ISSN 0147-6882, Scientific and Technical Information Processing, 2022, Vol. 49, No. 5, pp. 325–332. © Allerton Press, Inc., 2022.
Russian Text © The Author(s), 2021, published in Iskusstvennyi Intellekt i Prinyatie Reshenii, 2021, No. 1, pp. 15–24.

System-Object Determinant Analysis: Constructing a Domain Taxonomy

S. I. Matorin[a], * and V. V. Mikhelev[b]
[a] *ZAO SoftConnect, Belgorod, 308013 Russia*
[b] *Belgorod State University, Belgorod, 308015 Russia*
**e-mail: matorin@softconnect.ru*
Received January 28, 2021

Abstract—The article describes the first stage of a system-object determinant analysis of systems that involves the construction of a taxonomy of a given domain. System classes are analyzed using the system-object approach "Node-Function-Object" and formalization tools of the description logic ALCHOIQ. The connections between the procedure of taxonomy construction and system-wide patterns are considered. An example of a constructed taxonomy is given.

Keywords: system-object determinant analysis, domain taxonomy, system-object approach "Node-Function-Object", description logic, conceptual systems

DOI: 10.3103/S0147688222050069

INTRODUCTION

The undeniable successes of analytical research under the banner of system analysis have proven its usefulness and effectiveness for solving complex practical problems. However, at present "there is no unambiguous understanding of systems analysis itself" [1, p. 231]. There are several definitions of systems analysis and several methodologies of its implementation that aggregate various principles, approaches, and techniques, but these barely include any systems analytics proper [1].

In the professional literature on systems theory and systems analysis, it is noted that "there are no established technologies of systems analysis in research practice" [2]. It is not even possible to unambiguously classify the existing technologies [3]. An example of a fairly detailed, in our opinion, review of the technologies of systems analysis is presented in [4, 5]. We see the reasons for the numerous inconsistent technologies of systems analysis that do not include systems analytics proper in the specific and as yet unresolved methodological problems of the traditional systems approach and systems analysis. These problems include [6]:

—conflation of the concepts of "system" and "sets" and, consequently, a lack of understanding of a system as an integral functional object;

—formalization of the concepts of systems approaches and systems analyses that does not adjust for their specific content determined by system-wide patterns;

—the absence of means of analysis and synthesis of objects as systems, not sets, in the traditional systems approach;

—the lack of an option to apply the concept of class to the implementation of procedures and construction of models, which makes it impossible to analyze conceptual systems.

To solve the above problems, article [7] proposes a new and original means of systems analysis based on the system-object approach — the conceptual apparatus of system-object determinant analysis (SODA), which provides the researcher or designer with effective and versatile tools for analyzing and designing complex, poorly formalizable systems.

SODA consists of three stages. The first stage involves identifying the class to which the analyzed or designed system belongs by constructing a *taxonomic* (*subsumption*) *classification* of the domain. This enables the researcher or developer to determine the *external determinant of the system*, i.e., the functional requirement of a higher-order system (suprasystem) for a system with a given function. The second stage involves defining the stages of the formation or creation of the system by constructing a *genetic* (*stage*) *classification* of the considered class of systems. This makes it possible to specify the system requirements on the one hand and to unambiguously determine the *internal determinant of the system*, i.e., its actual functionality emerging in response to the external determinant, on the other hand. The third stage is decomposition of the requirements for the system as a phenomenon by constructing a *partitive* (*part–whole*) *classification* of the system, or its *meronomy*. This provides an understanding of the ways in which the analyzed or developed system keeps its subsystems aligned with its internal determinant (at the limit—with the external determinant as well), i.e., the mechanisms of the system's functioning or construction.

We consider the system-object determinant analysis proposed in [7] to be potentially useful for the analysis and design of poorly formalizable organizational, information, and technical systems. This article presents a detailed description of the first SODA stage—the procedure for constructing a taxonomy of a given domain. The means of formalized description of systems are introduced. A procedure for constructing a taxonomy (a subsumption classification, or a conceptual system) of a domain is developed. The connections of the developed procedure with system-wide patterns is also considered. An example of a constructed taxonomy is given.

1. MEANS OF FORMALIZATION FOR A DOMAIN TAXONOMY

Within the system-object approach, the most important issue in describing a system and the stages of its analysis or design is *maintaining the functionality of the whole*, both on the part of subsystems towards the system and on the part of the system towards its suprasystem. Incorporating this maintenance into the analysis or design is the defining feature of systems analysis [6, 8].

