ISSN 0147-6882, Scientific and Technical Information Processing, 2023, Vol. 50, No. 6, pp. 595–601. © Allerton Press, Inc., 2023.
Russian Text © The Author(s), 2022, published in Iskusstvennyi Intellekt i Prinyatie Reshenii, 2022, No. 1, pp. 26–34.

System-Object Determinant Analysis. Construction of Genetic and Partitive Classifications of the Subject Area

S. I. Matorin[a, b, *] and V. V. Mikhelev[a]

[a] *National Research University Belgorod State University, Belgorod, Russia*
[b] *Belgorod University of Cooperation, Economics & Law, Belgorod, Russia*
**e-mail: matorin@softconnect.ru*

Received October 20, 2021

Abstract—This paper considers the second and third stages of a system-object determinant analysis, including the construction of the genetic and partitive classifications of the subject area. Class and phenomenon systems are described using the single system-object approach Node–Function–Object and formalization means in the language of descriptive logic ALCHOIQ (D). Partitive classification is obtained using a formal-semantic normative system based on the classification of alphabetical nodes and links. Examples of constructed genetic and partitive classifications are given.

Keywords: system-object determinant analysis, genetic classification, partitive classification, subject domain meronomy, system-object approach Node–Function–Object, descriptive logic, conceptual systems, systems-phenomena, formal-semantic normative system

DOI: 10.3103/S0147688223060102

INTRODUCTION

The existence of methodological problems of traditional systems approach and system analysis that have been unresolved so far and are noted, for example, in works [1–5], led the authors to create a conceptual apparatus of system-object determinant analysis (SODA) [6], taking into account the relationship of maintaining functional ability whole as a main system-wide regularity [7].

SODA consists of three stages. First, the class to which the analyzed or designed system belongs in the course of construction is revealed, the *taxonomic* (*generic*) *classification* subject area. This makes it possible to unambiguously determine the *external determinant of the system*, i.e., a functional request of a higher-order system (supersystem) for a system with a given function. Second, the stages of the formation or creation of the system are traced in the course of building *genetic* (*stadial*) *classification* for a selected class of systems. This allows us, on the one hand, to specify the system requirements, and on the other hand, to unambiguously determine the *internal determinant of the system*, i.e., its actual functionality, arising under the influence of an external determinant. Third, the decomposition of the requirements for the system as a phenomenon (its internal determinant) is carried out in the course of building *partitive* (*whole-private*) *classification* or *meronomy* systems. This provides the analyst or developer with an idea of how to ensure that the subsystems of the system being analyzed or developed conform to its internal determinant, i.e., the ways of functioning or building a system.

The first stage of SODA—the procedure for constructing a taxonomy of a given subject area—is considered in detail in [8]. This article describes the second and third stages of SODA: the procedures for constructing genetic and partitive classifications. The formalization of these procedures and examples of their implementation are presented. The analytical procedures proposed in [6, 8] and in this paper will, according to the authors, be useful in the analysis or design of weakly formalized organizational, informational, and technical systems.

TOOLS FOR FORMALIZING SODA PROCEDURES

Description logic (DL) is used to formalize the SODA procedures *ALCOIQ(D)*, as in [6, 8]. The syntax of this DL in its short form is represented as the following expression:

$$\{\top, \bot, A, A \sqsubseteq c, \neg C, C \sqcap D, C \sqcup D, \exists RC,$$
$$\forall RC, \geq nRC, \{a\}, \exists[u_1, \ldots, u_n].P\},$$

where $\top$ And $\bot$ are concepts (called true and false); A is an atomic concept; C, D are arbitrary concepts; R is an atomic role; $\{a\}$ is a denomination, where individuals enclosed in curly brackets are represented as full-fledged concepts; $\exists[u_1, \ldots, u_n].P$ is the description of the concept of a specific area; P is a predicate symbol; and $u_1, \ldots, u_n$ are many attributes.

The following concepts are used in DL: TBox is a set of terminological axioms; RBox is a set of axioms for roles and their relationships; and ABox is a set of axioms for individuals and their relations, the totality of which determines the subject area K = TBox ∪ rbox ∪ Abox.

For a formalized description of the features of the second stage of SODA, i.e., the process of constructing a genetic (stage) classification, it is necessary to clarify the formal definition of a material system (system-phenomenon) in the triune structure Node–Function–Object (NFO-element). This refinement is based on the results presented in [7, 9–11] and combines the description of the NFO element in the form of a special object of the Abadi-Kardeli object calculus [7, 9] and its description by integrating the algebraic means of the Grenander pattern theory and the calculus of Milner processes (calculus of communication systems) [10, 11]. The application of the DL language to the above descriptions makes it possible to represent the NFO element as a special composite concept of descriptive logic. The result of this refinement is a formal representation of some arbitrarily chosen material system (system-phenomenon) **s** as an NFO element in the form of the following expression:

$$\mathbf{s} = [(\{LS?\} \sqcup \{LS\tau\} \sqcup \{LS!\});$$
$$(\{LS!\} \sqcap \exists f\tau.\{LS\tau\} \sqcap \exists f.\{LS?\});$$
$$(\exists hasOS?_{=\mathbf{k1}} \sqcup \exists hasOS\tau_{=\mathbf{k2}} \sqcup \exists hasOS!_{=\mathbf{k3}})],$$

where $(\{LS?\} \sqcup \{LS\tau\} \sqcup \{LS!\})$ is a concept for describing system node s as an intersection of a finite set of input $\{LS?\}$ and output $\{LS!\}$ connections in the structure of the supersystem, as well as a set of internal connections $\{LS\tau\}$; $(\{LS!\} \sqcap \exists f\tau.\{LS\tau\} \sqcap \exists f.\{LS?\})$ is a concept for describing the function of the system s specified by the supersystem, or a method that provides a functional correspondence between the output $\{LS!\}$ and input $\{LS?\}$ flows of a given node, taking into account the intermediate role (function) $f\tau$ transformations internal flows $\{LS\tau\}$; $(\exists hasOS? = \mathbf{k1} \sqcup \exists hasOS\tau = \mathbf{k2} \sqcup \exists hasOS! = \mathbf{k3})$ is a concept for describing the substantial (objective) characteristics of system s (input, internal, and output); and **k1**, **k2**, and **k3** are attributes with specific values.

As shown in [8], in the first stage of SODA, in the process of constructing a generic classification (taxonomy), a hierarchy of abstract classes (system-class) is determined, including the analysis or design of the system:

$$S^{i,n} = [S^{i,n-1};\ RS^{i,n} \sqsubset RS^{i,n-1}],$$

where $S^{i,n-1}$ is a field for indicating the system-class of a higher tier of the hierarchy; $RS^{i,n} \sqsubset RS^{i,n-1}$ is the field for describing the method corresponding to the role $RS^{i,n}$ (functions) of the system $S^{i,n}$nested in the role $RS^{i,n-1}$ supersystems $S^{i,n-1}$; $\sqsubset$ is the symbol for nesting a concept into a concept or a role into a role in the language of descriptive logic; index i denotes that the given abstract class includes the analyzed system-phenomenon $\mathbf{s}^i$, and index n is the number of the tier in the system-class hierarchy.

The first stage of SODA ends when some class of the lower tier of the hierarchy (system-class $S^{i,n+1}$) can be described using a node class, a function class, and an object class:

$$S^{i,n+1} = [(LS?^{i,n+1} \sqcup LS\tau^{i,n+1} \sqcup LS!^{i,n+1});$$
$$(LS!^{i,n+1} \sqcap \exists R.LS\tau^{i,n+1} \sqcap \exists R.LS?^{i,n+1});$$
$$(OS?^{i,n+1} \sqcup OS\tau^{i,n+1} \sqcup OS!^{i,n+1})]$$
$$= [S^{i,n};\ RS^{i,n+1} \sqsubset RS^{i,n}] \sqsubset S^{i,n},$$

where $(LS?^{i,n+1} \sqcup LS\tau^{i,n+1} \sqcup LS!^{i,n+1})$ is the concept describing the class of nodes as intersections of the class of input connections $LS?^{i,n+1}$ and the class of output connections $LS!^{i,n+1}$ in the structure of the supersystem-class $S^{i,n}$, and the class of internal connections $LS\tau^{i,n+1}$; $(LS!^{i,n+1} \sqcap \exists R.LS\tau^{i,n+1}\ \exists R.LS?^{i+1})$ is a concept that describes the class of functions of the system-class $S^{i,n+1}$, defined by the supersystem-class $S^{i,\ n}$, or a method that provides a functional correspondence between the classes of output $LS!^{i,n+1}$ and input $LS?^{i,n+1}$ flows of this class of nodes, taking into account intermediate transformations of the class of internal flows $LS\tau^{i,n+1}$; $(OS?^{i,n+1} \sqcup OS\tau^{i,n+1}, \sqcup OS!^{i,n+1})$ is a concept for describing classes of substantial (objective) system characteristics (input, output, transfer/internal). Representation of a conceptual system (class system) in the form of a class of nodes, a class of functions, and a class of objects allows us to proceed to the second stage of SODA: the construction of a genetic classification.

GENETIC/STAGE CLASSIFICATION

At the second stage of SODA, the analyzed or designed system-phenomenon is identified $\mathbf{s}^i$ in the process of building the genetic or staging classification system class $S^{i,n+1}$. At the first step of the stage, the classes of input and output connections are specified; at the second, the class of functions is given; and at the third, the classes of object characteristics are provided. These steps can be represented as follows.

