



ПРОБЛЕМИ ПРОГРАМУВАННЯ

НАУКОВИЙ ЖУРНАЛ

PROBLEMS
IN PROGRAMMING
SCIENTIFIC JOURNAL

2019
№ 4

Теми випуску:

- *Інструментальні засоби та середовища програмування*
- *Моделі та засоби систем баз даних і знань*
- *Експертні та інтелектуальні інформаційні системи*
- *Формальні методи розробки програмного забезпечення*
- *Прикладні засоби програмування та програмне забезпечення*

НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ПРОГРАМНИХ СИСТЕМ

ПРОБЛЕМИ ПРОГРАМУВАННЯ

науковий журнал

Головний редактор

Андон Пилип Іларіонович

академік НАН України,
директор Інституту програмних систем
НАН України

✉ Інститут програмних систем
НАН України
проспект Академіка Глушкова, 40, корп. 5
03187, Київ-187
☎ Тел. +380 (44) 526 5507
✉ E-mail: andon@isofts.kiev.ua
<http://www.pp.isofts.kiev.ua>

Редакційна колегія

Головний редактор

П.І. Андон (Україна)

Заступник

головного редактора

О.П. Ігнатенко (Україна)

Члени редколегії:

А.В. Анісімов	(Україна)	С.В. Пашко	(Україна)
О.С. Балабанов	(Україна)	А.М. Пелешишин	(Україна)
А.М. Глибовець	(Україна)	С.Д. Погорілий	(Україна)
М.М. Глибовець	(Україна)	О.І. Провотар	(Україна)
А.Ю. Дорошенко	(Україна)	І.В. Сергієнко	(Україна)
А. Корнілович	(Польща)	М.О. Сидоров	(Україна)
Н.М. Куусуль	(Україна)	І.П. Сініцин	(Україна)
Н.І. Недашківська	(Україна)	С.Ф. Теленик	(Україна)
М.С. Нікітченко	(Україна)	Л. Хлухі	(Словаччина)
В.В. Пасічник	(Україна)		

Адреса для кореспонденції

✉ Інститут програмних систем
НАН України
Проспект Академіка Глушкова, 40
03187, Київ-187

☎ Тел.: +380 (44) 526 5065
Факс: +380 (44) 526 6263
✉ E-mail: iss@isofts.kiev.ua

Затверджено до друку вченою радою Інституту програмних систем НАН України.
Протокол № 8 від 12.09.2019 р.

Редактор *В.П. Замула*
Комп'ютерна верстка *В.П. Замула*

Підписано до друку 28.11.2019. Формат 60x84/8. Папір офс. Ум. друк. арк. 15,11.
Обл.-вид. арк. 12,41. Тираж 120 прим. Ціна договірна. Замовл.

Віддруковано ВД «Академперіодика» НАН України
вул. Терещенківська, 4, м. Київ, 01004

Свідоцтво суб'єкта видавничої справи ДК № 544 від 27.07.2001

ПРОБЛЕМИ ПРОГРАМУВАННЯ

науковий журнал

№ 4

жовтень – грудень

2019

Заснований у березні 1999 р.

ЗМІСТ

Інструментальні засоби та середовища програмування

Rahozin Dmytro V. A resource limited parallel program model **3**

Моделі та засоби систем баз даних і знань

Рогошина Ю.В., Гришанова І.Ю. Використання онтологій для розширення можливостей пошуку та навігації в онлайн-версії «Великої української енциклопедії» **11**

Рогошина Ю.В. Використання тезаурусів для пошуку складних інформаційних об'єктів у Web на основі онтологій **28**

Експертні та інтелектуальні інформаційні системи

Захарова О. Класифікація метаданих великих даних **53**

Львіна О.П., Слабоспицька О.О. Підхід до побудови та використання моделі предметної області ризиків стратегічних рішень організації **75**

