DOI: http://doi.org/10.25728/cs.2022.5.4

APPLICATION OF SEMIOTIC MODELS TO DECISION-MAKING

A. A. Kulinich

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ alexkul@rambler.ru

Abstract. This paper introduces an approach to building decision support systems based on a semiotic domain model and natural language processing methods. The knowledge base of this model is a text corpus of linguistic information obtained from the Internet. The text corpus is relevant to the subject domain in which the subjective semiotic model of the situation is constructed. A method for solving the inverse problem in a semiotic system is proposed. The obtained solutions are interpreted in the subject domain using a semantic calculator. The semantic calculator extracts generic relations from the text corpus based on lexico-syntactic patterns and determines the frequency of joint occurrence of words in the solution based on the distributive analysis of the text corpus. The generalized structures of monitoring and decision-making subsystems with the semiotic model of the situation and natural language processing methods are described. A software layout of the decision-making subsystem is developed. The effectiveness of this approach is demonstrated by experiments.

Keywords: decision-making, semiotic system, subjective model, natural language processing, distributive analysis.

INTRODUCTION

Currently, decision support methods in complex socio-economic and political systems under uncertainty can be classified as follows. The first class of methods (Data Mining) obtains general trends of a subject domain by extracting knowledge from data represented in numerical scales.

Another class of methods directly extracts expert knowledge (the best solution in choice models or situation forecasting models according to experts' views). These methods use the subjective preferences of experts, their assessments and knowledge of the general trends of the subject domain, etc. However, in this case, there are difficulties in constructing a mathematical model of the object and measuring its parameters. Under uncertainty, such an expert model conceptually simulates and qualitatively reflects the main trends of the situation. In such conditions, the model of the situation is difficult to verify; therefore, the simulation results are difficult to interpret in terms of the subject domain and are unreliable.

Decision-making methods using linguistic information about a controlled object were investigated within situation management [1]. Here, the natural language description of an object is represented in a restricted natural language through core structures, which include language elements and various relations between them. Such a description is called the object's state, and management is possible if there exists a natural language description of the control action for some target state. In situation management, it is necessary to enumerate all possible states of the controlled object and assign a control action in the natural language to each state. For complex objects, this problem becomes difficult to solve and requires much expert work.

The ideas of situation management were further developed within applied semiotics [2]. Here, the model of an object is constructed using sign-symbols. A sign-symbol was defined by German logician G. Frege as a triplet consisting of a name, a sense, and a sign meaning [3]. A sign symbol relates the knowledge of an expert (name and sense) with an object of the real world (sign meaning).

In [4], a sign was defined as a quadruple: a name, an image (percept), a meaning, and a personal sense. Here, the mathematical model of a sign, the operators of binding all its elements, and the operations on different sets of sign elements were defined. The authors placed emphasis on the recognition of perceptual images in the form of the connected sign picture of the world; this picture determines the behavior of a subject based on its experience (personal meaning).

In [2], the model of a semiotic system known as Pospelov's semiotic square was introduced. Pospelov's semiotic square includes the following elements: a metasign that defines the name of a semiotic system (the set of sign symbols); a syntax that defines the rules of building a sign system; semantics that define the basic properties of the semiotic system; pragmatics that define the basic actions performed within this semiotic system.

The main aspects of semiotic systems (syntax, semantics, and pragmatics) were formulated in the classical works of logicians Ch. Peirce [5] and J. Morris [6].

The semiotic approach has the following applications: information systems design [7]; computer systems design (*computer semiotics*) [8]; system interface representation in different but equivalent sign systems (*algebraic semiotics*) [9]; conceptual modeling in databases [10] (the extended entity-relationship model with frame algebra and data images).

The semiotic approach was adopted to solve complex strategic problems in power engineering and other critical infrastructures; for details, see [11, 12].

This paper considers the construction of semiotic decision support systems under uncertainty. For this purpose, we combine a qualitative semiotic model of the situation and technologies for obtaining relevant information from the Internet with various natural language processing methods. The qualitative subjective semiotic model is used as a pattern for obtaining relevant information from the Internet. The following problems are considered: situation monitoring and forecasting; decision support to manage the situation, including the interpretation of solutions and search for their precedents on the Internet.

1. THE ARCHITECTURE OF A SEMIOTIC DECISION SUPPORT SYSTEM

The architecture of control systems based on applied semiotics [1, 2, 13] was formed as part of research on situation management systems for complex objects.

The architecture of a semiotic decision support system is oriented to work with objects (situations) described in a natural language. It includes the following subsystems (Fig. 1):

- an input language interpreter, which translates unstructured linguistic information about the control object in a natural language into the internal language of the system;
- an analyzer, which preliminarily classifies the current situation into situations requiring (and not requiring) control;
- a classifier, which generalizes and reduces the current situation to one or more classes of typical situations from a knowledge base to apply one-step control actions;

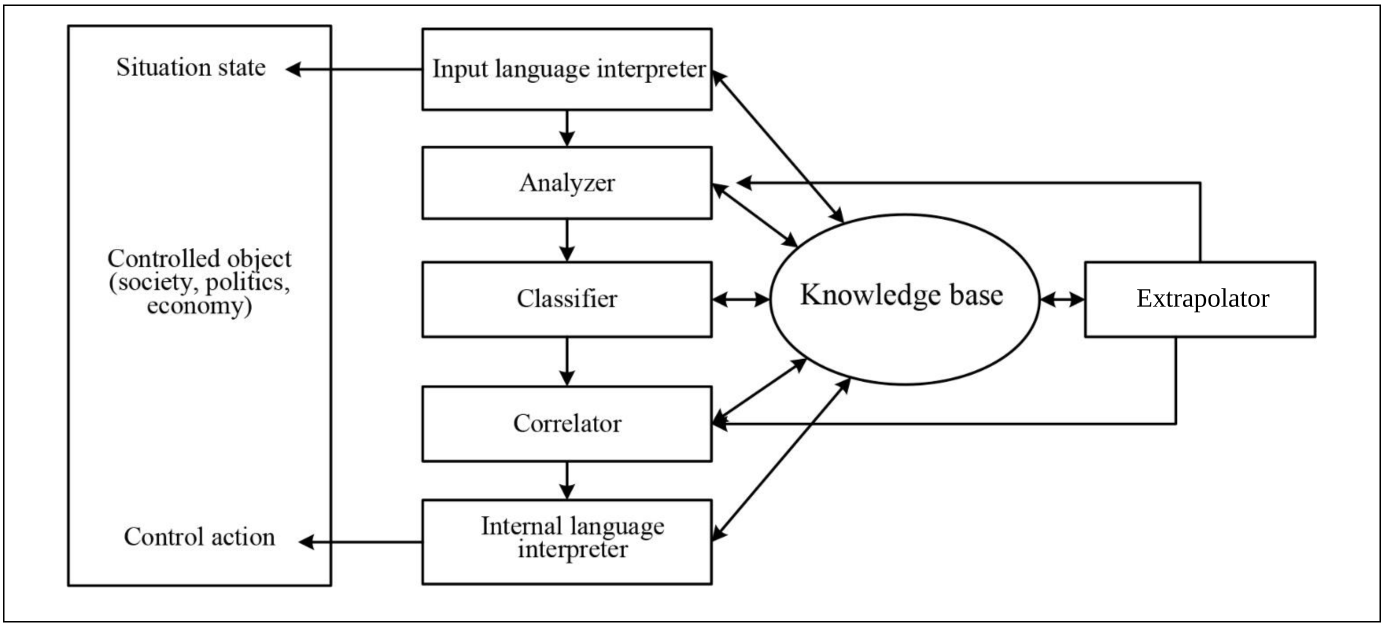

Fig. 1. The architecture of a semiotic system [13].

- a correlator, which forms adequate control actions for the controlled object in the current situations and the current world model from the knowledge base of the system;
- an extrapolator, which assesses the obtained solution based on situation forecasting in the current world model and possibly corrects it (by adjusting the world model in the “analyzer–classifier–correlator” cycle;
- an internal language interpreter, which presents the resulting solution in the natural language.

The main element of this architecture is a knowledge base, which includes a set of typical situations and models of possible worlds. The sequential transfer of the results from the interpreter to the analyzer, classifier, and then the correlator and extrapolator forms an operation cycle of the semiotic architecture.

In management systems built on the principles of applied semiotics, mathematical modeling of situations alternates with control design within fixed formal world models represented in the knowledge base. Semiotic modeling restructures formal world models based on knowledge about the subject domain and real situations that arise during the operation of the object and control system [13].

In this semiotic architecture, the models and possible worlds for decision-making represented in the knowledge base are closed with respect to knowledge of the problem domain.

Under uncertainty, the semiotic system must be open, i.e., it must have the capability to supplement knowledge. Therefore, the main distinctive feature of the semiotic decision support system considered here is that it uses information from the Internet as a knowledge base.

International and Russian standards provide different definitions of information. For example, in information technology, information is knowledge about facts, events, things, ideas, and concepts that has a particular meaning in a definite context (ISO/IEC 2382:2015); in information processing, information is any data, in electronic or any other form, to be processed by information and decision-making systems. Federal law no. 149-FZ “On Information, Information Technology, and Information Protection” interprets information as “knowledge (messages, data) regardless of its form of presentation.”

Data are defined as the presentation of information in a formalized way suitable for communication, interpretation, or processing (ISO/IEC 2382-1:2015).

Depending on the form of presentation, there exist structured and unstructured data. *Structured data* are organized according to a predefined set of rules. A predefined set of rules that regulates the basis of structured data must be clearly established and made public; it can be used to manage data structuring (ISO/IEC 20546:2019(ru), 3.1.35). Examples of structured data are data from database tables or manually or automatically marked-up text. During the markup procedure, certain labels (tags) are assigned to text words. As a result, information can be presented in tabular form and processed.

Unstructured data are characterized by the absence of any structure other than that at the record or file level. An example of unstructured data is free text (ISO/IEC 20546:2019(ru), 3.1.37). Formally, free text has a syntactic structure and conveys a particular meaning. However, text has to be preprocessed with natural language processing and intelligent analysis methods to retrieve information from it.