Concretization of classes of input, internal and output links of a system-class $S^{i,n+1}$ in DL language using an extension called *denomination*:

$$S^{i,n+2} = [(\{LS?^i\} \sqcup \{LS\tau^i\} \sqcup \{LS!^i\});$$
$$(\{LS!^i\} \sqcap \exists R.\{LS\tau^i\} \sqcap \exists R.\{LS?^i\});$$
$$(OS?^{i,n+1} \sqcup OS\tau^{i,n+1} \sqcup OS!^{i,n+1})] \sqsubset S^{i,n+1}.$$

In [12] it is shown that individuals enclosed in curly brackets are full-fledged concepts (the "nominal" concept). $\{LS?^i\} \sqsubset LS?^{i,n+1}$, $\{LS\tau^i\} \sqsubset LS\tau^{i,n+1}$, $\{LS!^i\} \sqsubset LS!^{i,n+1}$

are specific input, internal, and output flows/connections, respectively.

The concretization of the class of functions of the system-class $S^{i,n+2}$ in DL language, taking into account that the f and $f\tau$-role implements the transformation of input links into output ones:

$$S^{i,n+3} = [(\{LS?^i\} \sqcup \{LS\tau^i\} \sqcup \{LS!^i\});$$
$$(\{LS!^i\} \sqcap \exists f\tau.\{LS\tau^i\} \sqcap \exists f.\{LS?^i\});$$
$$(OS?^{i,n+1} \sqcup OS\tau^{i,n+1} \sqcup OS!^{i,n+1})] \sqsubset S^{i,n+2}.$$

Respectively, $(\{LS!^i\} \sqcap \exists f\tau.\{LS\tau^i\} \sqcap \exists f.\{LS?^i\}) \sqsubset (\{LS!^i\} \sqcap \exists R.\{LS\tau^i\} \sqcap \exists R.\{LS?^i\})$.

The concretization of classes of object characteristics of system-class $S^{i,n+3}$ specific attributes in the DL language using an extension called the *specific area*, reveals the system-phenomenon

$$\mathbf{s}^i = [(\{LS?^i\} \sqcup \{LS\tau^i\} \sqcup \{LS!^i\});$$
$$(\{LS!^i\} \sqcap \exists f\tau.\{LS\tau^i\} \sqcap \exists f.\{LS?^i\}) \sqsubset (\{LS!^i\};$$
$$(\exists hasOS?^i = \mathbf{p1} \sqcup \exists hasOS\tau^i$$
$$= \mathbf{p2} \sqcup \exists hasOS!^i = \mathbf{p3})] \sqsubset S^{i,n+3}.$$

Thus, $hasOS?^i$, $hasOS\tau^i$, $hasOS!^i$ are specific attributes (object characteristics), with specific values **p1**, **p2**, and **p3**.

The direction or criteria for concretization result from the fact that the system-phenomenon $\mathbf{s}^i$ (as part of a higher order system) must regard the maintenance of functional ability the whole, as in relation to system-class $S^{i,n+1}$ (as an element of the class), and in relation to the supersystem-phenomenon $\mathbf{s}^{i-1}$ (as its subsystem):

$$\mathbf{s}^{i-1} = [(\{LS?^{i-1}\} \sqcup \{LS\tau^{i-1}\} \sqcup \{LS!^{i-1}\});$$
$$(\{LS!^{i-1}\} \sqcap \exists f\tau.\{LS\tau^{i-1}\} \sqcap \exists f.\{LS?^{i-1}\});$$
$$(\exists hasOS?^{i-1} = \mathbf{k1} \sqcup \exists hasOS\tau^{i-1}$$
$$= \mathbf{k2} \sqcup \exists hasOS!^{i-1} = \mathbf{k3})]; \quad \mathbf{s}^i \sqsubset \mathbf{s}^{i-1}.$$

such that $\mathbf{s}^i$ regards the maintenance of functional ability of the whole to $S^{i,n+1}$ and $\mathbf{s}^{i-1}$; the following conditions must be met.

First,

$$\{LS?^{i-1}\} \sqcup \{LS\tau^{i-1}\} \Rightarrow \{LS\tau^{i-1}\}$$
$$\Rightarrow (\{LS?^i\} \sqcup \{LS\tau^i\}) \sqcap (LS!^{i-1} \sqcap \exists f.\{LS?^{i-1}\})$$
$$\Rightarrow \exists f\tau.\{LS\tau^{i-1}\} \Rightarrow (LS!^i \sqcap \exists f.\{LS?^i\} \sqcap (\exists hasOS?^{i-1}$$
$$= \mathbf{t1} \sqcup \exists hasOS!^{i-1} = \mathbf{t3}) \Rightarrow \exists hasOS\tau^{i-1} = \mathbf{t2}$$
$$\Rightarrow (\exists hasOS?^i = \mathbf{k1} \sqcup \exists hasOS!^i = \mathbf{k3}).$$

These conditions establish that to maintain the functional ability of the supersystem-phenomenon $\mathbf{s}^{i-1}$ and the supersystems-class $S^{i,n+1}$ from the system $\mathbf{s}^i$, all of the properties of the latter must be determined (conditioned) by the properties of the said supersystems.