For *material systems* (internal systems or *systems-phenomena*), that system relation manifests within the part—whole relation in the following way. If a set of streams/connections of parts (subsystems) includes a set of functional streams/connections of the whole (system), then and only then are these parts subsystems. The division into these parts, i.e., the part—whole relation, is systemic, and the streams/connections are also systemic, not formal. It all depends on the degree of functionality of the considered connections of the whole (system)!

For *conceptual systems* (external systems or *systems-classes*), this system relation manifests within the subsumption relation in the following way. If the content of subordinate concepts (subsystems) includes the content/function of the superordinate concept (system), then and only then are these subordinate concepts subsystems. The division into subordinate concepts, i.e., the subsumption relation, is systemic, and the subsumption connections are systemic, not formal. It all depends on the degree of functionality of the considered content of the subsumption system!

Let us consider more closely the procedure for constructing a taxonomy, or a subsumption classification (conceptual system), of a given domain while adjusting for the relation of maintaining the functional ability of the whole, by virtue of which the taxonomy will also correspond to the system-wide principles of communication, hierarchy, monocentrism, and organizational continuity (the definitions of these are given in [6, 9]).

To describe the domain taxonomy and the procedure for its construction, let us first consider the means of their formalization. One such means is description logic (DL) [10]. Description logic is a family of formalisms for representing knowledge, the basic concepts of which are concept, role, and individual; it is characterized by the use of various constructors to create composite concepts from simpler ones. In [7], the stages of determinant analysis are formalized using the DL ALCHOIQ. Its syntax in brief form is as follows:

$$\{\top; \bot; A; A \sqsubseteq C; \neg C; C \sqcap D; C \sqcup D; \exists R.C; \forall R.C; \geq n\, R.C\},$$

where the symbols $\top$ and $\bot$ are the concepts *true* and *false*; A is an atomic concept; C and D are arbitrary concepts; and R is an atomic role.

Study [11] proposes a substantive definition of the concept of system that applies to both systems-phenomena and systems-classes. The formal description of an abstract class (consisting of subclasses, not instances) as a system has the following form:

$$S^i = [S^{i-1}; R.S^{i-1} \sqsupset R.S^i]. \quad (1)$$

In this expression, in accordance with Abadi and Cardelli's object calculus rules: S^{i-1} is a field for indicating a system-class of a higher tier of the hierarchy corresponding to system node S_i; $R.S^i \sqsubset R.S^{i-1}$ is a method that corresponds to the role (function) of system S^i in its suprasystem S^{i-1}.

In [7], the formal definition of an abstract system-class (1) is specified in ALCHOIQ-logic terms as the following concept:

$$S^l_{i,j} = S^p_{i-1,l} \sqcap \exists RS^l_{i,j}, \quad (2)$$

where $S^l_{i,j}, S^j_{i-1,l}, RS^l_{i,j}$ are abstract classes (concepts; i is the hierarchy tier number $= \overline{0, N}$); j is the node number within one hierarchy tier; and l and p are the numbers of the parent nodes of the node of the current hierarchy tier.

A specific class system (that consists of instances, not subclasses) is described in DL terms as a set of three component elements "Node-Function-Object" (NFO elements):

$$C = [\mathbf{U}; \mathbf{F}; \mathbf{O}] = [(L? \sqcup L!);\ L! \sqcap \exists R.L?;\ OS? \sqcup OS! \sqcup OSf], \quad (3)$$

where $\mathbf{U} = L? \sqcup L!$ is a node as the intersection of a set of inputs $L?$ and outputs $L!$ and $\mathbf{F} = L! \sqcap \exists R.L?$ is a function that converts a set of inputs into a set of outputs.