Note that the architecture described in [13] is abstract. The main ideas were formulated therein: how such a semiotic system and its subsystems can function. Unfortunately, this architecture is impossible to implement. Below, we propose a possible implementation of a semiotic decision support system under uncertainty. The proposed architecture includes a semiotic model, a subsystem to monitor the situation, and a subsystem to generate solution alternatives and explain them. The internal language is set by a subjective semiotic model built by an expert. The input language interpreter is the state monitoring subsystem. This subsystem obtains free text from the Internet and represents it in the internal language of the semiotic system. The decision-making subsystem is based on solving the inverse problem in the semiotic system. In addition, this subsystem implements an interpreter of the internal language. The interpreter translates the solutions obtained in the semiotic system into natural language, naming the class of solutions and explaining the context in which the name is used (this sentence is free text).

Under uncertainty, we propose using unstructured Internet data (free text) as a knowledge base, which is pre-structured by natural language processing methods.

2. THE SEMIOTIC MODEL OF THE SITUATION

The semiotic model of a subject domain proposed in [14] serves as a qualitative subjective model of the situation. The semiotic model of the situation [14] is a subjective qualitative model that represents expert’s knowledge of dynamic systems. Its elements are G. Frege’s sign model [3] connecting the real world (denotation) with mental representations about the world (knowledge) in the form of a sign name (symbol) and meaning, defining its main features (properties). The semiotic model describes the situation in three aspects: syntactic, semantic, and pragmatic. Syntax is responsible for representing the relations between signs describing the reality. Semantics studies the relationship between signs and their significations in the real world. Pragmatics is responsible for the relationship between signs and their users. In the case under consideration, the sign system user is the decision maker (DM).

The syntactic model. When constructing a syntactic model, we employ a logical-linguistic representation for the main elements of the system under study [15]. In logical-linguistic models, the basic elements, relations between them, and their possible states (values) are represented in a natural language. The syntactic model describes the main parts of the system as a set of their names $D = \{d_i\}$, $i = 1,..., M$. The "part–whole" relation is defined on this set: $\Theta \subseteq D \times D$. For each constituent part d_i of the simulated situation, the names of its parameters form the set $F_i = \{f_{ij}\}$, where j is the parameter number of the ith part. The value set $Z_i = \{Z_{ij}\}$ of each parameter is defined as an ordered set of linguistic values, i.e., $Z_{ij} = \{z_{ij1},..., z_{ijq}\}$, where $z_{ijq+1} \succ z_{ijq}$, $q = 0,..., n - 1$. The vector $Z(t) = (z_{1je},..., z_{njq})$ with the values of all situation parameters at a time instant t is called the state of the situation.

The logical-linguistic model is intended to represent the dynamics of all situation parameters and solve the inverse problem (find the parameter values for changing the current state of the situation to the target state). To model the dynamics, we have to determine the cause-and-effect (causal) relations between the parameters and their strength.

The strength of a causality is defined in a natural language from the set of possible values, e.g., RF = {"*Heavily strengthens*," "*Strengthens*," "*Slightly strengthens*"}. The strength of a causality defines a binary relation between the sets of possible parameter values. For example, RF("*Heavily strengthens*") $\subseteq Z_{1j} \times Z_{2u}$ means that the jth parameter of the first subsystem (cause) is related to the uth parameter of the second subsystem (effect). The strength of influence is defined by the pairs of values from the set $Z_{1j} \times Z_{2u}$. As an example, we take the "*Heavily strengthens*" relation; for this relation, the pairs of values $(z_{1j2}, z_{2u3};..., z_{1jn}, z_{2um})$ mean that changing the value of the jth parameter of the first subsystem to the second element (z_{1j2}) of the value set Z_{1j} (cause) will increase the value of the uth parameter of the second subsystem to the third element (z_{2u3}) of the set Z_{2u}, etc.

In the syntactic model, the causal relations between different parameters are determined through expertise as a relation W on the value sets of all parameters. Consider this relation in the form of logical-linguistic equations for the system with f_i parameters ($i = 1,..., n$) and their value sets Z_i. To forecast the situation, we write a causal relation W as a mapping [15]

$$W: Z(t) \to Z(t + 1), \quad (1)$$

where $Z(t) \in \underset{i}{\times} Z_i$, $Z(t) = (z_{1e},..., z_{nq})$ is the state vector of the situation, and $\underset{i}{\times} Z_i$ denotes the set of all possible state vectors ($i = 1,..., n$).

Situation forecasts in the model specified by logical-linguistic equations are often calculated using the theory of fuzzy sets and systems [16]. In this case, membership functions and fuzzy causal relations have to be defined for all parameters. Note that for fuzzy forecasting, all theoretical issues have been settled. However, the procedure of constructing the membership functions and defining the fuzzy relations requires much expert work.

In [17], B. Kosko proposed a fuzzy causal algebra without the need to construct membership functions: it suffices to obtain ordered linguistic values of the strength of causal relations. His idea is to calculate the influence of one parameter on another through chains of causal relations. The strength of influence of a chain is determined by the minimum of all strengths in the chain, and the aggregate strength of all chains is determined as the maximum of all strengths for the parameter of these chains.

In what follows, we forecast the situation using the method proposed in the paper [18].

We construct the linguistic scale of a parameter $Z_{ij} = \{z_{ij1},..., z_{ijq}\}$ as a mapping into a numerical set $X_{ij} = \{x_{ij1},..., x_{ijq}\}$ whose elements are defined on a segment of the numerical axis [0, 1], i.e., $x_{ij1},..., x_{ijq} \in [0, 1]$. The points of the numerical axis form an ordered set X_{ij} of numerical points ordered the same way as the linguistic values: if $z_{ijq+1} \succ z_{ijq}$, then $x_{ijq+1} \succ x_{ijq}$. Thus, a mapping $\varphi: Z_{ij} \to X_{ij}$ is defined. The inverse mapping $\varphi^{-1}: X_{ij} \to Z_{ij}$ allows interpreting any value $x_{ijq} \in [0, 1]$ into a linguistic value $z_{ijq} \in Z_{ij}\ \forall q$.

In this case, the strength of a causality can be treated as a real-valued gain. The values of the cause and effect parameters have the relationship $x_{ijq} = w_{ijpt}^* x_{pto}$, where $x_{ijq} \in X_{ij}$ is the value of the effect parameter and $x_{pto} \in X_{pt}$ is the value of the cause parameter. The gain is $w_{ijpt}^* = 1$ for the "Strengthens" relation, $w_{ijpt}^* > 1$ for the "*Heavily strengthens*," and $w_{ijpt}^* < 1$ for the "*Slightly strengthens*." The issues of determining the strength of causalities (gains) were described in detail in [18].

The linear relationship between the values of the cause and effect parameters is justified under uncertainty with subjective assessments of the strength of the causality.

The situation forecast in the numerical system is calculated from the finite-difference equation

$$X(t+1) = W^{*}\circ X(t),$$

where: $X(t)$ and $X(t+1)$ denote the situation parameters vectors at sequential time instants; $W^{*} = |\ w_{ij}^{*}\ |_{n\times n}$ is the gain matrix; finally, $\circ$ specifies a rule for calculating the forecast values.

We calculate the forecast vector $X(t+1)$ by aggregating the max-product values (multiplication and taking the maximum). Therefore, the ith element of the forecast vector is given by the following rule:

$$x_i\,(t+1) = \max_{r=1,\dots,n} x_r\,(t)\,w_{ir} \quad \forall i.$$

The forecast vector can be written in the linguistic form: $Z(t+1) = \varphi^{-1}\,(X(t+1))$.

Thus, the syntactic model is defined by the quadruple

$$\langle F, Z, W, Z(t)\rangle, \qquad (2)$$

where: F is the set of parameters; Z is the set of parameter value sets; W is a causal relation on the set of parameter values; finally, $Z(t)$ is the state (the vector of all parameter values).

The semantic model. The semantic model of a subject domain describes possible states of the syntactic model (2) as a partially ordered set of the named classes of states.

Such a representation is based on interpreting the space of possible states of the dynamic system (1), $SS = \times_i Z_i$, as a *semantic space*.

In the feature semantic space, the situation states are represented as notions. Real situations (states-denotations) are defined by the names and value vectors of the attributes characterizing their content (meaning). In semantic spaces, situations with close parameter values form classes of states, and certain relations are defined between different classes ("class–subclass" or "genus–species"). In other words, a notional structure is defined.

The paper [19] proposed a method for structuring the state space SS of the dynamic system (1) into nested domains of possible states: $SS(d^H) \subset SS$. These domains have artificial names d^H determining the class of states of system (1), i.e., $SS(d^H) \Leftrightarrow d^H$, $H = 0,\dots, 3^N$, where N is the number of parameters.

For a system with two parameters (features), this method is illustrated in Figs. 2–5 below.

In particular, Fig. 2 shows an example of the semantic space for a situation with two features $F = \{f_1, f_2\}$ and value sets $Z_1 = \{Z_{11},\dots, Z_{1n}\}$ and $Z_2 = \{Z_{21},\dots, Z_{2m}\}$, respectively. The initial state $Z(0)$ of the situation is indicated by a point with the coordinates $Z(0) = (Z_{1q}, Z_{2s})$ in the space $SS = Z_1 \times Z_2$.

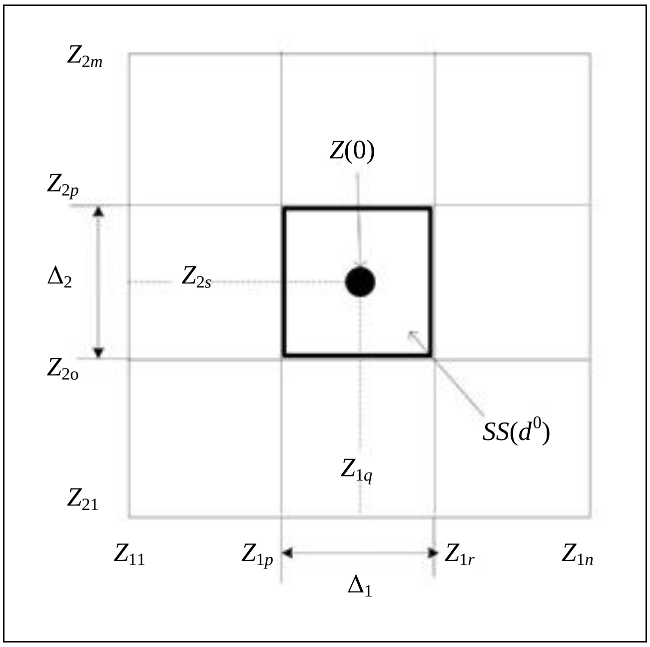

Fig. 2. The domain of the basic class of states.

