d</i> }
	Node $n \in Node$ and edge $e \in Edge$ of graph $G = (Node, Edge)$, Node, Edge $\subset D$	Connection of nodes by edges $e_k(n_i, n_j), n_i, n_j \in Node, e_k \in Edge, i, j = 1 Node , i \neq j, k = 1 Edge $	Add edge $G'=G + e$ Delete edge $G'=G - e$ Add node $G' = G + n$ Delete node $G' = G - n$
	Vector $((x_1, y_1, z_1), (x_2, y_2, z_2))$	Connection of vectors	Summation of vectors, parallel transfer etc.
М3	Point (the value of function $f(x_1,, x_n)$ and its derivatives (up to a fixed order) in the given points), line (traces of the function $f(x_1,, x_n)$ and its derivatives (up to a fixed order) on the specified lines), surface (traces of the function $f(x_1,, x_n)$ and its derivatives (up to a fixed order) for a given surface of dimension $m(0 \le m \le n-1)$, 3D region	equality $A=B$ separateness $A \cap B = \emptyset$ strict separateness $A \cap B = \emptyset \land A' \cap B' = \emptyset$ inside $A \subseteq B \lor B \subseteq A$ strict inside $(A \subseteq B \lor B \subseteq A) \land A' \cap B' = \emptyset$ touchiness $A \cap B = \emptyset \land A' \cap B' = \emptyset$ where A , B are the geometrical objects, and A', B' are the boundaries of these objects.	O_p - interpolation of functions of one or more variables $n \ (n \ge 1) \ O_L$ - interlination of functions of two or more variables $n \ (n \ge 2) \ O_F$ - interflatation of functions of three or more variables $n \ (n \ge 3)$

Table 3.Definition of M4 and samples of the meta-metamodels can be produced

Alphabet of vector-based meta-metamodel is produced from M4 as a having algebraic structure set of geometrical points in the physical space. This M3contains the metatype vector, built as ordered pair of two points. Each point is an ordered set of three coordinates (x, y, z). Applications of this M3 for CPS design is broad, let us mention for example modelling kinematical and dynamical parameters of mechanical movement. The limited size of the paper does not allow us to give the full definition of the meta-metamodels, but this is definitely will become the topic of our future papers.

5. Implementation of the Approach

The software tool that implements the proposed metamodeling approach is currently under development by the authors. This tool is the expansion of the Visual Environment for Cyber-Physical Modeling, which has been considered in^{18,19}. Architecture of software tools was considered in²⁰. The principal novelty is the implementation of the level M4, allowing designers to develop different meta-metamodels. All of the levels of the proposed architecture have user interfaces, which enables designers to express the syntax and semantics of metamodels and apply them for modelling domains.

6. Plans for Future Research

Our future studies will be devoted to the expansion of the proposed metamodeling approach to express the behavioral semantics of CPS. While in the paper we mainly show the possibilities of modeling the structural aspects of CPS, in which metatypes as having algebraic structure subsets were considered, the approach allows us to model other types of mathematical relationships. To express the behavior of CPS, we will present and learn properties of metamodels as special logical and algebraic systems²¹. One of the prospective directions is linking geometrical meta-metamodels with the temporal logic of actions, which will allow us to express spatiotemporal semantics of distributed on geometry actions of CPS.

7. Conclusions

This paper proposes the new metamodeling approach, which allows the development of the metamodels with different types of mathematical semantics (e.g., graphbased, vector-based, geometry-based). This opportunity is possible due to the introduction of the additional level of the metamodeling architecture (M4), which expresses the studied domain in terms of sets. To prove that M4 is mathematically sufficient for producing all considered in the paper meta-metamodels (M3s), we give definitions of M3s in terms of M4, i.e. as having algebraic structure subsets. Given approach allows us to develop the geometrical meta-metamodel Ω for cyberphysical modelling. Structural analysis of physical models showed, that there is a possible alphabet of Ω , which includes corresponding to the dimensions of the space metatypes; this has been a common result of abstraction from a geometrical structure of physical objects. Using the proposed geometrical meta-metamodel Ω allows us to unify the representation of the spatial information at different levels of the CPS design.