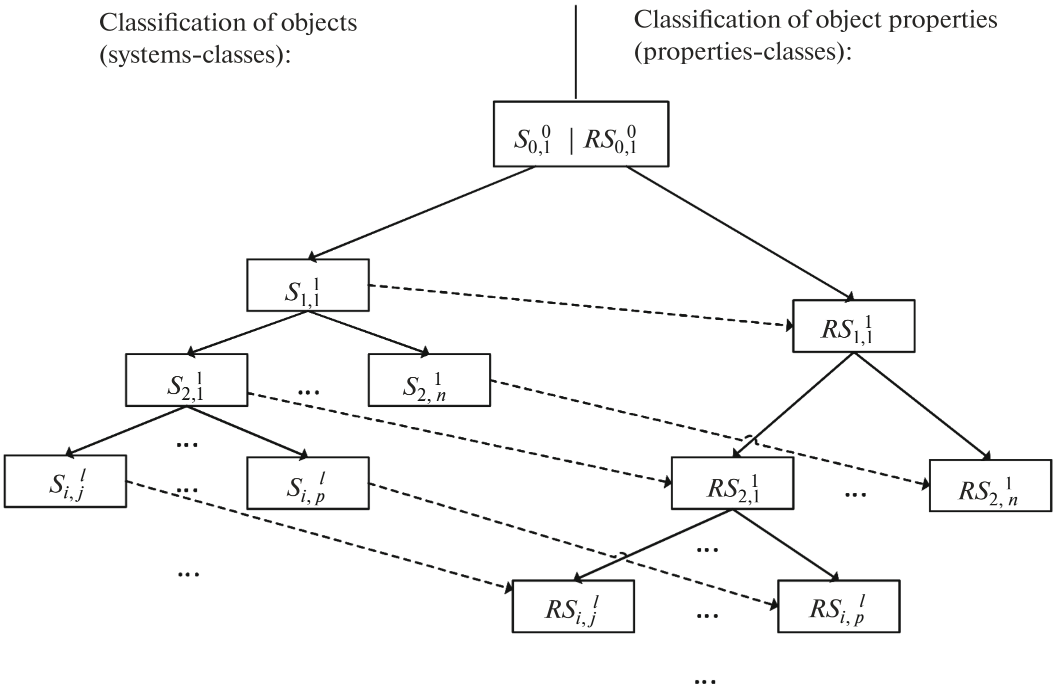

Fig. 1. Conceptual system.

The expression $L! \sqcap \exists R.L?$ shows that the set of outputs $L!$ is "connected" with the set of inputs $L?$ by role R; R is a functional role representing correspondence between concepts. If we introduce the role R – hasCorrespondence, we can specify the definition of the function: $\mathbf{F} = L! \sqcap \exists$ hasCorrespondence. $L?$; $\mathbf{O} = OS? \sqcup OS! \sqcup OSf$ is an object that implements the function and has substantive characteristics (input, output, and internal).

A conceptual system as a whole can be represented in DL terms as $\mathbf{K} = TBox \sqcup RBox$, where $TBox$ is a set of terminological axioms and $RBox$ is a set of axioms for roles and their relations. From the results obtained in [7], it follows that a conceptual system is a combination of $Tbox$ and $RBox$:

$$RBox = \begin{pmatrix} \cdots \\ RS_{i,j}^{l} \sqsubset RS_{i-1,l}^{p} \\ \cdots \end{pmatrix};$$
$$TBox = \begin{pmatrix} \cdots \\ S_{i,j}^{l} \sqsubseteq S_{i-1,l}^{p} \sqcap \exists RS_{i,j}^{l} \\ \cdots \end{pmatrix}. \quad (4)$$

2. CONSTRUCTION OF A DOMAIN TAXONOMY

The procedure for constructing a domain taxonomy consists of three actions.

(1) Selection and formulation of the subsumptive definition of the most basic concept/class of the domain ($S_{1,1}^{1}$). This definition should reflect the functional properties of the object or phenomenon that correspond to the defined concept/class ($S_{1,1}^{1}$) in its differentia ($RS_{1,1}^{1}$). The differentiae should ideally be properties that support the functional properties ($RS_{0,1}^{1}$) of the concept/class ($S_{0,1}^{1}$) that is superordinate in relation to the considered basic concept/class ($S_{1,1}^{1}$) of the domain. Thus, the differentia of the basic concept/class of the domain ($RS_{1,1}^{1}$) must be a type of the differentiae ($RS_{0,1}^{1}$) of the concept/class ($S_{0,1}^{1}$) that is superordinate for the considered concept/class ($S_{1,1}^{1}$). Expression (2) leads to $S_{1,1}^{1} = S_{0,1}^{0} \sqcap \exists RS_{1,1}^{1}$, where $S_{1,1}^{1} \sqsubset S_{0,1}^{1}$ and $RS_{1,1}^{1} \sqsubset RS_{0,1}^{1}$. The formulation of this basic subsumptive definition leads to the self-evident satisfaction of the *principle of monocentrism* [6], as the further constructed hierarchy will have a single (selected at this stage specifically for the given domain) root node (Fig. 1).

This stage is similar to construction of contextual model in graph-analytic modeling of material systems and processes that begins from the representation of the modeled processes as a single contectual one with a common name.