The neighborhoods of the initial state point by features 1 and 2, $\Delta_1 = \{Z_{1p},\dots, Z_{1r}\}$ and $\Delta_2 = \{Z_{2o},\dots, Z_{2p}\}$, respectively, are assigned through expertise. They are called the initial state tolerance intervals by features 1 and 2. The semantic space domain $SS(d^0) \subseteq SS$ obtained by the direct product of all tolerance intervals (by all features of the state $Z(0)$) defines the basic class of states: $SS(d^0) = \Delta_1 \times \Delta_2$. Any state $Z(t)$ of system (1) from the basic class $SS(d^0)$, i.e., $Z(t) \in SS(d^0)$, has the name d^0. In other words, the basic class defines the class of indistinguishable, equivalent states.

Figure 3 demonstrates the semantic space domains $SS(d^1)$, $SS(d^2)$, $SS(d^3)$, and $SS(d^4)$. For example, the domain $SS(d^3)$ is defined as $SS(d^3) = \{Z_{2o},\dots, Z_{2p}\} \times \{Z_{11},\dots, Z_{1r}\} = \Delta_2 \times \Delta_3$. Since the domains $SS(d^1)$, $SS(d^2)$, $SS(d^3)$, and $SS(d^4)$ include the domain of the basic class of states $SS(d^0)$, they are said to generalize this class. Moreover, the domains $SS(d^1)$ and $SS(d^3)$ generalize the domain $SS(d^0)$ by feature 2 whereas $SS(d^2)$ and $SS(d^4)$ by feature 1. The domains $SS(d^1)$, $SS(d^2)$, $SS(d^3)$, and $SS(d^4)$ have the corresponding names (those of the state classes d^1, d^2, d^3, and d^4, respectively). In the following, we operate the names of the state classes.

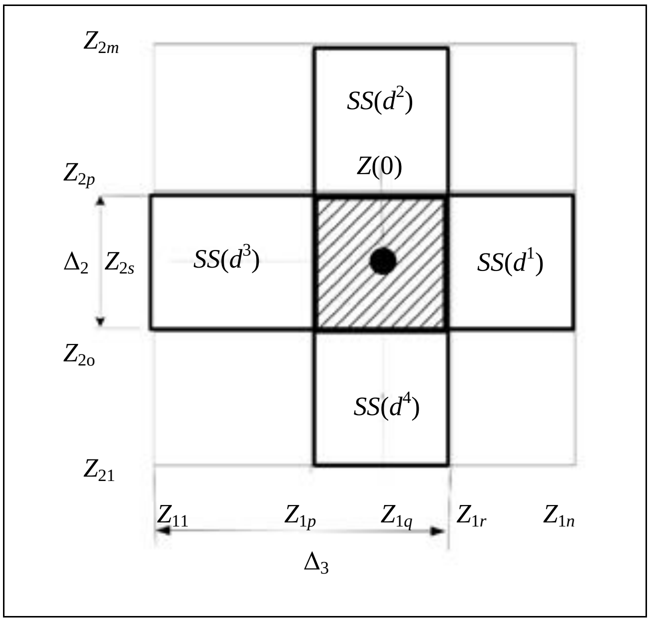

Fig. 3. The domains of generalized state classes by features 1 and 2.

Figure 4 shows the semantic space domains $SS(d^5)$, $SS(d^6)$, $SS(d^7)$, and $SS(d^8)$. For example, the domain $SS(d^5)$ is defined as $SS(d^5) = \{Z_{2o},..., Z_{2m}\} \times \{Z_{1r},..., Z_{1n}\} = \Delta_4 \times \Delta_3$. These domains include the basic class domain and generalized domains by one of the features. For example, $SS(d^5)$ includes the domains $SS(d^1)$ and $SS(d^2)$ and the basic class domain $SS(d^0)$. Due to such nesting, these domains generalize generalized domains by one feature and the basic state class by two features. These domains are denoted by the names d^5, d^6, d^7, and d^8.

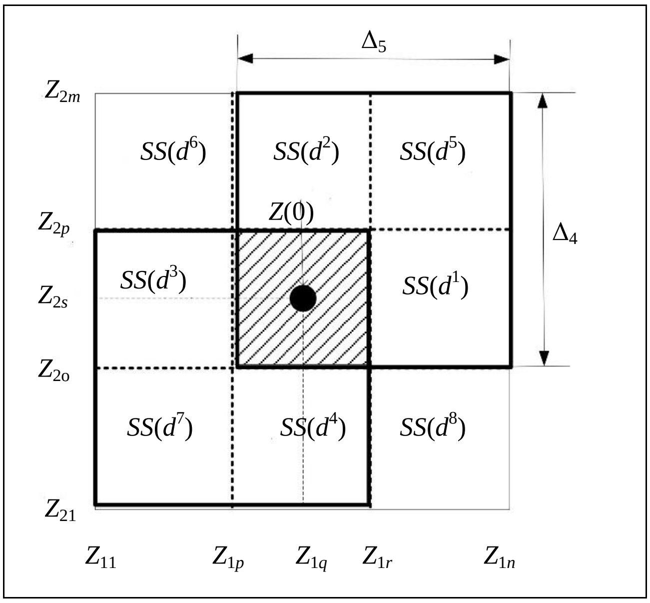

Fig. 4. The domains of generalized state classes by two features.

As shown in [16], the names d^H form a partially ordered set $\{d^H\}$ of the names of state classes $CF = (\{d^H\}, \leq)$ by the nesting $SS(d^H)$ of the state domains. This set is called a qualitative *conceptual framework*, which defines a qualitative ontology of the subject domain with the syntactic model (2).

A Hasse diagram in Fig. 5 defines a partial order of state class names. The first level of class names generalizes the basic class, whereas the second level generalizes the first-level names. Such a qualitative ontology determines a notional system of an ill-defined subject domain.

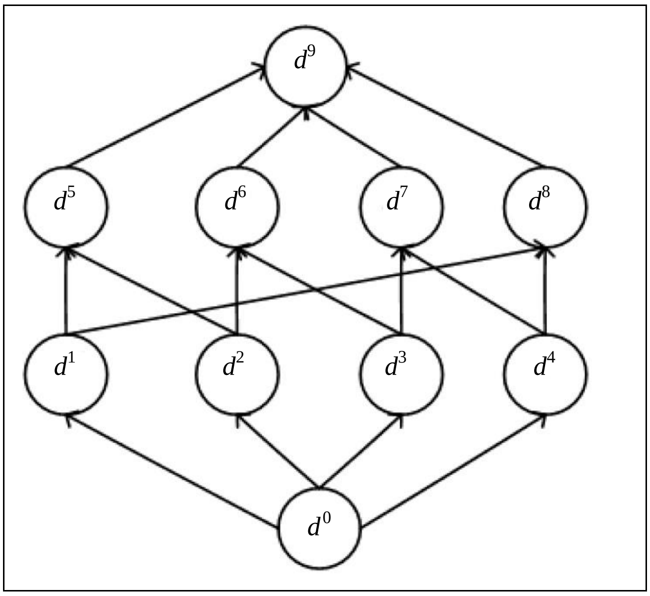

Fig. 5. The conceptual framework of a subject domain.

Thus, the semantics of the syntactic model (2) is defined by the qualitative conceptual framework

$$CF = (\{d^H\}, \leq), \quad (3)$$

where $d^H \Leftrightarrow SS(d^H)$ are the names of state classes that uniquely define the semantic space domains.

Note that at different time instants, the syntactic model states $Z(t)$ may belong to different domains $SS(d^H)$ and, therefore, have different names d^H, $Z(t) \in SS(d^H)$.

Also, we emphasize that the conceptual framework (3) is an idealized semantic model of the subject domain. Here, only the name d^0 of the basic notion (class) is explicitly defined: for all other state classes, we have artificial names d^H and the corresponding semantic domains $SS(d^H)$ they define.

Under uncertainty, when it is impossible to construct an ontology of the subject domain, the conceptual framework defines "reference points" of possible state classes. These points have the "class–subclass" relationship with the basic class. The real (not artificial) names of state classes correspond to the solutions of the inverse problem. We will find them by unstructured data processing (free text processing) methods on the Internet.

The pragmatic model. In semiotics, pragmatics is responsible for the relations between signs and their users. The pragmatic model assesses the utility of the system state for the DM. Assessment is based on determining the values of the expert's preference coefficients α_j with respect to the parameter values in the state vector ($Z(t)$).

The state estimate $O(Z(t))$ is given by the linear convolution

$$O(Z(t)) = \sum_j \alpha_j x_j(t), \; j = 1,..., n, \quad (4)$$

$$\sum_j a_j = 1,$$

where $x_j(t) \in [0, 1]$ is a mapping φ of the linguistic parameter values $z_j(t)$ on a segment of the numerical axis [0, 1], i.e., φ: $z_j(t) \to x_j(t) \in [0, 1]$, $z_j(t) \in Z(t)$, $O(Z(t)) \in [0, 1]$.

The three models have common parameters; therefore, a change in the state of one model will cause a change in the states of the others.

The general statement of the decision problem in the semiotic system is as follows. For the semiotic description (2)–(4) of a complex system, it is necessary to find a new description in the syntactic and semantic models with a better pragmatic assessment O^* compared to the existing one O. A solution method for this problem was proposed in [20]. The method is based on solving the inverse problem in the semiotic model by defining a target in the pragmatic model, solving the inverse problem in this model, and sequentially transferring the solution results from the pragmatic model to the syntactic model and then to the semantic model.

The inverse problem in the syntactic model is solved using methods for solving such problems for systems with qualitatively defined or fuzzy parameters. The methods were considered in [21, 22].