Second,

$$(\{LS?^i\} \sqcup \{LS!^i\}) \sqsubset \{LS\tau^{i-1}\} \sqcap (\{LS!^i\} \sqcap \exists f.\{LS?^i\})$$
$$\sqsubset \exists f\tau.\{LS\tau^{i-1}\} \sqcap (\exists hasOS?^i =$$
$$\mathbf{k1} \sqcup \exists hasOS!^i = \mathbf{k3}) \sqsubset \exists hasOSt^{i-1} = \mathbf{k2}.$$

These conditions establish the mechanism by which the functional ability of the supersystem-phenomenon is maintained $\mathbf{s}^{i-1}$ from the system-phenomenon $\mathbf{s}^i$.

After the system $\mathbf{s}^i$ is defined in accordance with the mentioned conditions, the third stage of SODA begins: the construction of a *partitive classification* (meronomies) of this system by its decomposition using a special alphabet of model elements proposed in [7] and refined in [13].

PARTITIVE CLASSIFICATION (MERONOMY)

The partitive classification stage is an analogue of the graphical-analytical modeling of the system that is being analyzed or designed, and it allows us to decompose the system into subsystems in accordance with its internal determinant. We introduce a formal semantic normative system (FSNS) based on the alphabet of links, formed by expanding the basic classification of links [7]. The types of data links and the control links are added to the basic classification (in this case, taking into account the example presented below). In the notation adopted in the DL, this extended classification of bonds can be represented as follows:

$$\mathbf{m} \sqsubset \mathbf{L},\ \mathbf{i} \sqsubset \mathbf{L},\ \mathbf{v} \sqsubset \mathbf{m},\ \mathbf{e} \sqsubset \mathbf{m},\ \mathbf{d} \sqsubset \mathbf{i},\ \mathbf{c} \sqsubset \mathbf{i},$$
$$\mathbf{dd} \sqsubset \mathbf{d},\ \mathbf{dp} \sqsubset \mathbf{d},\ \mathbf{cd} \sqsubset \mathbf{c},\ \mathbf{cp} \sqsubset \mathbf{c},$$

where **L** is a set of links/flows; **m** is a set of material connections/flows; **i** is a set of information links/flows; **v** is a set of real connections/flows; **e** is a set of energy connections/flows; **d** is a set of connections/data flows; **c** is a set of links/flows of control; **dd** is a set of links/flows of declarative data; **dp** is a set of connections/flows of procedural data; **cd** is a set of links/flows of data management; and **cp** is a set of links/flows of process control.

This assumes the possibility of further division of all types of relationships into subspecies if necessary.

The alphabet of nodes as intersections of alphabetic links is described using the alphabet of links. The rules for constructing the alphabet of FSNS nodes (Table 1) provide the subject (problem) orientation of the proposed alphabet, which makes this normative system formal-semantic and expandable/adaptable depending on the subject area.

Thus, the construction of a partitive classification (meronomy) of the system $\mathbf{s}^i$ can be reduced to the following steps:

Table 1. Rules for constructing the alphabet of FSNS nodes

Sign	Formal expression	Interpretation
V	$\equiv \mathbf{v}! \sqcap \exists f.(\mathbf{v}?)$	Substance transformation
E	$\equiv \mathbf{e}! \sqcap \exists f.(\mathbf{e}?)$	Energy conversion
D	$\equiv \mathbf{d}! \sqcap \exists f.(\mathbf{d}?)$	Data conversion
C	$\equiv \mathbf{c}! \sqcap \exists f.(\mathbf{c}?)$	Control flow transformation
VE	$\equiv (\mathbf{v}! \sqcup \mathbf{e}!) \sqcap \exists f.(\mathbf{v}? \sqcup \mathbf{e}?)$	Transformation of matter and energy
VD	$\equiv (\mathbf{v}! \sqcup \mathbf{d}!) \sqcap \exists f.(\mathbf{v}? \sqcup \mathbf{d}?)$	Substance and data transformation
ED	$\equiv (\mathbf{e}! \sqcup \mathbf{d}!) \sqcap \exists f.(\mathbf{e}? \sqcup \mathbf{d}?)$	Energy and data conversion
EU	$\equiv (\mathbf{e}! \sqcup \mathbf{c}!) \sqcap \exists f.(\mathbf{e}? \sqcup \mathbf{c}?)$	Energy conversion and control flow
DC	$\equiv (\mathbf{d}! \sqcup \mathbf{c}!) \sqcap \exists f.(\mathbf{d}? \sqcup \mathbf{c}?)$	Data transformation and control flow
DD	$\equiv \mathbf{dd}! \sqcap \exists f.(\mathbf{dd}?)$	Converting declarative data
D.P.	$\equiv \mathbf{dp}! \sqcap \exists f.(\mathbf{dp}?)$	Procedural data transformation
CD	$\equiv \mathbf{cd}! \sqcap \exists f.(\mathbf{cd}?)$	Data flow transformation
CP	$\equiv \mathbf{cp}! \sqcap \exists f.(\mathbf{cp}?)$	Process flow transformation
DDDP	$\equiv (\mathbf{dd}! \sqcup \mathbf{dp}!) \sqcap \exists f.(\mathbf{dd}? \sqcup \mathbf{dp}?)$	Converting declarative data and procedural data
DDCD	$\equiv (\mathbf{dd}! \sqcup \mathbf{cd}!) \sqcap \exists f.(\mathbf{dd}? \sqcup \mathbf{cd}?)$	Declarative data transformation and data control flow
DPCD	$\equiv (\mathbf{dp}! \sqcup \mathbf{cd}!) \sqcap \exists f.(\mathbf{dp}? \sqcup \mathbf{cd}?)$	Transformation of procedural data and data control flow
DDCP	$\equiv (\mathbf{dd}! \sqcup \mathbf{cp}!) \sqcap \exists f.(\mathbf{dd}? \sqcup \mathbf{cp}?)$	Declarative data transformation and process control flow
DPCP	$\equiv (\mathbf{dp}! \sqcup \mathbf{c}!) \sqcap \exists f.(\mathbf{dd}? \sqcup \mathbf{cp}?)$	Transformation of procedural data and process control flow
CDCP	$\equiv (\mathbf{cd}! \sqcup \mathbf{cp}!) \sqcap \exists f.(\mathbf{cd}? \sqcup \mathbf{cp}?)$	Transformation of data control flow and process control flow