Формальні методи розробки програмного забезпечення

Мороз О.Г., Мороз Г.Б. Применение машинного обучения в программной инженерии: обзор **92**

Прикладні засоби програмування та програмне забезпечення

Григорян Р.Д., Дегода А.Г., Джуринський Е.А. Программный симулятор механизмов долговременной регуляции гемодинамики человека **111**

Кузьміна К.І., Сьомик Т.М., Шереметова Л.М. Інноваційна педагогічна комп'ютерна технологія оптимізації шкільного колективу в аспекті знань про біосоціальну структуру особистості його членів **121**

Свідоцтво про державну реєстрацію КВ № 7490 від 01.07.2003

Науковий журнал «Проблеми програмування» занесений до переліку наукових фахових видань України, в яких можуть публікуватися основні результати дисертаційних робіт.



PROBLEMS IN PROGRAMMING

scientific journal

№ 4

October – December

2019

Founded in March, 1999

CONTENTS

Software Environment and Tools

Rahozin Dmytro V. A resource limited parallel program model 3

Models and facilities for data and knowledge bases

Rogushina J., Grishanova I. Use of ontologies for search and navigation improvement in the online version of the “Great ukrainian encyclopedia” 11

Rogushina J. Use of thesauruses for search of complex information objects on Web on the basis of ontologies 28

Expert and intelligent information systems

Zakharova O. Big data metadata classification 53

Ilina E.P., Slabospitskaya O.O. An approach for problem area elaboration and usage of organization’s strategic decisions risks 75

Formal Methods for Software Delelopment

Moroz O.H., Moroz H.B. Application of machine learning in software engineering: an overview 92

Critical Systems Software

Grygoryan R.D., Degoda A.G., Dzhurinsky E.A. A simulator of mechanisms of long-term control of human hemodynamics 111

Kuzmina K.I., Syomik T.M., Sheremetova L.N. Innovative pedagogical computer technology optimization school community in the aspect of knowledge about the biosocial personality structure of its members 121

A RESOURCE LIMITED PARALLEL PROGRAM MODEL

Modern parallel programs run in a complex, resource-limited environment, and this raises the new requirements for resource consumption and execution stability of long running processes. In order to help with checking resource constraints for such parallel software a resource-limited parallel program formal model was developed. The model expresses the resource and time constraints and is suitable both for fine grained and coarse-grained parallelism in programs. For higher degrees of parallelism (at independent procedure level, bigger loop iterations, large computing blocks for graphics, video and neural network processing) the interpretation of formal model can be done in run-time and avoid dead locks and hangs during resource allocation. We are discussing several modern software frameworks that are able to integrate the functionality to interpret the model and check the feasibility of the set of parallel programs running on hardware simultaneously with resource and time limitations. Real world tasks – neural network inference, video processing, general purpose computing on GPU – which get benefits after enabling such models - are discussed.

Key words: formal model, parallel computing, parallel computing on graphics processing units.

Introduction

Nowadays parallel programming still is an activity, which cannot be planned well in terms of spent time and parallelization quality (i. e. writing a good code, which fits underlying architecture well in limited time and money). This can be easily explained by increasing complexity of underlying computing architecture of the modern hardware. The parallel architectures of 80ies had the minimal number of cache levels, simpler memory hierarchy and quite simple MESI cache protocols, but even in this case the performant implementation of basic BLAS/LAPACK procedures was a tricky thing. Today the state-of-art parallel computing is based on Graphics Processing Units (GPUs) which has quite complex memory hierarchy, complicated data exchange paths between central processors and GPUs and highly parallel SIMD-type computational units. The sophisticated tuning of GPU-based programs is mostly economically ineffective due to high labor cost and longer time-to-market. This involves the use of programming tools, which enable more high-level programming structures than just basic CUDA or OpenCL code.