3. THE METHOD FOR SOLVING THE INVERSE PROBLEM

Consider the general algorithm for solving *the inverse problem*. The relation $W = |w_{ij}|_{n \times n}$ between the situation features and the target vector $G = (g_1, g_2,..., g_n)$ of their values are given. The problem is to find the set of input control vectors $\Omega = \{U_k\}$ such that $U_k \circ W = G$ $\forall k$ for all $U_k = (u_{k1}, u_{k2},..., u_{kn})$. This problem is solved in a numerical system, i.e., the elements of the relation W, the target vector $g_i \in G$ and the vector of control actions $u_{ij} \in U$ are defined as real numbers: $w_{ij} \in$ R, $g_i \in [0, 1]$, and $u_{ki} \in [0, 1]$.

Recall that the situation is forecasted using the max-product composition. Therefore, we consider algorithms for solving inverse problems for the max-product composition rule. In this case, it is required to find an inverse mapping for the max-product composition.

An iterative algorithm developed in [22] yields a set of solutions of the inverse problem in the form $\Omega = \{U_{\max}, U_{\min}\}$: one maximum solution $U_{\max}$ and a set of minimum solutions $U_{\min} = \{U_1, U_2,..., U_q\}$, where $U_{\max}, U_1,..., U_q$ are the value vectors of the situation parameters. The iterative algorithm includes the following steps.

1. Find the vector $U_{\max} = (u_{1\max},..., u_{n\max})$ of the maximum solution $U_{\max} = \min(W\alpha' G^{\mathrm{T}})$, where

$$w_{ij}\,\alpha\, g_j = \begin{cases} 1 \text{ if } g_j \geq w_{ij}, \\ \dfrac{g_j}{w_{ij}} \text{ otherwise.} \end{cases}$$

The ith element of the vector $U_{\max}$ is given by $u_{i\max} = \min\limits_{r=1}^{n} w_{ir}\alpha\, g_r$. When determining the vector $U_{\max}$ using the max-product composition, conventional matrix multiplication is replaced by the operation α and conventional matrix summation by the minimum operation.

2. Find the set of minimum solutions

$$U_{\min} = \{ \max \Phi[((W\beta G^{\mathrm{T}})\gamma(U_{\max})^{\mathrm{T}})]\}.$$

2.1. Here, the operation β is defined as follows:

$$U_\beta = w_{ij}\,\beta\, g_j = \begin{cases} 0 \text{ if } g_j > w_{ij} \text{ or } g_j = w_{ij} = 0, \\ \dfrac{g_j}{w_{ij}} \text{ otherwise.} \end{cases}$$

The ith element of the row vector $U_\beta = (u_{\beta 1},..., u_{\beta n})$ is given by $u_{\beta i} = \max\limits_{r=1}^{n} w_{ir}\alpha\, g_r$.

2.2. The operation γ for the matrices U_β and $U_{\max}^{\mathrm{T}}$ is defined as follows:

$$U_\gamma = u_{\beta i}\,\gamma\, u_{i\max} = \begin{cases} 0 \text{ if } u_{i\max} \neq u_\beta, \\ u_{i\max} \text{ if } u_{i\max} = u_\beta. \end{cases}$$

The elements of the matrix $U_\gamma = |u_{\gamma ij}|_{n\times n}$ are given by

$$u_{\gamma ij} = (u_{i\,max}\,\gamma\, u_{\beta j})_{\substack{j=1,...,n, \\ i=1,...,n}}.$$

2.3. The operation $\Phi(U_\gamma)$ forms a set of matrices $\Phi(U_\gamma) = \{\phi(U_{\gamma k})\}$ from U_γ by the following rule:

- Each column of the matrices $\phi(U_{\gamma k})$ contains only one non-zero element, and all other elements are zeroed. Hence, the sum of elements in each column of the matrix $\phi(U_{\gamma k})$ equals its non-zero element.
- Any matrix $\phi(U_{\gamma k}) \in \Phi(U_\gamma)$ contains a unique combination of nonzero column elements of the original matrix $U_{\gamma k}$.

2.4. The operation $\max(\Phi(U_\gamma))$ forms all minimum solutions of the inverse problem by taking the maximum over the rows of each matrix $\phi(U_{\gamma k}) = |u_{\gamma kij}|$. The minimum solutions are given by $U_k = \max\limits_{i=1,\dots,n} \phi(U_{\gamma k})$, $U_k = (u_{\gamma k1},\dots, u_{\gamma kn})$. Thus, the algorithm generates the column vectors $(U_1,\dots, U_k)$ of the solutions.

Applying the inverse mapping φ^{-1} to the elements of the solution vectors, we obtain the vector of linguistic values for the solution of the inverse problem, i.e., $Z^*_k = \varphi^{-1}(U_k) = (z_{k1},\dots, z_{kn})\ \forall z_{ki} \in Z_i$.

All solutions of the inverse problem are represented as points in the structured semantic space CF(3). Then the solutions (the points in the semantic space) fall into different domains $SS(d^H)$ characterizing state classes with different names d^H. Thus, the formal solution of the inverse problem gives a set of names for solution classes, which are structured by the nesting of the domains $SS(d^H)$ corresponding to these names in the form of a qualitative ontology (the conceptual framework of solutions).

The formal names of solution classes are given by the mathematical symbols d^H. In the paper [20], the solution classes in a subject domain were named using methods based on classification [23] and categorization [24] processes studied by psychologists. In particular, a method for determining the compound name of a new class was proposed: This method supplements the name of the basic class d^0 with an estimate of a distinctive feature or features. We explain this method below.

The semantic model defines the basic class domain $SS(d^0) = \bigtimes\limits_i \Delta_i$, where $\Delta_i = z_{i0} \pm \varepsilon_i$ is the tolerance interval for the ith feature, $z_{i0} \in Z_i$ is the feature value in the initial state, and $\pm\varepsilon_i$ specifies the tolerance interval limits for the feature f_i. The solution of the inverse problem is the vector $Z_k = (z_{k1},\dots, z_{kn})$, $z_{ki} \in Z_i$. The solution of the inverse problem in the semantic model (3) will be written as the vector $A_k = (a_{k1},\dots, a_{kn})$, $a_{ki} \in \{-1, 0, 1\}$, where $a_{ki} = -1$ if $u_{ki} < z_{i0} - \varepsilon_i$, $a_i = 1$ if $u_{ki} > z_{i0} + \varepsilon_i$, and $a_i = 0$ if $u_{ki} \in z_{i0} \pm \varepsilon_i$.

The component $a_{ki} \in \{1, 0, -1\}$ in the solution vector $A_k = (a_{k1},\dots, a_{kn})$ qualitatively assesses the value of the ith parameter (f_i) in the inverse problem solution: if $a_{ki} = 1$ ($a_{ki} = -1$), the parameter has a large value (a small value, respectively). This vector can be represented as a vector with the linguistic assessments $L_k = (l_{k1},\dots, l_{kn})$, where l_{ki} = "*Large*" if $a_{ki} = 1$, and l_{ki} = "*Small*" if $a_{ki} = -1$. Then the solution class has the compound artificial name

$$d_k^H = d^0 \mathop{\&}\limits_{i\,(a_{ki} \neq 0)} l_{ki}.$$

For example, consider a basic class with the name d^0 = "*Inflation.*" For this class, possible compound names of new classes by the feature "*Inflation rate*" are d_k^1 = "*Inflation high*" or d_k^2 = "*Inflation low.*"

For each solution from the set $Z_k\ \forall k$, this compound artificial naming method gives a solution expressed in a restricted natural language: the name of the basic solution class d^0 and the assessments (l_{ki}) of the feature values differing from the feature values of the basic class.

Psychology suggests another naming method for solution classes based on the psychological theory of prototypes (categories) [24]. In this theory, a prototypical name is determined by the name of the most characteristic and frequently used name of a representative of this class. Note that a prototype has a sociocultural context. This means that a prototypical name may have different meanings in different social or cultural communities. The meaning in G. Frege's definition [3] is information about an object (in other words, the set of its properties).

In the case under consideration, the prototypical solution class with the name d_k^H is the words (sign symbols) often used with the compound name of the solution class d_k^H in the context of the subject domain.

Semiotic models have a peculiarity: under uncertainty and incomplete knowledge, they represent the set of alternative syntactic models of the situation as a partially ordered set containing the names of state classes of the semantic model (the conceptual framework of the subject domain).

Subjective qualitative semiotic models suffer from the following drawbacks: they are difficult to verify, and the subjective interpretations of modeling results are multiple. The problem essence and some remedies were described in [25, 26]. In this paper, we apply natural language processing methods for a relevant text corpus from the Internet to support the interpretation of solutions in the subject domain.

4. SITUATION MONITORING

Monitoring refers to the process of observing and recording the state of some object or situation. In the case of a technical object with measurable parameters in numerical scales, the monitoring process can be often implemented without difficulties. The parameters of social and political situations are represented in linguistic scales, which reflect opinions, points of view, and the behavior of social groups or individual politicians. For such situations, the program implementation of monitoring is much more complicated.

Comprehensive linguistic analysis (*Knowledge Acquisition*) involving the morphological, syntactic, and semantic analysis of the text yields a semantic network of subject domain concepts. Due to the theoretical difficulties of natural language processing, knowledge acquisition is an unreasonable approach to determining the state of the situation.

The situation monitoring subsystem can be treated as an interpreter of the input language in the semiotic architecture of a decision support system [13].

Figure 6 shows the generalized structure of a situation monitoring subsystem with textual information processing technologies based on the subjective semiotic model.

It includes two main subsystems:

– a subsystem for acquiring and processing unstructured data from the Internet,

– a subsystem of the semiotic situation model.

Situation monitoring is based on the subjective semiotic situation model (2)–(4). The parameters of this model are used as parameters in the information retrieval subsystem for obtaining unstructured data about the current state of the situation from the Internet. The current state means the state of the situation at the observation instant, $Z(t)$. The current state may differ from the initial one $Z(0)$. Unstructured data from news lines are analyzed.