(1) Let $\mathbf{s}^i = [(\{LS?^i\} \sqcup \{LS\tau^i\} \sqcup \{LS!^i\}); (\{LS!^i\} \sqcap f.\{LS\tau^i\} \sqcap \exists f.\{LS?^i\}); (\exists hasOS?^i_{=}\ \mathbf{p1} \sqcup \exists hasOS\tau^i = \mathbf{p2} \sqcup \exists hasOS!^i = \mathbf{p3})]$ is the system-phenomenon to be analyzed or designed. System $\mathbf{s}^i$ is decomposed into subsystems, the functional properties of which are supporting for the system $\mathbf{s}^i$, described as follows: $\mathbf{s}^{i,\,m} \sqsubset \mathbf{s}^i$ $(m = 1, N)$; or $\mathbf{s}^i \equiv \mathbf{s}^{i,\,1} \sqcup \ldots \sqcup \mathrm{s}^{i,N}$, such that: $(\{LS!^{i,m}\} \sqcap \exists f\tau.\{LS\tau^{i,m}\} \sqcap \exists f.\{LS?^{i,m}\}) \sqsubset \exists f\tau.\{LS\tau^i\}$.

(2) For each subsystem $\mathbf{s}^{i,m} = [(\{LS?^{i,m}\} \sqcup \{LS\tau^{i,m}\} \sqcup \{LS!^{i,m}\}); (\{LS!^{i,m}\} \sqcap \exists f.\{LS\tau^{i,m}\} \sqcap \exists f.\{LS?^{i,\,m}\}); (\exists hasOS?^{i,\,m} = \mathbf{q1} \sqcup \exists hasOS\tau^{i,m} = \mathbf{q2} \sqcup \exists hasOS!^{i,\,m} = \mathbf{q3})]$ we determine the types of links and nodes based on the normative system introduced earlier.

(3) Repeat step 2 for all subsystems of systems-phenomena $\mathbf{s}^{i,\,m}$.

EXAMPLES OF BUILDING GENETIC AND PARTITIVE CLASSIFICATIONS

Consider the functioning of the SODA algorithm, using the example of designing an information accounting system including several subsystems, including a document routing and storage system (DRSS).

At the first stage of SODA, a generic classification of information systems (IS) is built according to the types of data used (Fig. 1). The stage ends with the following representation of the accounting system in the form of classes of NFO elements:

$$\mathrm{DRSS}^{1.1} = [(LS?^{i,n+1} \sqcup LSt^{i,n+1} \sqcup LS!^{i,n+1});$$
$$(LS!^{\,i,n+1} \sqcap \exists R.LS\tau^{i,n+1} \sqcap \exists R.LS?^{i,n+1});$$
$$(OS?^{i,n+1} \sqcup OS\tau^{i,n+1} OS\tau^{i,n+1} \sqcup OS!^{i,n+1})] \sqsubset \mathrm{DRSS}.$$

At the second stage of SODA, the following sequential concretization of node classes, function classes, and object characteristics classes is conducted.

Accounting $\mathrm{DRSS}^{1.1}$ = [(User impact)? ⊔ (Ordered information)! ⊔ (Various kinds of data)$\boldsymbol{\tau}$; (Ordered information)! ⊓ **∃has a match.**(Various types of data)$\boldsymbol{\tau}$ ⊓ **∃has a match.**(Invoices)?; (Properties of the input data)? ⊔ (Internal data properties)$\boldsymbol{\tau}$ ⊔ (Output data properties)!].