The good examples of such tool are the neural network descriptions tools. Starting from Caffe tool [1], the inference and deep learning phases in neural networks are based on a high-level description of the network. The network therefore is defined as a pipeline of standard computing blocks (usually more

than 50 types of blocks are defined) and data paths between them. This definition format practically enables the development of a neural network from idea to optimized implementation without moving down to hand programming the neural network behavior. The success of Caffe lead other development groups to implement alternate or competitive neural network definition languages (Darknet [2]) or integrate a set of packages into programming languages and frameworks (TensorFlow [3]), sometimes incompatible. Of course, the use of high level definitions is not limited to neural network structure descriptions.

1. Practical expressions of high-level programming structures

Usually the simplified description of a big computational pipeline lacks ability for semi-precise optimization of resource use, as there is no way to estimate possible changes in computation pipeline which can use GPU resources more effectively or estimate how efficiently the neural network pipeline will run on GPU simultaneously with other applications. The optimizations of such computing pipeline are hard and strictly depend on underlying hardware architecture, which is a trade secret for the main hardware market players.

Other example is OpenCV framework [4] extensions for matrix/image operations on

GPUs. Program can simply move matrices or image operations to GPU without critical changes in code, but the GPU memory management is done by hand by moving the image data between central processor unit (CPU) and GPU memory. Such a helper (here helper mean the OpenCV functions set) does not deal with actual computational resource allocation on GPU, so the user cannot reach the best performance on GPU side without additional and expensive platform-dependent optimizations.

An interesting example is Gstreamer [5] media processing pipeline. Started as an eager media processing pipeline more than 30 years ago, now it is enriched with data processing on GPU side. A good example is Gstreamer-based DeepStream framework from Nvidia [6], which provides optimized neural network inference at GPU side. For this case the data processing pipeline may reach more than a hundred components, which are controlled by a dozen of threads. Despite of many efforts applied from Nvidia side, basic Gstreamer functionality still uses only rudimentary and hidden thread control and does not bother with resource allocation (either GPU memory or GPU computational power).

Looking through these three cases the can lighten the following flaws: 1) too high-level model for neural network description, where resource allocation (computational, memory, communication) is out of scope of the model; 2) too low-level model where the user should deal with resources by hand but not by describing them in model description; 3) even ignorance of resource allocation. All these flaws greatly affect the parallel software system which goes from proof of concept stage to customer environment (i. e. productization process) experiencing all computational, memory and communication resource constraints. Due to modern shift of heavy computations to GPU side, the productization process becomes a headache for engineers, full of bug hunting and code tricks to resolve various resource constraints. The problem appears more complex in case if applications are long-running, so the memory and other resource leaks can be fatal.

In order to overcome the difficulties of constraint management process we propose a model, which allows to introduce resource management process to frameworks such as OpenCV, TensorFlow or Gstreamer. It should be noted that these frameworks are different in levels of parallelism expression. OpenCV provides only basic parallelism, Gstreamer – thread-level and Caffe/Tensorflow – coarse-grain. Still we model will fit to all parallelism types.

There are a lot of previous work introducing various concepts for parallel software models, starting from Timed Finite Automata [7]. This article introduces resource constraints for parallel software models, which are necessary for executing parallel program in modern highly parallel and resource constrained environment.

2. A Model of Resource-Constrained Parallel System

The definition of the system is derived from [8], extending the previous definitions. The word “real-time” is not used intentionally, as the complex parallel system may have both hardware and resource constraints that prevent real-time behavior.