In this subsystem, the situation parameter names of the semiotic model, $f_i \in F$, and their linguistic values Z_i are used for constructing a pattern base $\langle\{f_i\}, \{Z_i\}\rangle$. The situation parameter names $f_i \in F$ and the basic class name d^0 are used when forming a query for the Information Retrieval subsystem. The textual information about the parameter values (the current state of the situation) is extracted from the retrieved information.

To extract information from text, we will apply the technology presented in [27–29]. This technology is oriented to the semiotic model of the situation and uses the following parameters of the syntactic model: $\{f_i\}$ (the set of parameter names) and $\{Z_i\}$ (the set of their possible values). The method constructs patterns on a text corpus of a subject domain. During the pattern construction procedure, a reference element is defined in the sentences of the subject domain (the name of the semiotic system parameter f_i. In the sentence to the left or right of the reference element, an expert uses a text markup program to determine the text values of this parameter.

For example, for the parameter "*Social tension*," possible text values are as follows: "*Single picket*," "*Mass rally*," etc. An expert assigns possible linguistic assessments to these text values: "*Very much grows*," "*Strongly grows*," "*Grows*," "*Slightly grows*," "*Does not change*," "*Slightly decreases*," "*Decreases*," "*Strongly decreases*," and "*Very much decreases*." For example, the text value "*Single picket*" in the current state may be assessed as "*Weakly grows*" whereas the text value "*Mass rally*" as "*Strongly grows*." In this technology, the value scale of each parameter Z_i also has a linguistic assessment scale (see the previously listed assessments). Any textual assessment identified

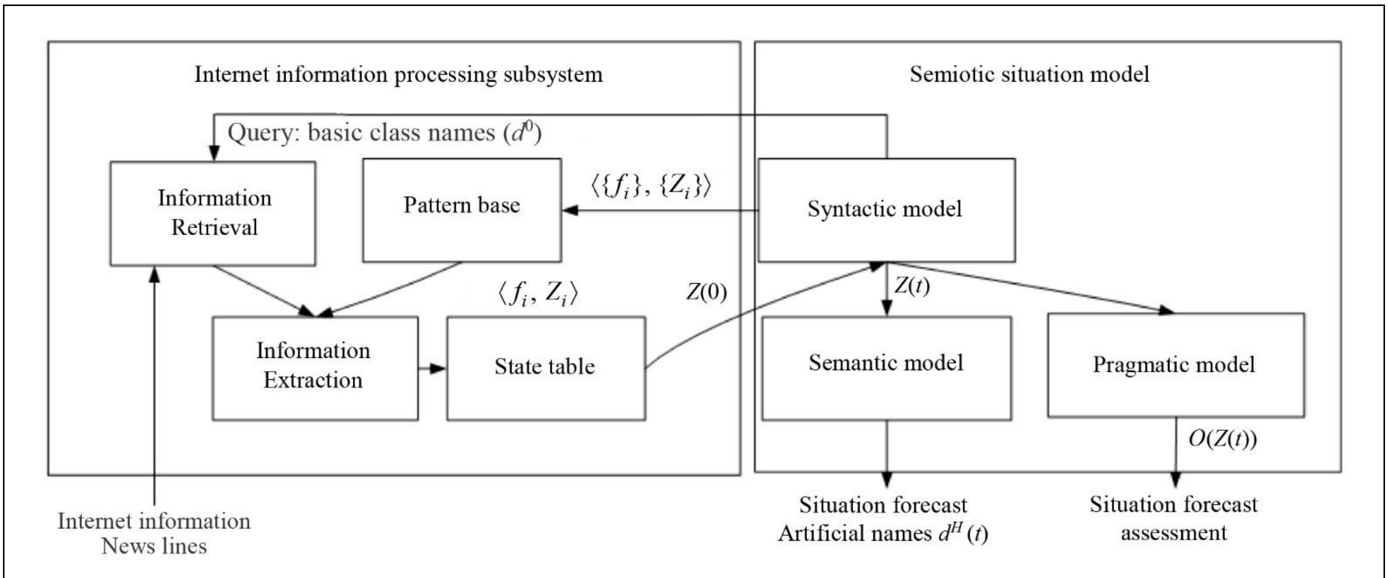

Fig. 6. The generalized structure of the situation monitoring subsystem.

by the expert in a text corpus is given the linguistic value of the parameter scale Z_i if their assessments coincide. Thus, a text pattern is formed: it consists of the parameter name, the text value in the text corpus, and the corresponding linguistic value of the parameter scale.

The quality of the information extraction method depends on the number of constructed patterns. But constructing each pattern requires much expert work. Each text pattern is unique and can be identified only in a particular text. To reduce the amount of expert work and improve the quality of identifying factographic information, the authors proposed an intelligent thesaurus revealing the synonyms of possible text values of the parameters.

The final pattern is formed through the conceptual assembly of a text pattern considering the synonyms of text values and the concepts of the subject domain ontology (the "general–particular" relationship). The pattern base contains conceptual patterns, which are employed to identify classes of similar factographic information considering synonymy and the "part–whole" relationships in the ontology.

Thus, the information extraction subsystem produces the situation parameter vector by extracting the linguistic parameter values: $Z^* = (z^*_{1h},..., z^*_{nq})$, $z^*_{ij} \in Z_i$. This vector is normalized, φ: $Z^* \rightarrow X^*$, $X^* = (x^*_{1h},..., x^*_{nq})$, $x^*_{ij} \in [0, 1]$, and then passed to the situation state table.

If the newly obtained state Z^* differs from the current one $Z(0)$, the semiotic model will forecast the situation. In the syntactic model, the forecast is the value vector $Z(n)$; in the semantic model, the state class name ($d^H \in CF$); in the pragmatic model, the new state assessment $O(Z^*)$.

The main elements of the monitoring subsystem are two technologies: Information Retrieval and Information Extraction. These technologies are studied by many researchers and engineers; different methods and algorithms were proposed for their implementation. With the quality estimations of these technologies available in the literature, we can understand and assess the effectiveness of the proposed monitoring approach based on the semiotic model.

The information retrieval technology is described as a set of Internet search services to get information from the Internet based on queries. Queries include parameters of the semiotic model (the parameter names and basic class names).

The quality of this technology (the completeness and relevance of the retrieved information) is provided by the developers of the corresponding services, and the results can be used due to libraries for different programming languages. Hence, the technology is applicable to end-user software development for specific tasks.

This technology extracts information from the text with a pattern contained in the pattern base; patterns include the semiotic model parameters. This quality of this subsystem is satisfactory for structured data with an explicitly identifiable pattern.

However, in the case of unstructured data (no pattern), this system works only after solving specific linguistic tasks: defining named entities, settling the coreference referential identity, and constructing relationships and scenarios. All these tasks are complex: for example, even an approximate solution of the coreference referential identity problem is possible only in some subject domains with an available knowledge base [29]. According to the presentations at the Message Understanding Conference (MUC-6, 1995), the best solutions of the coreference referential identity problem reached 59% of completeness under 72% of accuracy. Human performance in this case was estimated at 80% [30]. These figures are considered some quality limit of Information Extraction when analyzing unstructured data, reflecting the natural language properties. Further quality improvement of this subsystem for unstructured data incurs considerable costs [30].

Methods for extracting generic relations from text to supplement the taxonomies, thesauri, and ontologies of subject domains are of interest. Several international conferences and competitions [31–33] were organized on hyperonym extraction algorithms for the automatic or automated enrichment of the existing taxonomies of English and other Western European languages.

At the Dialogue 2020 conference (Moscow, 2020) the task was set to extract hyperonyms for the automated enrichment of RuWordNet, a Russian-language thesaurus [34]. The task was to find hyperonyms for a target word (noun or verb) based on text corpus analysis [35]. The developers proposed combined methods with calculating the co-occurrence vectors of the target word and the set of words from the text corpus [35]. The set of candidate hyperonyms from the co-occurrence vector is selected using different techniques (word weighting based on heuristics, closeness estimation for the text corpus word vectors and the vectors of known taxonomies and thesauri marked manually).

Different dictionaries (e.g., Wiktionary), lexical templates, and pre-trained multilingual neural networks (R-BERT) [36] are used to extract hyperonyms as well.

Nowadays, there are many commercial systems implementing Information Extraction. Most of them preliminarily prepare and structure text corpora and extract numerical information about the values of some parameters.

We note GATE (General Architecture for Text Engineering), a modular natural language processing system developed at the Department of Computer Science, the University of Sheffield [37]. ANNIE (ANearly-NewIEsystem) [38], an information extraction system, was developed based on GATE's architecture.

Presently, it seems reasonable to apply the information extraction technology together with the subjective semiotic model in situation monitoring to preliminarily structure text corpora and construct patterns for identifying situation dynamics. This approach eliminates a considerable part of the routine work of the analyst in situation monitoring.

5. DECISION-MAKING IN THE SEMIOTIC SYSTEM

The monitoring subsystem assesses the situation forecast, outputting $O(Z(n))$. If $O(Z(n))$ is worse than the current assessment $O(Z(0))$, the decision-making problem arises. In [20], decision-making was reduced to solving the inverse problem in the semiotic model. In this case, the target vector $G = (g_1, g_2,..., g_n)$ is set (Section 3). Its elements contain parameter values that will improve the pragmatic assessment of the situation from the expert's point of view, i.e., $O(G) > O(Z(n))$. Solving the inverse problem with a given target in the syntactic model yields the set of solutions U_{max}, U_1,..., U_k. These solutions are the control actions (solution alternatives) for achieving the target and, consequently, improving the pragmatic assessment. These solutions are described in the semantic model by the compound names of the solution classes d_k^H. The compound names are represented in the internal formal language of the expert (the developer of the subjective expert model). Under uncertainty, the expert's reasoning, justification, and choice of an acceptable solution using the subjective model are possible within his knowledge, which may be incomplete and contradictory. In this case, an external knowledge base is needed to support the expert's work, e.g., unstructured data (free text) from the Internet.

The decision-making subsystem based on the semiotic model is intended to find the interpretations of artificial compound names of solution classes on the Internet and explain them.

The generalized structure of the decision-making subsystem with textual information processing technologies based on the subjective semiotic model is shown in Fig. 7.