(2) Decomposition of the basic concept/class of the domain into subordinate concepts/classes towards which the general concept/class is superordinate, i.e., $S^1_{2,1},\dots,S^1_{2,n} \sqsubset S^1_{1.1}$. The properties of the subordinate concepts/classes reflected in their differentiae must support the functional properties of the superordinate concept/class. Thus, the differentiae of concepts/classes must be types of differentiae of the superordinate concept/class. I.e., $RS^1_{2,1},\dots,RS^1_{2,n} \sqsubset RS^1_{1.1}$. Expression (2) leads to $S^1_{2,j} = S^1_{1,1} \sqcap \exists RS^1_{2,j}$, where $S^1_{2,j} \sqsubset S^1_{1,1}$ and $RS^1_{2,j} \sqsubset RS^1_{1,1}$.

This stage corresponds to the decomposition stage in the graph-analytic modeling of material systems and processes, which involves determining the parts (subprocesses) that support the functioning of the contextual process. From the substantive definition of the system-class [6, 11] and the formal representation of the system-class (1) the satisfaction of the *principles of communication* (the system is connected with its environment by numerous communications) and *hierarchy* (at any level of the hierarchy the system is part of a higher-tier system, i.e., a suprasystem) follow [6]. The satisfaction of the principle of hierarchy reinforces the satisfaction of the *principle of monocentrism*.

An expanded understanding of the principle of monocentrism leads to the satisfaction of Bogdanov's *principle of organizational continuity*, which is the idea that between any two systems there are connections that link them together into a single "chain of ingression" [6], which we have proven for systems-phenomena [9] and for systems-classes [11].

3. Repeat step 2 for each subordinate concept/class ($S^1_{2,1},\dots,S^1_{2,n}$). I.e., define first-level systems-classes through higher-level (in this case – second-level) classes, for example, $S^l_{i,j} = S^1_{2,1} \sqcap \exists RS^l_{i,j}$; …; $S^l_{i,p} = S^1_{2,n} \sqcap \exists RS^l_{i,p}$, such that $RS^l_{i,j},\dots,RS^l_{i,p} \sqsubset RS^1_{2,1}$ is true for their properties-classes (Fig. 1).

The procedure for constructing a taxonomy of classes that can be described as abstract systems-classes, for example, of form (1) or (2), ends with a class that represents a specific system-class that can be described using classes-connections (streams), classes-functions (processes), and classes-objects (object properties), i.e., an expression of form (3), emerging at some level of the hierarchy.

The result of the domain classification is a concept/class that includes the analyzed or designed system and specifies the functional properties of this system in a general form. The definition of the class of connections is essentially the same as the definition of the functional requirement for the analyzed or designed system (its *external determinant*), and the definition of object properties specifies the implementation mechanisms of the system's functions.

The detailed procedure of the first stage of system-object determinant analysis (the construction of a domain taxonomy) can be presented in the form of an algorithm, the flowchart for which is given in Figs. 2 and 3, using the terms and designations of descriptive logic introduced above and also in [7]. The conceptual system of the domain "Passenger car" is presented in Fig. 4 in the same form as in Fig. 1.

3. EXAMPLE OF A DOMAIN TAXONOMY

As an example of constructing a domain taxonomy, let us consider the classification of cars.

1.1. *Car* ($S^1_{0,1}$)—a vehicle ($S^1_{-1,1}$) intended for transporting people and goods ($RS^1_{0,1}$).

1.2. *Passenger car* ($S^1_{1,1}$)—a car ($S^1_{0,1}$) intended for transporting passengers and luggage and designed to seat 2 to 8 people ($RS^1_{1,1}$).

1.2.1. *Regular passenger car* ($S^1_{2,1}$)—a PC ($S^1_{1,1}$) for driving on paved roads ($RS^1_{2,1}$).

1.2.1.1. *Sports car* ($S^1_{3,1}$)—an RPC ($S^1_{2,1}$) for highly dynamic driving ($RS^1_{3,1}$).

1.2.1.2. *Executive car* ($S^1_{3,2}$)—an RPC ($S^1_{2,1}$) that provides increased passenger comfort ($RS^1_{3,2}$).

1.2.1.3. *Crossover car* ($S^1_{3,3}$)—an RPC ($S^1_{2,1}$) for paved and unpaved roads ($RS^1_{3,3}$).

1.2.1.3.1. Minicompact ($S^3_{4,1}$), see Table 1;

1.2.1.3.2. Subcompact ($S^3_{4,2}$);

1.2.1.3.3. Compact ($S^3_{4,3}$) 1;

1.2.1.3.4. Mid-size ($S^3_{4,4}$) 1;

1.2.1.3.5. Full-size ($S^3_{4,5}$);

1.2.1.4. *Off-road vehicle* ($S^1_{3,4}$)—a PC ($S^1_{2,1}$) with off-road capability ($RS^1_{3,4}$).