Accounting $\mathrm{DRSS}^{1.2}$ = [{User search info}? ⊔ {Invoice report}! ⊔ {Internal data}$\boldsymbol{\tau}$; {Invoice report}! ⊓ **∃has a match.**{Internal data}$\boldsymbol{\tau}$ ⊓ **∃has a match.**{Search user information}?; (Properties of the input data)? ⊔ (Internal data properties)$\boldsymbol{\tau}$ ⊔ (Output data properties)!].

Accounting $\mathrm{DRSS}^{1.3}$ = [{User search info}? ⊔ {Invoice report}! ⊔ {Internal data}$\boldsymbol{\tau}$; {Invoice report}! ⊓ **∃data processing.** {Internal data}$\boldsymbol{\tau}$ ⊓ **∃search query processing.**{Search user information}?; (Properties of the input data)? ⊔ (Internal data properties)$\boldsymbol{\tau}$ ⊔ (Output data properties)!].

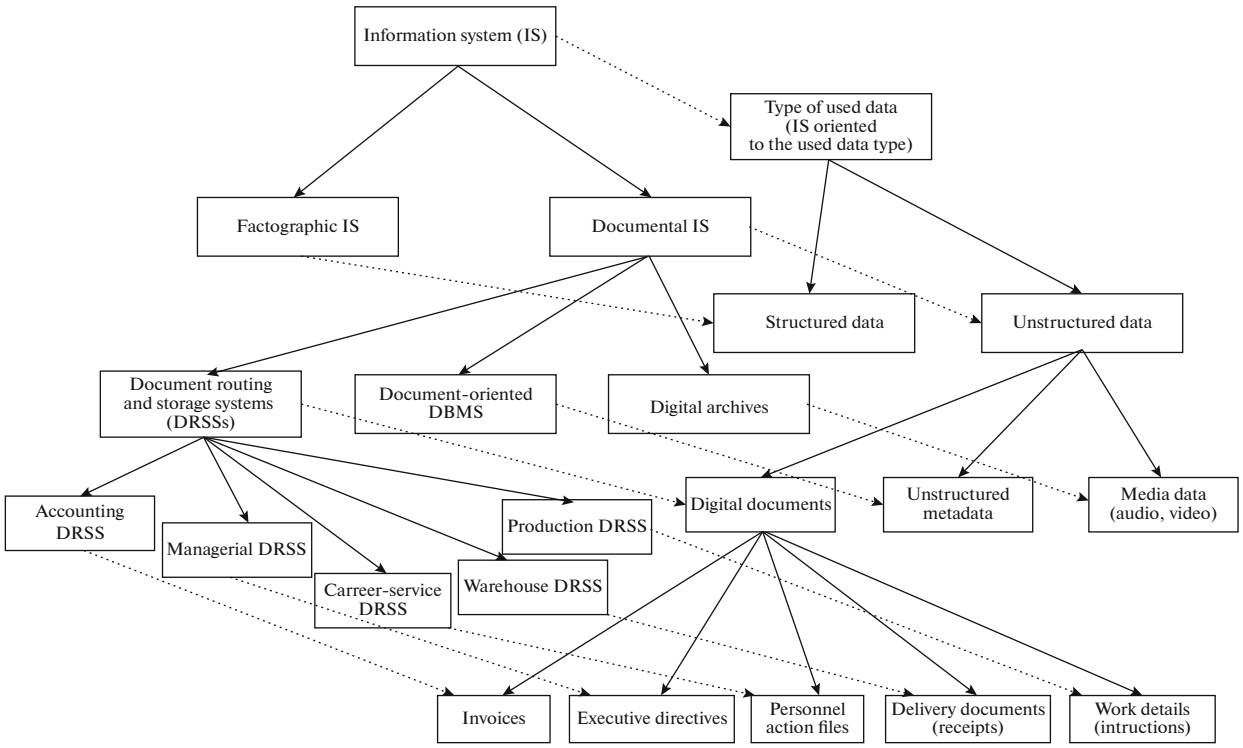

Fig. 1. Classification (taxonomy) of IP depending on the type of data used.

Accounting DRSS = [{User search information}? ⊔ {Invoice report}! ⊔ {Internal data}**τ**; (Ordered information)! ⊓ **∃data processing.**{Internal data}**τ** ⊓ **∃search query processing.**{Search user information}?; ((Max items) = 20 ⊔ (VAT amount ≤ 10,000 rubles))? ⊔ ((Number of concurrent Users) ≥ 100 ⊔ (system response time ≤ 2 sec))**τ** ⊔ ((Maximum number of pages) ≤10)!].

After determining the system-phenomenon Accounting DRSS, the third stage of SODA is performed the construction of a partitive classification. Subsystems for the system-phenomenon are initially determined (Fig. 2).

Using description logic tools, it is possible to define TBox and ABox for the subject area Accounting DRSS.