The decision-making system includes the following main subsystems:
– the semiotic model subsystem,
– the unstructured Internet data processing subsystem,
– the alternative solution subsystem (generation and explanation).

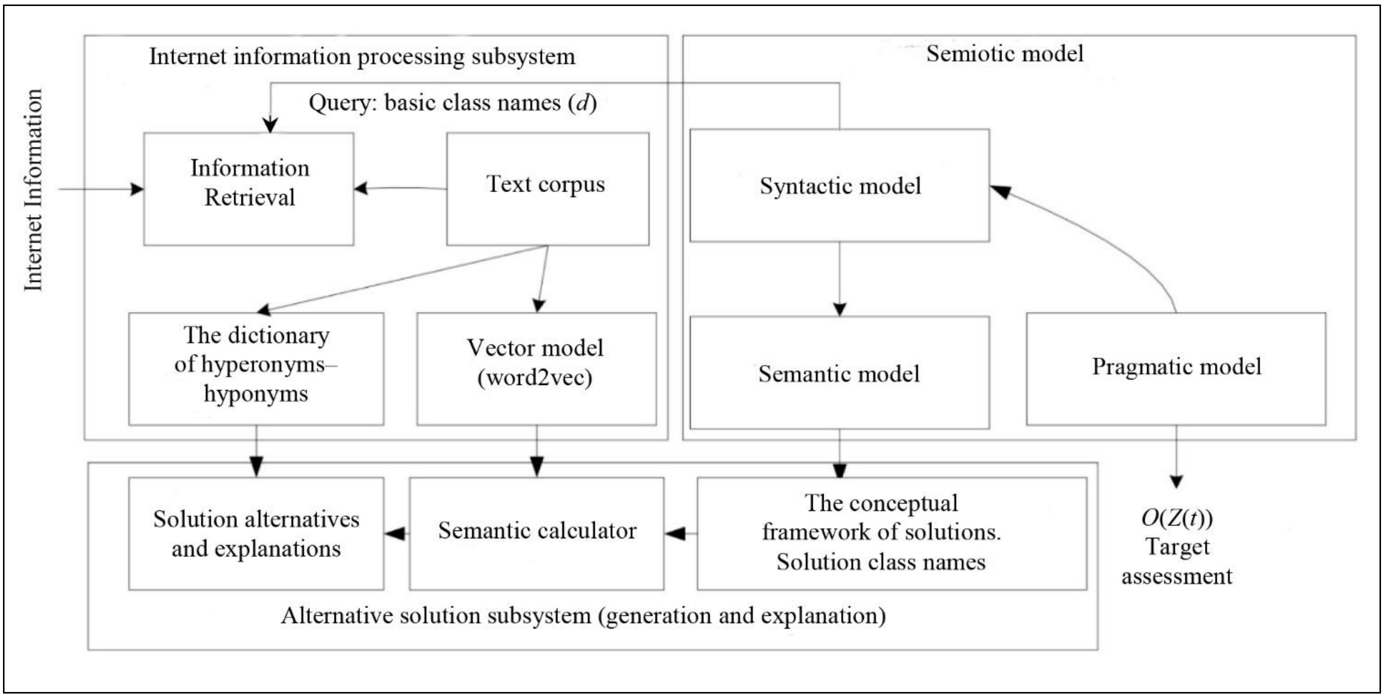

Fig. 7. The generalized structure of the decision-making subsystem.

5.1. The Unstructured Internet Data Processing Subsystem

This subsystem includes the following blocks: information retrieval on the Internet; text corpus; the dictionary of "hyperonyms–hyponyms"; the vector model.

Text corpus. The information retrieval block is to find as much information relevant to the query as possible. An information retrieval query on the Internet includes the names of the basic concepts of the semiotic model. As a result, we obtain a text corpus relevant to the name of the basic concept of the semiotic model. However, the text corpus needs normalization considering the syntactic relationships of the generic relations to build the word2vec vector model [39]. For this purpose, a syntactic window separates the nouns in the text corpus and reduces them to the normal form (nominative case, singular).

The vector model. The vector model of a text corpus is based on the distributive analysis of texts. According to [40], distributive analysis is a method to study languages depending on the environment (distribution) of individual linguistic units in the text without information about their lexical or grammatical meaning. In distributive analysis, each word (word combination) in some text is represented as a vector of words used jointly with this word in a given context. Each pair of words in this vector is characterized by the frequency of their co-occurrence in this context. Under the hypothesis formulated in [41], the linguistic units occurring in similar contexts have the close vectors of jointly used words.

Formally, this technology can be represented as follows. Consider a given text corpus, i.e., a set of sentences characterizing a subject domain. In distributive analysis, syntactic relationships in sentences are ignored. The subject domain is characterized by the word set of all sentences, $Tp = \{v_{gh}\}$, where g is the sentence number and h is the word number in the sentence (text corpus). The words without repetitions are defined on the set of all words as a word subset $V \subseteq Tp$, called a subject domain dictionary. It has the form $V = \{v_r\}$, where $r = 1,..., q$ are the word numbers in the dictionary. The joint usage of dictionary words in a given context is defined as the relation

$$R_{w2v}: V \times V \rightarrow r_{ij},$$

where $r_{ij} \in [0, 1]$ characterizes the co-occurrence of the words v_i and v_j in the subject domain under consideration.

For each dictionary word $v_r \in V$, the vector $R_p = (v_1/r_{p1},..., v_q/r_{pq})$, $p = 1,..., q$, characterizes its co-occurrence r_{pi} with other words $(v_1,..., v_q)$ of the considered subject domain (the so-called context vector). A slash in the context vector separates a dictionary word and its co-occurrence with other words.

Currently, the mapping R_{w2v} is constructed by the machine learning of an artificial neural network [39] in which the training sample is a text corpus Tp.

The word2vec technology introduces operations with word vectors to define new vectors determining the joint usage frequency of individual words from a dictionary V with other words of a subject domain. Operations in the word2vec technology can be represented as a mapping

$$\text{w2v}:((\bullet)(v_1,..., v_p)) \rightarrow R^*_w,$$

where the resulting vector R^*_w characterizes the joint usage frequency of words $(v_1,..., v_n)$ with other words of a subject domain.

Here $(\bullet)$ are the operations defined in word2vec as positive() and negative(). The resulting vector R^*_w of the positive() operation characterizes the joint usage frequency of the argument-specified words with other words of a subject domain. The resulting vector R^*_w of the negative() operation characterizes the frequency of subject domain words that are not used with the argument-specified words.

The operation of such vector models can be illustrated by Google's browser. When typing a word in the search line, the system shows a word vector frequently used with this word; adding one more word, we get another hint (a word vector frequently used with two words), etc. All typed words are arguments of the positive() operation. The word2vec technology allows searching for words that are not used jointly (the argument of the negative() operation). This operation also outputs a word vector.

Context vectors in this technology include words and their co-occurrence with other words of a subject domain. In this paper, words are the names of signs (G. Frege [3]) that denote a real object or situation and determine its properties (the meaning of the object or situation). In G. Frege's definition, the meaning is information about an object (i.e., a set of its properties in the word usage context). Given a word denoting an object, we can determine its properties (meaning).

In other words, the word2vec technology defines the function

$$\text{w2v}(\text{positive}(v_t,..., v_s); \text{negative}(v_q,..., v_n)) = R^*_w, \quad (5)$$

where $(v_t,..., v_s) \in$ V are the argument-specified words of the positive() operation and $(v_q,..., v_n) \in$ V are the argument-specified words of the negative() operation.

Recall that the name of a solution class is a prototype name, which is determined by the frequency of use to denote a category. The context vector contains information about the co-occurrence of words. Therefore, we assume that words with large co-occurrence values with the argument-specified words of function (5) can be prototype names.

The dictionary of hyperonyms-hyponyms. This dictionary reflects the generic relations extracted from a text corpus of a subject domain. They are extracted using lexico-syntactic patterns [42]. A lexico-syntactic pattern is a structural pattern of a linguistic construction reflecting its lexical and surface-syntactic properties. In the general case, a pattern defines a sequence of linguistic construction elements and sets grammatical agreement conditions for them.

In the scientific literature, there are many works devoted to extracting generic relations from English and Russian texts, assessing the identification quality of relations, and constructing and debugging patterns and their applications. According to the authors, the patterns proposed in [43] allow extracting 78.5% of generic relations contained in a text.

The patterns [43] were adopted to develop algorithms for extracting generic relations from a typical text of a subject domain. The morphological characteristics of the words in the analyzed sentences were used to develop the algorithms implementing the patterns. In particular, hyperonyms and hyponyms were considered to be nouns with the same animate characteristic. The patterns involved the agreement rule of case endings for the hyperonyms and hyponyms in a sentence.

The rules based on morphological analysis can improve the quality of extracting generic relations in sentences.

In distributive analysis, the words on the left and right of a given word are equivalent since the syntactic relationships between them are excluded. To eliminate this drawback of distributive analysis, it was proposed to use separate dictionaries for model words and contexts [44, 45]. We include the context explaining the generic relations in the dictionary of hyperonyms–hyponyms.

Lexico-syntactic patterns allow forming the dictionary of hyperonyms–hyponyms of a subject domain and the context dictionary associated with these words [46]. In the cited work, the context dictionary was extended to the explanatory dictionary, which contains sentences with generic relations extracted using patterns. Adding the context to the dictionary of hyperonyms–Hyponyms helps to select a hyperonym as a prototype, focusing on its usage context in the subject domain.

This is how the dictionary of hyperonyms–hyponyms and sentences describing generic relations is formed. Thus, the corresponding structure is described by the triplet

$$\langle \text{HYPER, HYPO, Context} \rangle. \qquad (6)$$

HYPER and HYPO are the sets of hyponyms and hyperonyms, respectively, of a subject domain reduced to the normal form; Context is the set of sentences with defined generic relations.

Note that the generic relations included in the dictionary reflect the structure of knowledge about the subject domain. As a matter of fact, they are elements of the subject domain ontology. In the semantic model, we have introduced a qualitative ontology of an ill-defined subject domain as an idealized conceptual structure with artificial names (the conceptual framework (3)). The identified hyperonyms can replace the artificial names of the state class of the conceptual framework, and hyponyms can serve as the name of this class. By extracting generic relations, we try to identify possible names of the state classes of the idealized conceptual framework obtained from the text corpus of the subject domain.