1.2.1.5. *Commercial* ($S^1_{3,5}$)—a PC ($S^1_{2,1}$) intended for transporting small shipments of goods ($RS^1_{3,5}$).

The presented taxonomy (conceptual system) can be described in the form of *RBox* and *TBox*:

$$Box = \begin{bmatrix} RS^1_{1,1} \sqsubset RS^1_{0,1} \\ RS^1_{2,1} \sqsubset RS^1_{1,1} \\ RS^1_{3,1} \sqsubset RS^1_{2,1} \\ RS^1_{3,2} \sqsubset RS^1_{2,1} \\ RS^1_{3,3} \sqsubset RS^1_{2,1} \\ RS^1_{3,4} \sqsubset RS^1_{2,1} \\ RS^1_{3,5} \sqsubset RS^1_{2,1} \end{bmatrix};$$

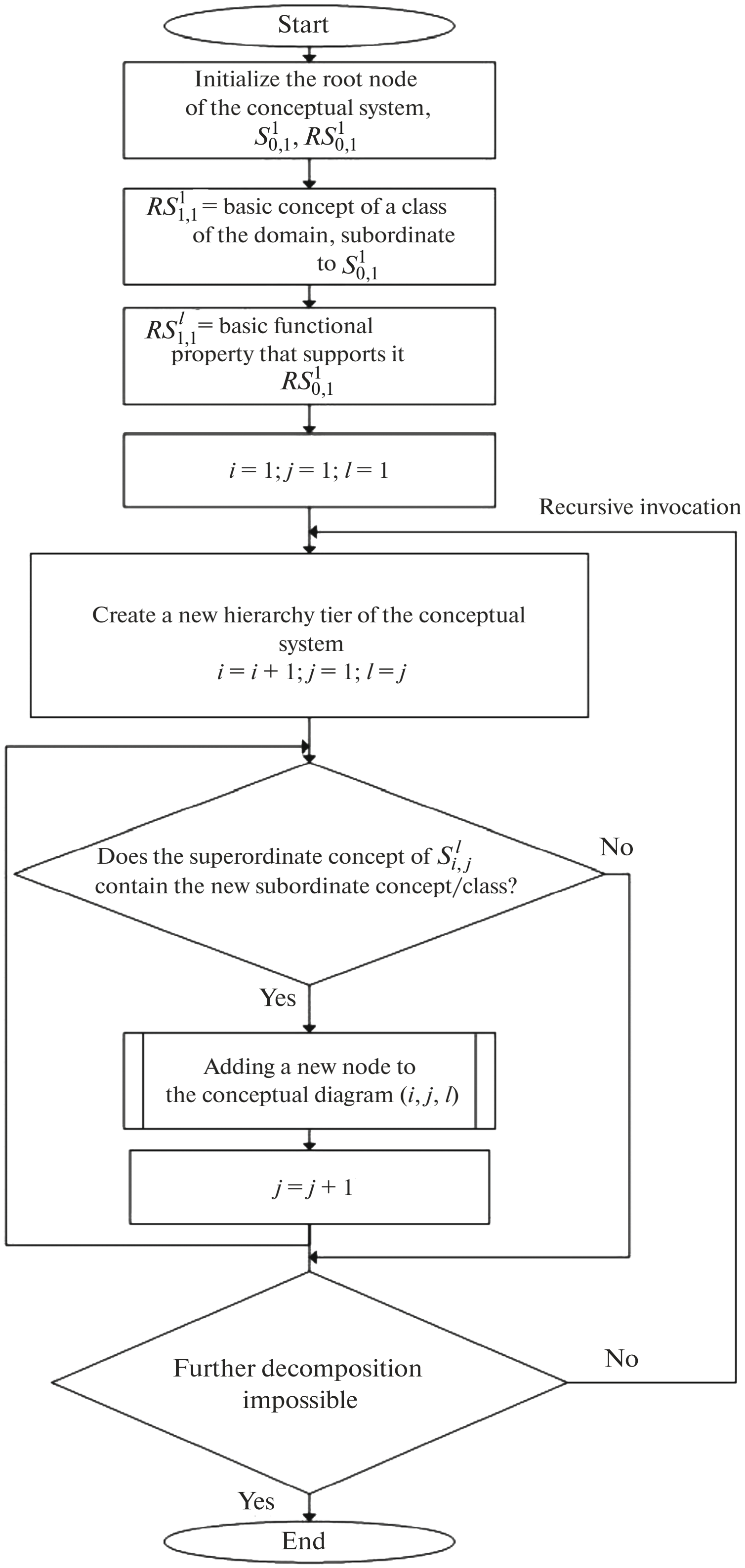

Fig. 2. Flowchart of the algorithm for constructing a conceptual system.

