

Formal-semantic normative system for graphic-analytical modelling

S I Matorin¹ and V V Mikhelev¹

¹Belgorod State University, 85, Pobedy St., Belgorod, 308015, Russia

E-mail: matorin@bsu.edu.ru

Abstract. A formal-semantic normative system (FSNS) is presented in this paper. The FSNS allows us to describe the links and functional units of typical information processes. The possibility of using the FSNS to automate graphical-analytical modelling and machine learning is shown. Properties of the FSNS are described. On its basis, a method is developed for specifying a formal alphabet of system connection nodes that has a specific subject content. The rules for manipulating the received alphabetic characters are also presented. The possibility and expediency of building a normative system with increased expressive capabilities is substantiated. This paper provides a brief description of the «Unit-Function-Object» system-object approach. Considered the possibility of using DL for the FSNS in an unambiguous, formalized form. An example of using the FSNS is presented.

1. Introduction

The authors research modern problems of system analysis (presented, for example, in [1, 2]). It led to the creation of a new original method of system-object determinant analysis (SODA), which unlike traditional methods and techniques is a formalized procedure that takes into account the systemic effect and some system-wide patterns [3]. SODA includes three stages.

First, the identification of the class to which the analyzed system belongs, which is carried out when constructing a taxonomic (generic) classification of the subject area. This allows a researcher to determine the functional request of a higher-order system (supersystem) for a system with a given function, i.e. (in terms of the system-object approach) the external determinant of the system.

Secondly, following the stages of formation or creation of a system, which is carried out when constructing a genetic (stadial) classification of a selected class of systems. This allows to specify the requirements for the system and to unambiguously determine its actual functionality arising under the influence of an external determinant, i.e. (in terms of the system-object approach) the internal determinant of the system.

Third, the decomposition of the requirements for the system as a phenomenon, which is carried out when constructing a partitive (whole-part) classification of the system or its meronomy. This gives an idea of the ways to ensure the correspondence of the subsystems of the analyzed or developed system to its internal determinant (and, in the limit, external), i.e. about the ways of functioning or construction of the system.

The last stage, in fact, is a graphic-analytical modeling of the analyzed system. To automate the construction of a graphic-analytical model the idea of creating a formal-semantic normative system (FSNS) is proposed [4]. It is assumed that the FSNS should be based on the alphabet, which consists of symbols that are unambiguously interpreted according to formal characteristics. Moreover, all the used symbols have subject (problem)-oriented semantics. To assign such semantics to the symbols of



the alphabet it is proposed to apply the basic classification of links of the system-object approach [5]. In this classification, the abstract class «*Link (L)*» is divided into disjoint subclasses «*Material link (m)*» and «*Information link (i)*»; the class of material links is divided into disjoint subclasses «*Material link (v)*» and «*Energy link (e)*»; class of information links - into disjoint subclasses «*Data link (d)*» and «*Control link (c)*». At the same time, it is possible to continue and refine this classification depending on the simulated subject area. The alphabetic symbols of the FSNS themselves represent the crossroads of these links, i.e. nodes that are structural elements of the modeled system.

This paper presents an option for constructing the FSNS using the classification of links focused on information processes. The apparatus of descriptive logic (DL) is used for a formal description of the alphabet. DL is a logical language for describing concepts and relationships between them using individuals, concepts and roles.

2. Building a formal-semantic normative system

We use the descriptive logic ALCOIQ(D) presented in [3, 6]. The syntax of this DL is summarized as the following expression:

$$\{\top, \perp, A, A \sqsubseteq C, \neg C, C \sqcap D, C \sqcup D, \exists R.C, \forall R.C, \geq n R.C, \{a\}, \exists [u_1, \dots, u_n].P\}$$

Here symbols \top and \perp are concepts (called truth and false). A is an atomic concept; C, D are any concepts; R is an atomic role. $\{a\}$ is called nominal from [3] and means an individual used as a concept; $\exists [u_1, \dots, u_n].P$ is a description of the concept of a particular area; P is a predicate symbol; u_1, \dots, u_n is a set of attributes. The DL also uses concepts: TBox is a set of terminological axioms; RBox is a set of axioms for roles and their relationships; ABox is a set of axioms for individuals and their relationships. The aggregate of previously mentioned sets of axiom defines the subject area $K = \text{TBox} \cup \text{RBox} \cup \text{ABox}$.

The systems-classes (conceptual systems on Akoff or external systems on Schrader) can be presented in the following form as part of the system-object approach:

$$S_{i,j}^l = S_{i-1,l}^p \sqcap \exists RS_{i,j}^l, \quad (1)$$

where $S_{i,j}^l, S_{i-1,l}^p, RS_{i,j}^l$ – abstract classes (concepts), where $i = \overline{0, N}$, i – hierarchy tier number; j – node number within one tier of the hierarchy; l, p – the parent node numbers for the current node in the hierarchy tier.