5.2. The Alternative Solution Subsystem (Generation and Explanation)

The subsystem for generating solution alternatives and their explanations includes a conceptual framework of solutions, a semantic calculator, and a block for identifying and explaining alternatives for solution class names.

The conceptual framework of solutions. The formal solution of the inverse problem in the semiotic model gives a set of names of solution classes structured as a qualitative ontology (a conceptual framework of solutions). These solutions are sign symbols with names and content and are expressed in an internal language of the semiotic model. They must be interpreted in the subject domain under consideration.

The solution of the inverse problem in the semantic model (3) is written as the vector $A_k = (a_{k1},..., a_{kn})$, where $a_{ki} \in \{1, 0, –1\}$, where a_{ki} qualitatively assesses the value of the ith parameter (f_i) in the solution: a_{ki} = 1 if the parameter has a large value and a_{ki} = –1 otherwise.

For example, the solution vector (1, 0, 0) means that the value of the first feature in the inverse problem solution is significantly larger than its counterpart in the basic notion d^0, whereas the other two features remain the same. This vector defines a domain of the semantic space $SS(d_k^H)$ and correspondingly the name of a solution class. All solution class names (d_k^H) can be represented as a partially ordered set by the nestings of the semantic space domains they define, (d_k^H, ≤). As demonstrated above, all solution classes have compound names. These names will be employed below to search a text corpus for prototype names using a semantic calculator.

The semantic calculator. It is intended to determine the joint usage vector for words included in the compound name of the inverse problem solution class obtained in the semiotic model (on the one part) and the words of a text corpus of the subject domain (on the other part). The semantic calculator is based on the trained distributive model of the subject domain. Its operation is described by function (5), where the arguments are the artificial names of the inverse problem solution classes. The compound name of a solution contains the basic notion name and qualitative assessments of the dynamics of different parameters: "*Large*," "*Small*", or their synonyms.

In the semantic calculator, the inverse problem solution $A_k = (a_{k1},..., a_{kn})$ in the semantic model is written as

$$\text{w2v (positive}(d^0, f_t|a_{kt} = 1,..., f_s|a_{ks} = 1);$$
$$\text{negative}(f_q|a_{kq} = -1,..., f_n|fa_{kn} = -1)) = R^*_w,$$

where f_t and f_s are the names of the model parameters for which the element $a_{kt} = 1$ is included in the argument of the positive() operation, and f_q and f_n are the names of the model parameters for which the element $a_{kq} = -1$ is included in the argument of the negative() operation. The basic class name (d^0) is also added to the argument of the positive() operation.

The calculator yields the word vector $R^*_w = (v_i/r_{i1},..., v_n/r_{in})$, which orders the joint usage of words (r_{ij}), the model parameters determined in the inverse problem solution, and all words of a text corpus of the subject domain included in the dictionary $v_i \in V$. As stated above, words with a high frequency of occurrence can be regarded as name prototypes for a solution class.

Alternatives names of solution classes and their explanation. Solutions in the semiotic system are possible names of solution classes. The hyperonyms extracted from a text corpus can be the names of solution classes since they define the elements of the ontology of the subject domain. Therefore, we find the intersection of all hyperonyms from the word dictionary (6) in the solution vector R^*_w to obtain alternative solution classes. Let the possible names of solution classes be written as

$$\langle(V \cap \text{HYPER}); \text{Context}\rangle,$$

where the intersection of the set $V = \{v_i\} \in R^*_w$ and the set of hyperonyms from the dictionary (5) gives the set of solution class names, and Context (the sentence text) helps to choose the desired name.

6. AN EXAMPLE AND EXPERIMENTS

The proposed semiotic system was experimental tested for the decision support subsystem [46]. The semiotic model of a sociopolitical situation was developed. The following elements were defined in the syntactic model: "*Power*" (d_1^0), "*Population*" (d_2^0), "*Economy*" (d_3^0), and "*Oligarchs*" (d_4^0) as the basic notions; the features of these notions, $f_i \in F$; the possible values $Z_i \in Z$ of the features and a causal network W.

The basic notion "*Oligarchs*" was assigned the features "*The level of discontent*" and "*The level of patriotism*." For a given target $O(Z(t))$, when solving the inverse problem, the feature "*The level of discontent*" was increased for "*Oligarchs*." Thus, the new notion d_4^1 ("*Oligarchs*" with a high value of "*The level of discontent*") was obtained. In the semantic model, this solution is formally represented by the vector $A_4 = (1, 0)$ and denoted by the artificial name d_4^1 = "*Discontented oligarchs*."

It is required to interpret this solution in the subject domain.

For this purpose, a program layout was developed in Python3.

WebScraper was developed to extract relevant information from the Internet and build the text corpus. Information from 150 URLs (sites) was read; in addition, the text corpus was supplemented with the book [47] devoted to Russia's oligarchs. The Google library, googlesearch, was used as a search engine with the following parameters: the name of the basic notion and the number of links to the retrieved web pages. The syntactic analysis of the html code of the retrieved web pages (parsing) was carried out using BeautifulSoup, a Python3 library.

Lexico-syntactic patterns were developed and debugged to build the dictionary of hyperonyms–hyponyms. When constructing the patterns, the morphological analysis of the Russian text was performed using Pymorphy2 [48].

The vector model of the text corpus was obtained using word2vec. The word2vec model was trained with the following parameters: the training model—skipgram, the training window—5, training iterations––10, the aggregation method— softmax, the word occurrence threshold—3, and the word vector dimension—150.

The dictionary and vector model were stored in a SQLite-3 database.

SQL queries to the SQLite-3 database were developed to generate solution alternatives and their explanations. They return the names of solution classes with comments.

Example. To interpret the inverse problem solution d_4^1= "*Discontented oligarchs*", we normalized it and substituted the result in the semantic calculator:

$$\text{w2v(positive("}\textit{Oligarchs}\text{," "}\textit{Discontent}\text{"))} = R^*_w.$$

The trained word2vec model yielded the word vector R^*_w reflecting the joint usage frequency of the words "*Oligarchs*" and "*Discontent*":

R^*_w = (*Harm/0.904; Fact/0.885; Respondent/0.873; Expert/0.872; Annexation/0.866; Regret/0.863; Position/0.852; Factor/0.844; Trend/0.833; Claim/0.830; Distrust/0.817; Effectiveness/0.813; Advantage/0.808; Character/0.806; Request/0.805; Reason/0.805;...*)

Then we obtained possible names of the solution classes by intersecting the word vector R^*_w with the hyperonyms of the subject domain:

($W \cap$ HYPER) = (*Harm/0.904; Fact/0.885; Regret/0.863; Position/0.852; Claim/0.830; Distrust/0.817; Character/0.806;...;*).

Clearly, in the word vector, "*Harm*" is the closest word to the inverse problem solution with the name "*Discontented oligarchs*": the co-occurrence is 0.904. Therefore, a possible new class of solutions is the one named "*Harmful oligarchs*."

The context of the candidate hyperonym was analyzed to select the solution class names through expertise. For example, the following context was found for the word "*Distrust*" with a co-occurrence of 0.817: "Between this category of business and the conditional "collective Putin" there has been a steady mutual distrust: the former has always feared the seizure of property, whereas the "collective Putin" has feared disloyalty."[1] ♦

Thus, the proposed semiotic architecture of the decision support system allows getting alternative names of the solution classes and choosing a solution based on relevant text analysis.

Note that the text corpus of the example included about 30 000 sentences describing different aspects of the subject domain. With the described approach, the possible names of solution classes were determined by analyzing a much smaller number of sentences. This illustrates the effectiveness of the method: the routine analytical work of an expert is reduced, and his intellectual productivity is improved accordingly.

[1] https://carnegie.ru/commentary/76115 (Accessed February 24, 2022.)

CONCLUSIONS

This paper has proposed a semiotic decision support system in complex dynamic systems under uncertainty. The support is based on extracting, processing, and structuring information from the Internet and a relevant semiotic model of the situation. This model includes three parametrically interconnected submodels: syntactic, semantic, and pragmatic. The inverse problem solution in the semiotic model has been represented as a qualitative ontology of solution classes (the conceptual framework of solutions). Methods for determining and interpreting solution class names extract relevant information from the Internet. Lexico-semantic patterns in a text corpus of a subject domain serve to define the dictionary of generic relations (hyperonyms–hyponyms) and the contextual dictionary. Distributive semantics methods (word2vec) have been applied to construct a semantic calculator. This calculator determines the meanings of the solution class names in the conceptual framework.

Experimental testing of the proposed architecture has shown its effectiveness. Further experimental research will aim at improving the quality of the proposed approach by increasing the volume of text corpora of a subject domain, using the free dictionaries of hyperonyms–hyponyms, and performing the additional semantic analysis of sentences containing solution vector words with a high co-occurrence with the compound name of the solution class.

REFERENCES

1. Pospelov, D.A., *Situatsionnoe upravlenie: teoriya i praktika* (Situation Management: Theory and Practice), Moscow: Nauka, 1986. (In Russian.)
2. Pospelov, D.A. and Osipov, G.S., Applied Semiotics, *Novosti Iskusstvennogo Intellekta*, 1999, no. 1, pp. 9–35. (In Russian.)
3. Frege, G., On Sense and Nominatum, *Readings in Philosophical Analysis*, 1949, pp. 85–102.
4. Osipov, G.S., Chudova, N.V., Panov, A.I., and Kuznetsova, Yu.M., *Znakovaya kartina mira sub"ekta povedeniya* (The Sign Picture of the World of the Behavioral Subject), Moscow: Fizmatlit, 2018. (In Russian.)
5. Ch.S. Peirce: What is Sign?, *Vestnik Tomsk. Gos. Univ. Filosofiya. Sotsiologiya. Politologiya*, 2009, no. 3(7), pp. 88–95. (In Russian.)
6. Morris, Ch.W., *Foundations of the Theory of Signs*, The University of Chicago Press, 1938.
7. Barron, T.M., Chiang Roger, H.L., and Storey, V.C., A Semiotics Framework for Information Systems Classification and Development, *Decision Support Systems*, 1999, no. 25, pp. 1–17.
8. Andersen, P.B., *A Theory of Computer Semiotics: Semiotic Approaches to Construction and Assessment of Computer Systems*, Cambridge University Press, 2006.
9. Goguen, J., An Introduction to Algebraic Semiotics, with Applications to User Interface Design, in *Computation for Meta-*

phor, Analogy and Agents, Nehaniv, Ch., Ed., Springer Lecture Notes in Artificial Intelligence, 1999, vol. 1562, pp. 242–291.