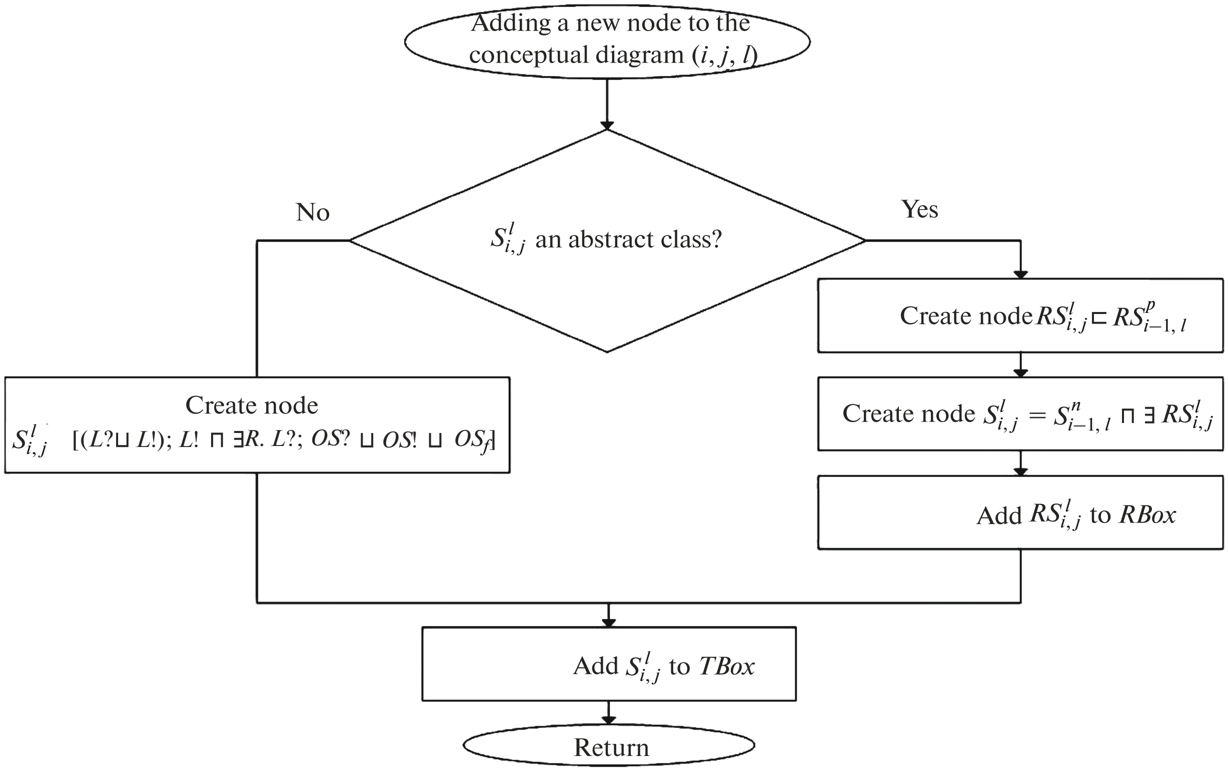

Fig. 3. Flowchart of the subprogram for adding a new node to the conceptual system.

$$TBox = \begin{bmatrix} S_{0,1}^{1} \sqsubseteq S_{-1,1}^{1} \sqcap \exists RS_{0,1}^{1} \\ S_{1,1}^{1} \sqsubseteq S_{0,1}^{1} \sqcap \exists RS_{1,1}^{1} \\ S_{2,1}^{1} \sqsubseteq S_{1,1}^{1} \sqcap \exists RS_{2,1}^{1} \\ S_{3,1}^{1} \sqsubseteq S_{2,1}^{1} \sqcap \exists RS_{3,1}^{1} \\ S_{3,2}^{1} \sqsubseteq S_{2,1}^{1} \sqcap \exists RS_{3,2}^{1} \\ S_{3,3}^{1} \sqsubseteq S_{2,1}^{1} \sqcap \exists RS_{3,3}^{1} \\ S_{3,4}^{1} \sqsubseteq S_{2,1}^{1} \sqcap \exists RS_{3,4}^{1} \\ S_{3,5}^{1} \sqsubseteq S_{2,1}^{1} \sqcap \exists RS_{3,5}^{1} \\ S_{3,5}^{1} \sqsubseteq S_{2,1}^{1} \sqcap \exists RS_{3,5}^{1} \\ S_{4,1}^{3} \sqsubseteq S_{3,3}^{1} \\ S_{4,2}^{3} \sqsubseteq S_{3,3}^{1} \\ S_{4,3}^{3} \sqsubseteq S_{3,3}^{1} \\ S_{4,4}^{3} \sqsubseteq S_{3,3}^{1} \\ S_{4,5}^{3} \sqsubseteq S_{3,3}^{1} \end{bmatrix};$$