8. References

- 1. Catalog of UML Profile Specifications. Available from: http://www.uml.org/#UMLProfiles
- Kelly S, Tolvanen J-P. Domain-Specific Modeling: Enabling Full Code Generation. Wiley-IEEE Computer Society Pr. 2008; 444.
- Gregory G. Nordstrom. Metamodeling rapid design and evolution of domain-specific modeling environments. Dissertation for the degree of Doctor of Philosophy in Electrical Engineering. Nashville, Tennessee, 1999; 170.
- 4. Cyber-Physical Systems. Available from: http:// cyberphysicalsystems.org/
- Lee EA. Cyber-Physical Systems Are Computing Foundations Adequate? NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap. Position Paper. 2006 Oct 16-17.
- 6. Gupta R. Programming models and methods for spatiotemporal actions and reasoning in cyber-physical systems. Workshop on Cyber-Physical Systems, Austin, Texas. 2006.
- ISO/IEC 19502:2005, Information technology. Meta Object Facility (MOF). ANSI. 2007; 292.
- Mezhuyev V. Metamodelling Architecture for Modelling Domains with Different Mathematical Structure. Advanced Computer and Communication Engineering Technology. Lecture Notes in Electrical Engineering, 2015; 315:1049–55.
- Mezhuyev V. Methodology of Domain Specific Mathematical Modelling. Proceedings of the 3rd International Congress on Natural Sciences and Engineering ICNSE 2014, Kyoto, Japan, 2014 May 7-9: 54–64.
- 10. Model-Based Design of cyber-physical systems in MATLAB and Simulink. Available from: http://www.mathworks. com/discovery/cyber-physical-systems.html

- Mezhuyev V, Sputh B, Verhulst E. Interacting Entities Modelling Methodology for Robust Systems Design. Advances in System Testing and Validation Lifecycle. CPS publishing. 2010; 75–80.
- Sputh BHC, Faust O, Verhulst E, Mezhuyev V. OpenComRTOS: A Runtime Environment for Interacting Entities. Communicating Process Architectures. IOS Press. 2009; 173–184.
- 13. Verhulst E, Boute RT, Sampaio Faria JM, Sputh BHC, Mezhuyev V. Formal Development of a Network-Centric RTOS. Springer, 2011; 227.
- 14. Mezhuyev V. Domain specific modelling distributed parallel real time applications. The systems of information processing. 2010; 5(86): 98–103.
- Mezhuyev V, Verhulst E. Modelling OpenComRTOS tasks interaction. Mathematical systems and machines. 2010; 2:32–41.
- 16. Mezhuyev V. Theory and Applications of Domain Specific Mathematical Modelling. Proceedings of the 3rd International Conference on Computer Engineering and Mathematical Sciences ICCEMS 2014, Langkawi, Malaysia. 2014 Dec 4-5. p. 853–74.
- Mezhuyev V, Lytvyn OM, Mohamad Zain J. Metamodel for mathematical modelling surfaces of celestial bodies on the base of radiolocation data. Indian Journal of Science and Technology. 2015 Jul; 8(13):70630.
- Mezhuyev V, Perez-Rodriguez F. Metamodelling approach and software tools for physical modelling and simulation. International Journal of Software Engineering and Computer Sciences (IJSECS). 2015 Feb; 1:1–13.
- Mezhuyev V, Samet R. Geometrical Meta-metamodel for Cyber-Physical Modelling. International Conference Cyberworlds. 2013. p. 89–93.
- Mezhuyev V. Architecture of software tools for Domain-Specific Mathematical Modelling. Proceedings of 2014 International Conference on Computer, Communication, and Control Technology, Langkawi, Malaysia. 2014 Sep 2-4. p. 166–70.
- 21. Mezhuyev V. Development of Metamodels as Logical and Algebraic Systems. Proceedings of the 2014 International Conference on Information Science, Electronics and Electrical Engineering ISEEE, Hokkaido, Japan. 2014 Apr 26-28. p. 1850–5.
- 22. Park J, Lee S, Yoon T. Designing Goal Model for Autonomic Control Point of Cyber-Physical Systems (CPS). Indian Journal of Science and Technology. 2015 Aug; 8(19).