10. Furtado, A.L., Casanova, M.A., and Barbosa, S.D.J., A Semiotic Approach to Conceptual Modelling, in *ER 2014: Conceptual Modeling*, Yu, E., Dobbie, G., Jarke, M., and Purao, S., Eds., Lecture Notes in Computer Science, 2014, vol. 8824, pp. 1–12.
11. Massel, L.V. and Massel, A.G., Situation Management and Semantic Modeling in Power Engineering, *Proceedings of the 4th International Conference on "Open Semantic Technology for Intelligent Systems" (OSTIS-2014)*, Minsk, 2014, pp. 111–116.
12. Massel, L.V., Creation Problems of Intelligent Systems of Semiotic Type for Strategic Contingency Management in Critical Infrastructures, *Information and Mathematical Technologies in Science and Management*, 2016, no. 1, pp. 7–27. (In Russian.)
13. Ehrlikh, A.I., Applied Semiotics and Management of Complex Objects, *Software & Systems*, 1997, no. 3, pp. 3–11. (In Russian.)
14. Kulinich, A.A., Semiotic Approach in Modeling and Decision-making in Ill-defined Complex Situations, *2018 Eleventh International Conference "Management of Large-Scale System Development" (MLSD-2018)*, Moscow, October 1–3, 2018, Trapeznikov Institute of Control Sciences RAS, pp. 1–5. DOI: 10.1109/MLSD.2018.8551835.
15. Pospelov, D.A., *Logiko-lingvisticheskie modeli v sistemakh upravleniya* (Logic-Linguistic Models in Control Systems), Moscow: Energoizdat, 1981. (In Russian.)
16. Borisov, V.V., Kruglov, V.V., and Fedulov, A.S., *Nechetkie modeli i seti* (Fuzzy Models and Networks), 2nd ed., Moscow: Goryachaya Liniya-Telekom, 2012. (In Russian.)
17. Kosko, B., Fuzzy Cognitive Maps, *International Journal of Man-Machine Studies*, 1986, vol. 24, iss. 1, pp. 65–75.
18. Kulinich, A.A., Cognitive Maps Modelling Computer Systems (Approaches and Methods), *Control Sciences*, 2010, no. 3, pp. 2–16. (In Russian.)
19. Kulinich, A.A., Conceptual Frameworks of Ontologies for Ill-structured Problem Domains, *Scientific and Technical Information Processing*, 2015, vol. 42, pp. 411–419.
20. Kulinich, A.A., Solution of the Inverse Problem in Semiotic Systems under Uncertainty, *Trudy 9-oi Mezhdunarodnoi nauchno-prakticheskoi konferentsii "Integrirovannye modeli i myagkie vychisleniya v iskusstvennom intellekte"* (Proceedings of the 9th International Scientific and Practical Conference "Integrated Models and Soft Computing in Artificial Intelligence"), Kolomna, 2019, vol. 2, pp. 572–582. (In Russian.)
21. Pedrycz, W., Fuzzy Models and Relational Equations, *Math. Modeling*, 1987, no. 9, pp. 427–434.
22. Pedrycz, W., Algorithms for Solving Fuzzy Relational Equations in a Probabilistic Setting, *Fuzzy Sets and Systems*, 1990, no. 38, pp. 313–327.
23. Vygotskii, L.S., *Myshlenie i rech'* (Thinking and Speaking), Moscow: Pedagogika, 1982. (In Russian.)
24. Rosch, E., Principles of Categorization, in *Cognition and Categorization*, Rosch, E. and Lloyd, B.D., New York: Lawrence Erlbaum Association, 1978.
25. Abramova, N.A., Expert Verification in Formal Cognitive Map Application. Approaches and Practices, *Large-Scale Systems Control*, 2010, no. 30.1, pp. 371–410. (In Russian.)
26. Abramova, N.A., Interdisciplinary Approach to Verification in Decision-Making with Formal Methods, in *Handbook on Psychology of Decision-Making: New Research. Series: Psychology of Emotions, Motivations and Actions*, Hauppauge, USA: Nova Science Publishers, 2012, pp. 89–111.
27. Rubashkin, V.Sh., Ontology – Problems and Solutions. The Designer, *Proceedings of International Conference on Computational Linguistics and Intellectual Technologies (Dialogue-2007)*, Bekasovo, 2007, pp. 481–485. (In Russian.)
28. Pivovarova, L.M., Factographic Text Analysis in a Decision Support System, *Vestnik of St. Petersburg State University. Language and Literature*, 2010, no. 4, pp. 190–197. (In Russian.)
29. Rubashkin, V.Sh., *Predstavlenie i analiz smysla v intellektual'nykh informatsionnykh sistemakh* (Representation and Analysis of Meaning in Intelligent Information Systems), Moscow: Nauka, 1989. (In Russian.)
30. Grishman, R., Information Extraction: Techniques and Challenges, in *Information Extraction*, Pazienza, M.T., Ed., Lecture Notes in Artificial Intelligence, Rome: Springer-Verlag, 1997, vol. 1299, pp. 10–27.
31. Camacho-Collados, J., Delli Bovi, C., Espinosa-Anke, L., et al. SemEval-2018 Task 9: Hypernym Discovery, *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, Stroudsburg, 2018, pp. 712–724.
32. Jurgens, D. and Pilehvar, M.T., SemEval-2016 Task 14: Semantic Taxonomy Enrichment, *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California, 2016, pp. 1092–1102.
33. Bernier-Colborne, G. and Barriere, C., CRIM at SemEval-2018 Task 9: A Hybrid Approach to Hypernym Discovery, *Proceedings of the 12th International Workshop on Semantic Evaluation*, New Orleans, Louisiana, 2018, pp. 725–731.
34. Loukachevitch, N.V. and Dobrov, B.V., RuThes Linguistic Ontology vs. Russian Wordnets, *Proceedings of the 7th Global Wordnet Conference*, Tartu, 2014, pp. 154–162.
35. Nikishina, I., Logacheva, V., Panchenko, A., and Loukachevitch, N., RUSSE'2020: Findings of the First Taxonomy Enrichment Task for the Russian Language, *Proceedings of International Conference on Computational Linguistics and Intellectual Technologies (Dialogue 2020)*, Moscow, 2020, pp. 579–595. DOI: 10.28995/2075-7182-2020-19-579-595.
36. Tikhomirov, M.M., Loukachevitch, N.V., and Parkhomenko, E.A., Combined Approach to Hypernym Detection for Thesaurus Enrichment, *Proceedings of International Conference on Computational Linguistics and Intellectual Technologies (Dialogue 2020)*, Moscow, 2020, pp. 736–746. DOI: 10.28995/2075-7182-2020-19-736-746.
37. Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V., GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, 2002.
38. *GATE's ANNIE System*. URL: http://gate.ac.uk/ie/annie.html. (Accessed February 24, 2022.)
39. Mikolov, T., Sutskever, I., View, M., et al., Distributed Representations of Words and Phrases and Their Compositionality, in *Advances in Neural Information Processing Systems: the 27th Annual Conference on Neural Information Processing Systems*. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, 2013, pp. 3111–3119.
40. *Lingvisticheskii entsiklopedicheskii slovar'* (The Linguistic Encyclopedic Dictionary), Yartsev, V.N, Ed., Moscow: Sovetskaya Entsiklopediya, 1990. (In Russian.)

41. Sahlgren, M., The Distributional Hypothesis, *Italian Journal of Linguistics*, 2008, vol. 20, no. 1, pp. 33–53.
42. Bol'shakova, E.I., Vasil'eva, N.E., and Morozov, S.S., Lexico-syntactic Patterns for Automatic Analysis of Scientific and Technical Texts, *Trudy 10-oi Natsional'noi konferentsii po iskusstvennomu intellektu (KII-2006)* (Proceedings of the 10th National Conference on Artificial Intelligence (AIC-2006)), Moscow: Fizmatlit, 2006, vol. 2, pp. 506–524 (In Russian.)
43. Sabirova, K. and Lukanin, A., Automatic Extraction of Hypernyms and Hyponyms from Russian Texts, *CEUR Workshop Proceedings*, 2014, vol. 1197, pp. 35–40.
44. Levy, O. and Goldberg, Y., Dependency-Based Word Embeddings, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Short Papers, Baltimore, Maryland, 2014, pp. 302–308.
45. Trofimov, I.V. and Suleymanova, E.A., A Dependency-based Distributional Semantic Model for Identifying Taxonomic Similarity, *Program Systems: Theory and Applications*, 2018, vol. 9, no. 4(39), pp. 443–460. (In Russian.)
46. Kulinich, A., Semiotic Models in Monitoring and Decision Support Systems, in *RCAI 2021: Artificial Intelligence*, Kovalev, S.M., Kuznetsov, S.O., and Panov, A.I., Eds., Lecture Notes in Computer Science, Cham: Springer, 2021, vol. 12948, pp. 303–316.
47. Hoffman, D.E., *The Oligarchs: Wealth and Power in the New Russia*, New York: PublicAffairs, 2011.
48. Korobov, M., Morphological Analyzer and Generator for Russian and Ukrainian Languages, *Analysis of Images, Social Networks and Texts*, 2015, pp. 320–332.

This paper was recommended for publication by V.G. Lebedev, a member of the Editorial Board.

Received March 1, 2022, and revised November 1, 2022.
Accepted November 22, 2022.

Author information

Kulinich, Alexander Alexeevich. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexkul@rambler.ru

Cite this paper

Kulinich, A.A., Application of Semiotic Models to Decision-Making. *Control Sciences* **5,** 34–50 (2022). http://doi.org/10.25728/cs.2022.5.4

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com