Then, for example, a specific system-class $S_{4,2}^{3}$—“subcompact crossover”—can be formally described using expression (3):

$$S_{4,2}^{3} = [(L?_{4,2}^{3} \sqcup L!_{4,2}^{3});\ L!_{4,2}^{3} \sqcap \exists RS_{4,2}^{3}.L?_{4,2}^{3};\ OS?_{4,2}^{3} \sqcup OS!_{4,2}^{3} \sqcup OSf_{4,2}^{3}], \qquad (5)$$

where $L?_{4,2}^{3}$ = passengers ⊔ luggage ⊔ consumables ⊔ control actions.

$L!_{4,2}^{3}$ = passengers ⊔ luggage ⊔ torque ⊔ replaceable parts ⊔ effects of control actions.

$RS_{4,2}^{3}$ = torque control ⊔ boarding, disembarking, seating passengers and storing luggage ⊔ controlling the movement and various elements of the car ⊔ conducting the MOT test.

$OS_{4,2}^{3} = OS?_{4,2}^{3} \sqcup OS!_{4,2}^{3} \sqcup OSf_{4,2}^{3}$ = dimensions of the input and output elements ⊔ capacity for storing materials and objects ⊔ engine capacity and horsepower ⊔ weight and size characteristics of the car as a whole.

Thus, by constructing a taxonomy (conceptual system) of a domain, we can formulate a set of structural, functional, and substantive properties of the analyzed or designed system.

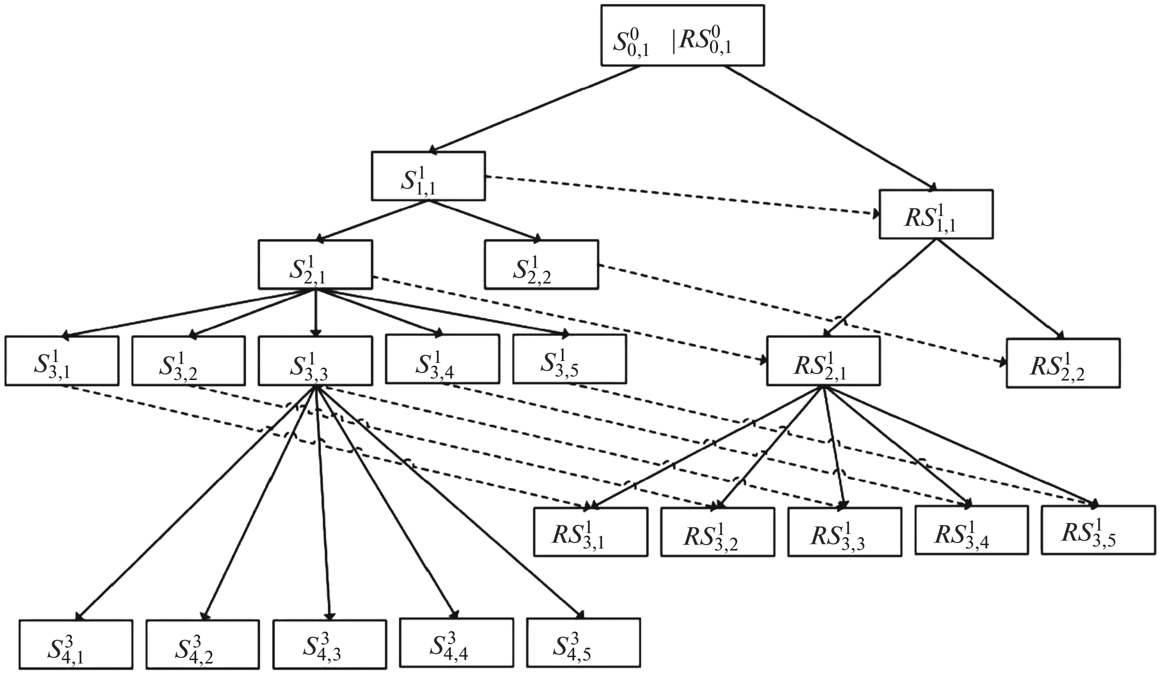

Fig. 4. Conceptual diagram of the domain Car.