A system that is a specific system-class (consisting of instances, not subclasses), in terms of DL is described as a set of three component elements "Unit-Function-Object" (UFO-elements) [5]:

$$S = [L? \sqcup L!; L! \sqcap \exists RS.L?; OS? \sqcup OS! \sqcup OSf], \quad (2)$$

where $L? \sqcup L!$ is a node that is the intersection of a set of inputs and a set of outputs; $L! \sqcap \exists RS.L?$ is a function, which transforms the set of inputs to the set of outputs; $OS? \sqcup OS! \sqcup OSf$ is an object, implements the function and describes the substantial characteristics.

On the other hand, systems-phenomena (material systems according to Ackoff or internal systems according to Schrader), by means of DL (concepts of nominal value and specific areas), can be represented in the following form [5]:

$$s = [\{L?\} \sqcup \{L!\}; \{L!\} \sqcap \exists \text{hasRelation}.\{L?\}; OS? =_{n1} \sqcup OS! =_{n2} \sqcup OSf =_{n3}]. \quad (3)$$

The normative system considered in this work is based on the alphabet of links, which is formed by expanding the basic classification of links. At the same time, the types of data links and control links are introduced into the basic classification. In the notation accepted in DL this extended classification of links can be represented as follows: $\mathbf{m} \sqsubseteq \mathbf{L}, \mathbf{i} \sqsubseteq \mathbf{L}, \mathbf{v} \sqsubseteq \mathbf{m}, \mathbf{e} \sqsubseteq \mathbf{m}, \mathbf{d} \sqsubseteq \mathbf{i}, \mathbf{c} \sqsubseteq \mathbf{i}, \mathbf{dd} \sqsubseteq \mathbf{d}, \mathbf{dp} \sqsubseteq \mathbf{d}, \mathbf{cd} \sqsubseteq \mathbf{c}, \mathbf{cp} \sqsubseteq \mathbf{c}$. Where \mathbf{L} is a whole set of links/flows, \mathbf{m} is a set of material links/flows, \mathbf{i} is a set of information links/flows, \mathbf{v} is a set of substantial links/flows, \mathbf{e} is a set of energy links/flows, \mathbf{d} is a set of links/flows of data, \mathbf{c} is a set of control links/flows, \mathbf{dd} is a set of links/flows of declarative data, \mathbf{dp} is a set of links/flows of procedural data, \mathbf{cd} is a set of links/flows of data management, \mathbf{cc} is a set of links/flows of process management. This assumes the possibility of further dividing the presented links into subsets, if necessary.

With the help of the alphabet of links the alphabet of nodes is described as intersections of alphabetic links in accordance with expression (1). The presented rules for constructing the alphabet of the nodes of the proposed FSNS using the above-mentioned classification of links provide the subject (problem) orientation of the proposed alphabet. It actually makes this normative system formal-semantic and expandable/adaptable depending on the subject area (Table 1).

At the next we have an example of using the proposed FSNS for graphic-analytical modeling of information processes using the example of an ATM serving a client.

Table 1. Rules for constructing the alphabet of the FSNS nodes.