Tbox consists of the following axioms:

- Accounting DRSS$^{1.1}$ ⊏ DRSS;
- Accounting DRSS$^{1.2}$ ⊏ Accounting DRSS$^{1.1}$;
- Accounting DRSS$^{1.3}$ ⊏ Accounting DRSS$^{1.2}$;
- Accounting DRSS ⊏ Accounting DRSS$^{1.3}$.

ABox consists of the following axioms:

- Interface ⊏ Accounting DRSS;
- Search module ⊏ Accounting DRSS;
- Export module ⊏ Accounting DRSS;
- Accounting DRSS ≡ Interface ⊔ Search module ⊔ Export module;
- Interface ≡ [({User input} ⊔ {Request to use})? ⊔ {Internal Interface Links}**τ** ⊔ ({User output} ⊔ {search request})!; ({User output} ⊔ {search request})! ⊓ **∃processing the User Request.** ({User input} ⊔ {Request to use})? ⊓ **∃processing the User Request.** {Internal Interface Links}**τ**; (Login password length = 5 characters)? ⊔ (Number of clicks on links ≤ 3)**τ** ⊔ (Number of objects to search ≤ 20)!];
- Search module ≡ [({Database data} ⊔ {search request})? ⊔ {Data filter}**τ** ⊔ ({Database output} ⊔ {Output search request})!; ({Database output} ⊔ {Output search query})! ⊓ **∃Processing request.** ({DB data} ⊔ {search request})? ⊓ **∃applyingFilter.** {Data filter}**τ**; (Number of objects to search.≤20)? ⊔ (Number of filter options. ≤ 3)**τ** ⊔ (Response time. ≤ 3 sec.)!];

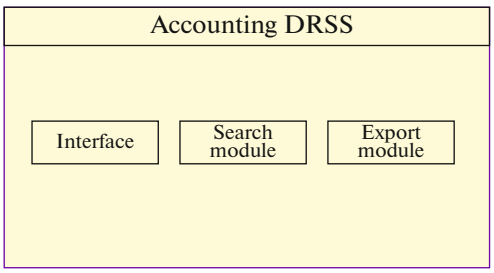

Fig. 2. Subsystem of the decomposition diagram Accounting DRSS.

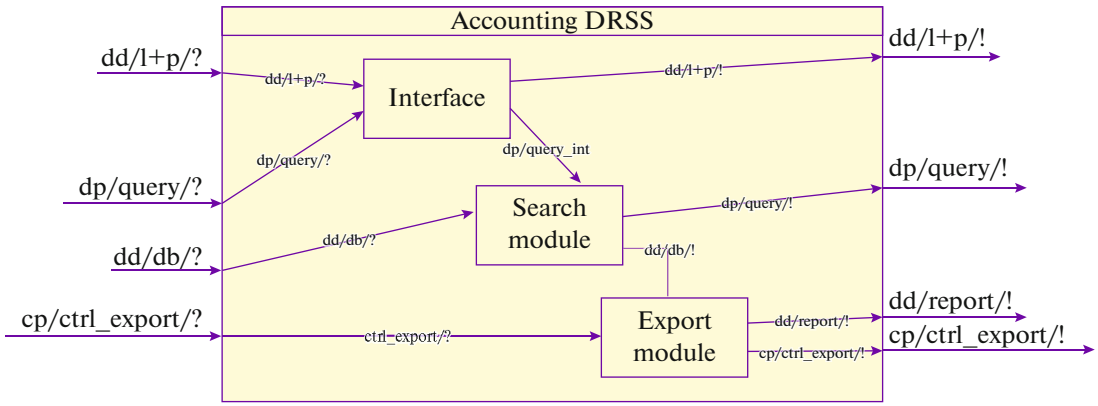

Fig. 3. Extended decomposition diagram of subsystems Accounting DRSS.

• Export module ≡ [({Filtered database data} ⊔ {Export management})? ⊔ {Export format}**τ** ({Invoice report} ⊔ {output export control})!; ({Invoice report} ⊔ {output export control})! ⊓ **∃Performance port.** ({Filtered DB data} ⊔ {Export management})? ⊓ **∃saving the Document.**{Export format}**τ**; (Number of processed objects. ≤ 20)? ⊔ (VAT amount ≤ 10000 rub.)**τ** ⊔ (Working hours. ≤ 3 sec.)!].

Further, the interaction of the data of the subsystem is represented using the introduced FSNS. The necessary alphabetic nodes and links are determined using the FSNS classification based on the nature of the input and output links of the subsystems. Thus (see table) Interface, **DDDP**; Search module, **DDDP**; and Export module, **DDCP**. Types of connections and nodes and their interactions, performed in the NFO-toolkit package, are shown in Figs. 3 and 4.