CONCLUSIONS

The described procedure of the subsumption (taxonomic) classification of a domain can be used to determine the class to which the analyzed or designed system belongs and thereby also determine its external determinant, i.e., the functional requirement of a higher-order system (suprasystem) for a system with a given function. This definition of a concept/class that includes the analyzed or designed system incorporates its functional properties, namely, the classes of input and output connections that define the suprasystem's functional requirement to the analyzed or designed system, while object characteristics of the system specify implementation mechanisms of its functions. The formulation of a set of requirements for a specific class of such systems essentially formalizes the process of creating a technical specification for the development of a new technical or information system.

In the future, we plan to present the remaining stages of system-object determinant analysis, which include the construction of a genetic (stage) classification of a given class of systems to determine the internal determinant of the system, i.e., its actual functionality, and the construction of a partitive (part–whole)

Table 1. Subordinate concepts of the class Crossovers and their object characteristics

SUV class	Length, mm	Width, mm	Width, mm	Engine capacity, L	Horsepower, h.p.
(A) Minicompact ($S^3_{4,1}$)	Under 4241	Under 1765	Under 2591	Under 1.6	Under 120
(B) Subcompact ($S^3_{4,2}$)	4241–4340	1765–1813	2519–2674	1.6–2.0	120–150
(C) Compact ($S^3_{4,3}$)	4382–4695	1805–1855	2638–2727	1.6–2.5	150–200
(D) Mid-size ($S^3_{4,4}$)	4680–4898	1855–1925	2745–2824	2.4–3.5	170–250
(E) Full-size ($S^3_{4,5}$)	Over 4980	Over 1961	Over 2900	2.4–3.5 and over	200–350 and over

classification of the system to determine the ways in which the system functions or is constructed.

FUNDING

The study was supported by the Russian Foundation for Basic Research, project nos. 19-07-00290a, 19-07-00111a, 19-29-01047мк.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Kachala, V.V., *Obshhaja teorija sistem i sistemnyj analiz* (General Systems Theory and Systems Analysis), Moscow: Gorjachaja Linija—Telekom, 2017.
2. Surmin, Yu.P., *Teoriya sistem i sistemnyy analiz* (Systems Theory and Systems Analysis), Kiev: MAUP, 2003.
3. Spitsnadel', V.N., *Osnovy sistemnogo analiza* (Fundamentals of Systems Analysis), St. Petersburg: Biznes-Pressa, 2000.
4. *Teoriya sistem i sistemnyj analiz. Spravochnik* (Systems Theory and Systems Analysis: Reference Book), Volkova, V.N. and Emel'yanov, A.A., Eds., Moscow: Finansy i Statistika, 2006.
5. Volkova, V.N. and Denisov, A.A., *Teoriya sistem i sistemnyi analiz* (Systems Theory and Systems Analysis), Moscow: Yurait, 2015.
6. *Teoriya system i sistemnyi analiz. Uchebnik* (Systems Systems Theory and Systems Analysis), Matorin, S.I., Zhikharev, A.G., Zimovets, O.A., , Eds., Moscow: Direktmedia Pablishing, 2020. http://biblioc1ub.ru/index.php?pagwbook&id=574641.
7. Matorin, S.I. and Mikhelev, V.V., A system—object approach to determinant analysis of complex systems, *Sci. Tech. Inf. Process.*, 2021, vol. 48, no. 5, pp. 327—332. https://doi.org/10.3103/S014768822105004X
8. Mel'nikov, G.P., *Sistemologiya i yazykovye aspekty kibernetiki* (Systemology and Language Aspects of Cybernetics), Moscow: Sovetskoe Radio, 1978.
9. Matorin, S.I. and Zhikharev, A.G., Accounting for system-wide regularities in system—object modeling of organizational knowledge, *Sci. Tech. Inf. Process.*, 2018, vol. 46, no. 6, pp. 388—396. https://doi.org/10.3103/S0147688219060042
10. Schmidt-Schauß, M. and Smolka, G., Attributive concept descriptions with complements, *Artif. Intell.*, 1991, vol. 48, no. 1, pp. 1—26. https://doi.org/10.1016/0004-3702(91)90078-X
11. Matorin, S.I., Zhikharev, A.G., and Mikhelev, V.V., Accounting for system-wide patterns of conceptual systems in modeling of conceptual knowledge, *Sci. Tech. Inf. Process.*, 2020, vol. 47, no. 5, pp. 314—321. https://doi.org/10.3103/S0147688220050020

Translated by A. Ovchinnikova