Symbol	Formal expression	Interpretation
V	$\equiv \mathbf{v!} \sqcap \exists \text{hasRelation.}(\mathbf{v?})$	Transformation of matter
E	$\equiv \mathbf{e!} \sqcap \exists \text{hasRelation.}(\mathbf{e?})$	Transformation of energy
D	$\equiv \mathbf{d!} \sqcap \exists \text{hasRelation.}(\mathbf{d?})$	Transformation of data
C	$\equiv \mathbf{c!} \sqcap \exists \text{hasRelation.}(\mathbf{c?})$	Transformation of data
VE	$\equiv (\mathbf{v!} \sqcup \mathbf{e!}) \sqcap \exists \text{hasRelation.}(\mathbf{v?} \sqcup \mathbf{e?})$	Transformation of matter and energy
VD	$\equiv (\mathbf{v!} \sqcup \mathbf{d!}) \sqcap \exists \text{hasRelation.}(\mathbf{v?} \sqcup \mathbf{d?})$	Transformation of matter and data
ED	$\equiv (\mathbf{e!} \sqcup \mathbf{d!}) \sqcap \exists \text{hasRelation.}(\mathbf{e?} \sqcup \mathbf{d?})$	Transformation of energy and data
EC	$\equiv (\mathbf{e!} \sqcup \mathbf{c!}) \sqcap \exists \text{hasRelation.}(\mathbf{e?} \sqcup \mathbf{c?})$	Transformation of energy and control flow
DC	$\equiv (\mathbf{d!} \sqcup \mathbf{c!}) \sqcap \exists \text{hasRelation.}(\mathbf{d?} \sqcup \mathbf{c?})$	Transformation of data and control flow
DD	$\equiv \mathbf{dd!} \sqcap \exists \text{hasRelation.}(\mathbf{dd?})$	Transformation of declarative data
DP	$\equiv (\mathbf{d!} \sqcup \mathbf{p!}) \sqcap \exists \text{hasRelation.}(\mathbf{d?} \sqcup \mathbf{p?})$	Transforming of procedural data
CD	$\equiv (\mathbf{c!} \sqcup \mathbf{d!}) \sqcap \exists \text{hasRelation.}(\mathbf{c?} \sqcup \mathbf{d?})$	Transforming of data control flow
CP	$\equiv (\mathbf{c!} \sqcup \mathbf{p!}) \sqcap \exists \text{hasRelation.}(\mathbf{c?} \sqcup \mathbf{p?})$	Transforming of process control flow
DDDP	$\equiv (\mathbf{dd!} \sqcup \mathbf{dp!}) \sqcap \exists \text{hasRelation.}(\mathbf{dd?} \sqcup \mathbf{dp?})$	Transforming of declarative data and procedural data
DDCD	$\equiv (\mathbf{dd!} \sqcup \mathbf{cd!}) \sqcap \exists \text{hasRelation.}(\mathbf{dd?} \sqcup \mathbf{cd?})$	Transforming of declarative data and data control flow
DPCD	$\equiv (\mathbf{dp!} \sqcup \mathbf{cd!}) \sqcap \exists \text{hasRelation.}(\mathbf{dp?} \sqcup \mathbf{cd?})$	Transforming of procedural data and data control flow
DDCP	$\equiv (\mathbf{dd!} \sqcup \mathbf{cp!}) \sqcap \exists \text{hasRelation.}(\mathbf{dd?} \sqcup \mathbf{cp?})$	Transforming of declarative data and process control flow
CDCP	$\equiv (\mathbf{cd!} \sqcup \mathbf{cp!}) \sqcap \exists \text{hasRelation.}(\mathbf{cd?} \sqcup \mathbf{cp?})$	Transforming of data control flow and process control flow

3. An example of using the formal-semantic normative system

The example shows that even a not very big alphabet of the FSNS makes it easy to interpret, for example, the main processes when issuing money to a client.

At the same time, Figure 1 shows an extended classification of links. It makes possible to form an alphabet of nodes necessary for modelling the selected processes in an ATM; Figure 2 is a context diagram of the ATM service process; Figure 3 is a diagram of the decomposition of information processes (excluding the work of mechanics).

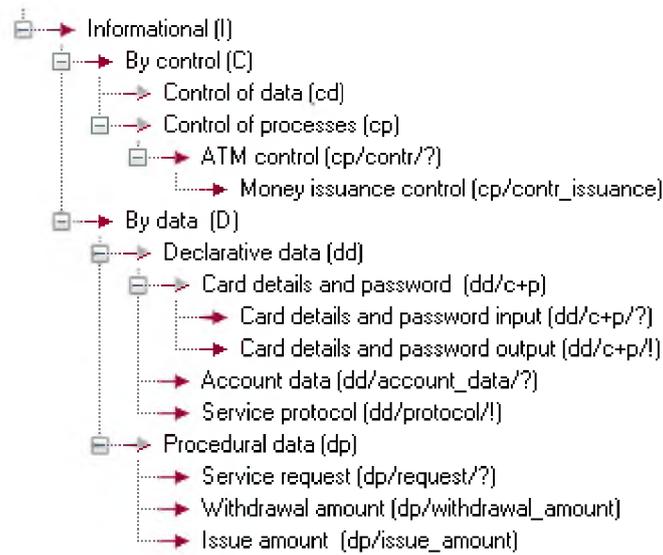


Figure 1. Classification of links for the formation of the alphabet of the nodes of the service process using an ATM