We consider a discrete-time model where the time is represented as a set of non-negative integer values denoted by \mathbb{N} . The time progress is measured by clocks, which are non-negative integer variables increased constantly by one for some time. If compared to [8] the definition “synchronously” is dropped, as in real life system usually a common reference clock is used. For the set of clocks \mathbf{X} , a *valuation* $v: \mathbf{X} \rightarrow \mathbb{N}$ is defined – it is a function associating with each clock its value $v(x)$. For a clocks subset $\mathbf{X}' \subseteq \mathbf{X}$ and a clock value $l \in \mathbb{N}$ we denote by $v(\mathbf{X}', l, x)$ the valuation that coincides with v for all clocks $x \in \mathbf{X} \setminus \mathbf{X}'$, and that associates l to all clocks $x \in \mathbf{X}'$. It is defined by:

$$v(\mathbf{X}', l, x) = \begin{cases} l & \text{if } x \in \mathbf{X}' \\ v(x) & \text{otherwise.} \end{cases}$$

Guards are used to specify when actions are enabled. Simple constrains over

clocks \mathbf{X} are considered. The grammar allows to build general constrains over clocks:

$$c := true \mid false \mid x \leq k \mid x < k \mid x \geq k \mid x > k \mid c \wedge c \mid c \vee c \mid \neg c.$$

The evaluation of a clock constraint c for a valuation v of clocks \mathbf{X} denoted by $c(v)$ is obtained by replacing each clock x by its value $v(x)$.

A *guard* g is a clock constraint c with an urgency type $\tau \in \{n, d, u\}$, denoted by $g = [c^\tau]$. Here urgency types are used to specify the need of the action in case if action execution is enabled (i. e. when the clocks constraint is true). Non-urgent actions are denoted by \mathbf{n} , delayable actions (which should be executed during their enable time interval) by \mathbf{d} , urgent actions (should be executed as soon as they are enabled) are denoted by \mathbf{u} . The predicate $urg[g]$ that characterizes the valuations of clocks for which the guard $g = [c^\tau]$ is *urgent* is defined by:

$$urg[g](v) \Leftrightarrow \begin{cases} false & \text{if } g \text{ is nonurgent } (\tau = n) \\ c(v) \wedge \neg c(v + K) & \text{if } g \text{ is delayable } (\tau = d) \\ c(v) & \text{if } g \text{ is urgent } (\tau = u) \end{cases}$$

The set of guards over the set of clocks \mathbf{X} is denoted as $\mathbf{G}(\mathbf{X})$.

For given guards

$$g_1 = [c_1]^{\tau_1} \text{ and } g_2 = [c_2]^{\tau_2},$$

the conjunction of g_1 and g_2 is denoted by $g_1 \wedge g_2$ and is defined by $g_1 \wedge g_2 = [c_1 \wedge c_2]^{max \tau_1, \tau_2}$, considering that urgency types are ordered as follows: $n < d < u$. So, for given guard $g = [c]^\tau$ and a valuation v , we also write $g(v)$ for the expression $c(v)$.

Additionally, the resource constraints are defined.

Additionally to model in [8] let us define the resource set $\mathbf{P} = \{p_i^t\}$, where t is the resource type, I is the resource index for some number of identical resources in computational system. Also the resource p_i^t may be allocated in part $-p_i^t(k)$, where k is the amount of resource p_i^t .

Let us define availability function for resource: $avl(p_i^t(k))$, is it *true* is case if the resource is available.

3. Abstract model with constraints

Definition 1. Abstract model with constrains.

An *abstract model* is a timed automaton $M = (\mathbf{A}, \mathbf{Q}, \mathbf{X}, \mathbf{P}, \rightarrow)$ such that:

- \mathbf{A} is a finite set of (*observable*) actions. In addition to actions \mathbf{A} internal actions β . The set of actions $\mathbf{A} \cup \{\beta\}$ is denoted as \mathbf{A}^β ;
 - \mathbf{Q} is a finite set of control locations;
 - \mathbf{P} is the resource set;
 - \mathbf{X} is finite set of clocks
- $\rightarrow \subseteq \mathbf{Q} \times (\mathbf{A}^\beta \times \mathbf{G}(\mathbf{X}) \times 2^{\mathbf{X}}) \times \mathbf{Q}$ is a finite set of labeled transitions. A transition is a tuple (q, a, g, r, p, q') where a is an action executed by the transition, g is guard over \mathbf{X} , r is a subset of clocks that are reset by the transition and p is the resource necessary to be allocated during the transaction. We write $q \xrightarrow{a, g, r, p} q'$ for $(q, a, g, r, p, q') \in \rightarrow$.