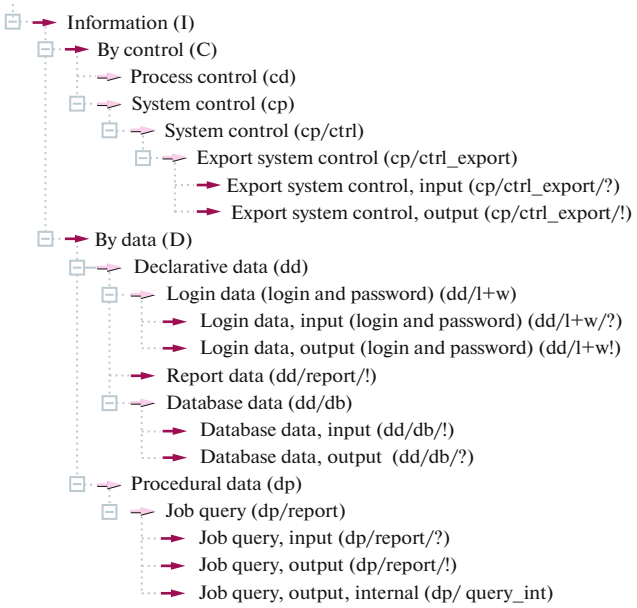

Fig. 4. Hierarchy of alphabetical links of the subsystem Accounting DRSS.

CONCLUSIONS

The results presented in this paper complete the description of the research and development of SODA. According to the authors, SODA compensates for some of the shortcomings of traditional systems analysis tools because it is performed in accordance with formalized algorithmic procedures that take into account a number of system-wide patterns.

At the first stage of SODA [8], the class to which the analyzed or designed system belongs is determined, and thereby, the external determinant of the system is determined, i.e., its functional request of a higher order (supersystem) for a system with a given function. At the same time, a class is defined that includes the system being analyzed or designed, the functional properties of this system are fixed, and the classes of input and output connections, which set a functional request to the system from the side of the supersystem, and its object characteristics clarify the ways of implementing functions.

The subsequent stages of SODA are discussed in this article. The generic procedure of classifying the classification of the subject area allows a consistent refinement of system-classes to be performed for specific systems-phenomena, the properties of which should be determined by the properties of the supersystem. The formulation of a set of requirements for systems in the course of generic and genetic classification formalizes the process of developing terms of reference for the creation of a new technical or information system, which determines the functionality of the system, arising from its internal determinants. The partitive classification stage, together with the use of a formal-semantic normative system, implements the decomposition of the analyzed system into subsystems, which formalizes the procedure for designing a new system.

Further research involves the use of the SODA procedure for the analysis and design of systems in specific subject areas.

FUNDING

The work is supported by the Russian Foundation for Basic Research, project nos. 19-07-00290a, 19-07-00111a, and 19-29-01047mk.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. Kachala, V.V., *Obshchaya teoriya sistem i sistemnyi analiz* (General Systems Theory and Systems Analysis), Moscow: Goryachaya Liniya Telekom, 2017.
2. Surmin, Yu.P., *Teoriya sistem i sistemnyi analiz* (System Analysis in Science and Education), Kiev: MAUP, 2003.
3. Spitsnadel', V.N., *Osnovy sistemnogo analiza* (Fundamentals of Systems Analysis), St. Petersburg: Biznes-Pressa, 2000.
4. *Teoriya sistem i sistemnyi analiz v upravlenii organizatsiyami. Spravochnik* (Systems Theory and Systems Analysis in Management of Enterprises: 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. 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
7. Zhikharev, A.G., Zimovets, O.A., Tubol'tsev, M.F., and Kondratenko, A.A., *Teoriya sistem i sistemnyi analiz.* Uchebnik, 2021.
8. Matorin, S.I. and Mikhelev, V.V., System–object determinant analysis: Constructing a domain taxonomy, *Sci. Tech. Inf. Process.*, 2022, vol. 49, no. 5, pp. 325–332. https://doi.org/10.3103/S0147688222050069
9. Matorin, S.I., Zhikharev, A.G., and Zimovets, O.A., Object calculus in the system–object method of knowledge representation, *Sci. Tech. Inf. Process.*, 2018, vol. 45, no. 5, pp. 307–316. https://doi.org/10.3103/S0147688218050039
10. Zimovets, O.A. and Matorin, S.I., Integration of formalization tools for graphical-analytical "unit–function–object models, *Sci. Tech. Inf. Process.*, 2013, vol. 40, no. 6, pp. 396–402. https://doi.org/10.3103/S0147688213060099
11. Zimovets, O. and Matorin, S., *Sistemnoe grafoanaliticheskoe modelirovanie administrativnykh protsedur* (Systematic Graph-Analytic Modeling of Administrative Procedures), Belgorod: OOO GiK, 2014.
12. Baader, F., Calvanese, D., McGuinness, L., Nardi, D., and Patel-Schneider, P., *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge Univ. Press, 2003. https://doi.org/10.1017/cbo9780511711787
13. Mikhelev, V.V., Matorin, S.I., and Zhikharev, A.G., Normative system of system–object analysis and modeling, *Ekon. Inf.*, 2020, vol. 47, no. 3, pp. 623–637. https://doi.org/10.18413/2687-0932-2020-47-3-623-637