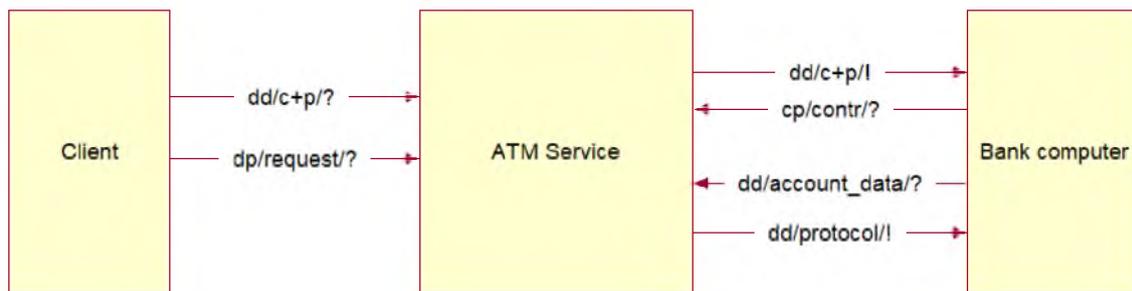


Figure 2. A contextual diagram of the ATM service process

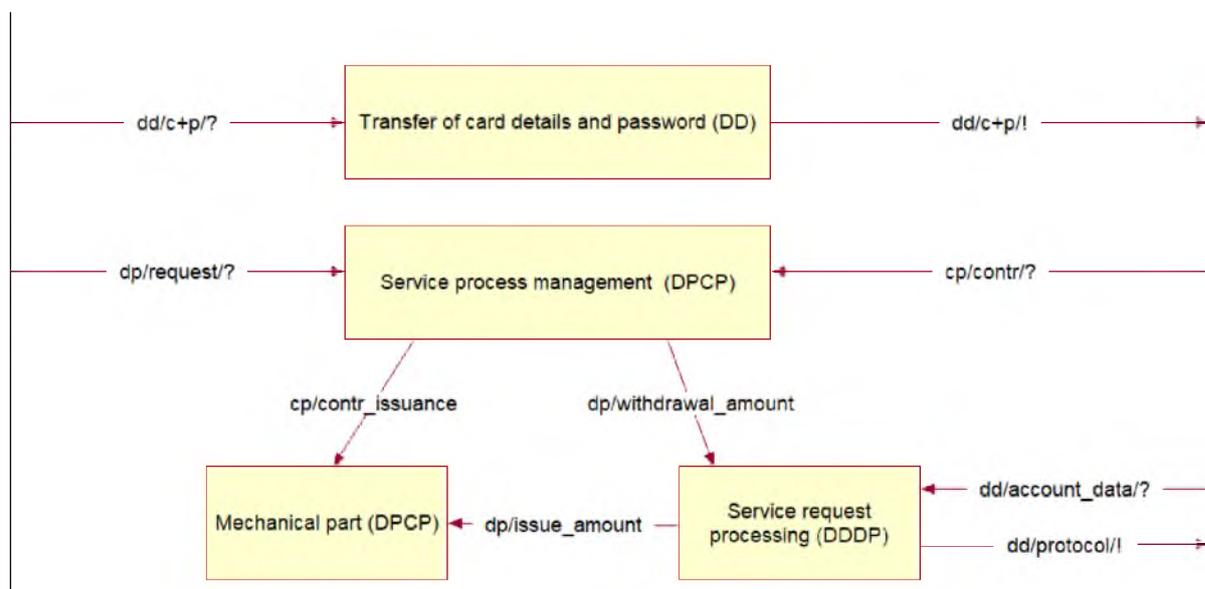


Figure 3. A decomposition diagram of the service process using an ATM, built using the formal semantic alphabet

4. Conclusion

The paper discusses the ability of creating the FSNS. The use of the FSNS of the information system (IS) allows the system itself to offer options for decomposition diagrams for the user and automates the construction of graphic-analytical models. Thus, the FSNS uses alphabetical nodes depending on the knowledge stored in the system. The use of a formal-semantic alphabet (i.e. FSNS) provides an imitation of understanding the IS of properties of classes and instances, using it to build models. Thus, such a system can be viewed as a knowledge-based system. At the same time, the possibility of using rational models allows teaching IS to successful graph-analytical modelling.

References

- [1] Kachala V V 2017 General systems theory and systems analysis *Hot line, Telecom* 431
- [2] Surmin Yu P 2003 Systems theory and systems analysis *MAUP* 368
- [3] Matorin S I and Mikhelev V V 2020 System-Object Approach to the Determinant Analysis of Complex Systems *Artificial Intelligence and Decision Making* **2** 86-93
- [4] Matorin S I, Mikhelev V V, Zhikharev A G 2020 Normative system of system-object analysis and modeling *Economics. Computer science* **3** 623-637
- [5] Matorin S I, Zhikharev A G, Zimovets O A and others 2021 *Systems theory and system analysis KNORUS* 456.
- [6] Baader F, Calvanese D, McGuinness L, Nardi D and Patel-Schneider P F 2003 The Description logic handbook: theory, implementation, and applications *Cambridge University Press* 576