An abstract model describes the platform-independent behavior of the system.

Definition 2. (Abstract model semantics). An abstract model $M = (\mathbf{A}, \mathbf{Q}, \mathbf{X}, \mathbf{P}, \rightarrow)$ defines a transition system *TS*. States of *TS* are pairs (q, v) , where q is a control location of M and v is a valuation of the clocks \mathbf{X} .

Actions: We have $(q, v) \xrightarrow{a} (q', v[r \mapsto 0])$ if $q \xrightarrow{a, g, r, p} q'$ in M and $g(v)$ is true and $avl(p)$ is true.

Time steps. For a waiting time $\delta \in \mathbb{N}, \delta > 0$, we have $(q, v) \xrightarrow{\delta} (q, v + \delta)$ if for all transitions

$$(q \xrightarrow{a, g, r} q' \text{ of } M \text{ for for all } \delta' \in [0, \delta[, \neg urg[g](v + \delta')).$$

Urgency corresponds to priorities induced by the timing constraints: urgent transitions have priority compared to other possible transitions. We denote by $wait(q, v)$ the maximal waiting time allowed at (q, v) . Always $wait(q, v + \delta) = wait(q, v) - \delta$ for all $\delta \in$

$\in [0, wait(q, v)]$, and is formally defined as follows:

$$\begin{aligned} & wait(q, v) = \\ & = \min \left(\left\{ \delta \geq 0 \mid \bigvee_{q \xrightarrow{a_i, g_i, r_i, p_i} q_i} urg[g_i](v + \delta) \right\} \cup \right. \\ & \left. \cup \{+\infty\} \right). \end{aligned}$$

For an abstract model $M = (\mathbf{A}, \mathbf{Q}, \{x\}, \mathbf{P}, \rightarrow)$, a finite execution sequence of M from an initial state (q_0, v_0) is a maximal sequence of observable actions and time-steps $(q_i, v_i) \rightsquigarrow^{\sigma_i} (q_{i+1}, v_{i+1})$, $\sigma_i \in \mathbf{A} \cup \mathbf{N}$ and $i \in \{0, 1, 2, \dots, n\}$ ($i \in \mathbb{N}$), such that \rightsquigarrow is the transitive closure of \rightarrow for β -transitions, that is $(q_i, v_i) \rightsquigarrow^{\sigma_i} (q_{i+1}, v_{i+1})$ if $(q_i, v_i) \xrightarrow{\beta^*} (q'_i, v'_i) \xrightarrow{\sigma_i} (q''_i, v''_i) \xrightarrow{\beta^*} (q_{i+1}, v_{i+1})$.

For example of this model you can refer to Model 1 in [8].

Definition 3. (Composition of abstract models). Let $M_i = (\mathbf{A}_i, \mathbf{Q}_i, \mathbf{X}_i, \mathbf{P}_i, \rightarrow_i)$, $1 \leq i \leq n$, be a set of abstract models. We assume that their sets of action and clocks are disjoint, i.e. for all $i \neq j$ we have $\mathbf{A}_i \cap \mathbf{A}_j = \emptyset$ and $\mathbf{X}_i \cap \mathbf{X}_j = \emptyset$. A set of interactions γ is a subset of 2^A , where $A = \bigcup_{i=1}^n A_i$, such that any interaction $a \in \gamma$ contains at most one action of each component M_i , that is, $a = \{a_i \mid i \in I\}$ where $a_i \in A_i$ and $I \subseteq \{1, 2, \dots, n\}$. The composition of the abstract models M_i , $1 \leq i \leq n$, by using a set of interactions γ , denoted by $M = (\gamma, \mathbf{Q}, \mathbf{X}, \mathbf{P}, \rightarrow)$, is the composite abstract model $M = (\gamma, \mathbf{Q}, \mathbf{X}, \mathbf{P}, \rightarrow)$ such that $\mathbf{Q} = \mathbf{Q}_1 \times \mathbf{Q}_2 \times \dots \times \mathbf{Q}_n$, $X = \bigcup_{i=1}^n X_i$ and \rightarrow_γ is defined by the rules:

$$\begin{aligned} & a = \{a_i\}_{i \in I} \in \gamma \\ & g = \bigwedge_{i \in I} g_i \\ & r = \bigcup_{i \in I} r_i \\ & \forall i \in I, q_i \xrightarrow{a_i, g_i, r_i, p_i} q'_i \\ & \forall i \notin I, q'_i = q_i \\ & (q_1, \dots, q_n) \xrightarrow{(a, g, r, p)_\gamma} (q'_1, \dots, q'_n) \\ & \text{also } \exists i \in \{1, \dots, n\}. q_i \xrightarrow{\beta, g_i, r_i, p_i} q'_i \end{aligned}$$

$$\forall i \neq j. q'_j = q_j$$

$$(q_1, \dots, q_n) \xrightarrow{\beta, g_i, r_i, p_i}_\gamma (q'_1, \dots, q'_n).$$

A composition $M = \gamma(M_1, \dots, M_n)$ of abstract models M_i , $1 \leq i \leq n$, can execute two types of transitions: interactions $a = \{a_i\}_{i \in I} \in \gamma$ which corresponds to synchronizations of actions a_i of models M_i , $i \in I$, and internal actions β of the models M_i . An interaction $a = \{a_i\}_{i \in I} \in \gamma$ is enabled from a state of M if all actions a_i are enabled.

In a composite model $M = \gamma(M_1, \dots, M_n)$ many interaction can be enabled to act simultaneously (in the same time) introducing a degree of non-determinism in the behavior of M .

In order to restrict non-determinism, *priorities* are introduced that specify which interaction should be executed among the enabled ones. A priority on $M = \gamma(M_1, \dots, M_n)$ is a relation $\pi \subseteq \gamma \times Q \times \gamma$ such that for all q the relation $\pi_q = \{(a, a') \mid (a, q, a') \in \pi\}$ is a partial order. We write $a \pi_q a'$ for $(a, q, a') \in \pi$ to express the fact that a has weaker priority than a' at state q . That is if both a and a' are enabled at state q , only the action a' can be executed. Thus, priority $a \pi_q a'$ is applied only when the conjunction of the guards and resources of a and a' is true. Let $q \xrightarrow{a, g, r, p}_\gamma q'$ and $q \xrightarrow{a', g', r', p'}_\gamma q''$ be transitions of M such that $g = [c]^\tau$ and $g' = [c']^\tau$. Applying priority $a \pi_q a'$ boils down to transforming the guard g of a into the guard $g_\pi = [c \wedge \neg c']^\tau$ and leaving the guard g' of a' unchanged.

Furthermore we denote by $en_q(a)$ the predicate characterizing valuations of clocks for which an interaction a is enabled at state q . It is defined by:

$$en_q(a) = \begin{cases} \text{false, if } \nexists (q, a, g, r, p, g') \in \rightarrow_\gamma \\ \bigvee_{(q, a, [c]^\tau, r, p, q') \in \rightarrow_\gamma} c - \text{otherwise.} \end{cases}$$

Definition 4. Priority. Given a composite model $M = (\gamma, \mathbf{Q}, \mathbf{X}, \mathbf{P}, \rightarrow_\gamma)$ the application of priority π to M defines a new model $\pi M = (\gamma, \mathbf{Q}, \mathbf{X}, \mathbf{P}, \rightarrow_\pi)$ such that \rightarrow_π is defined by the rule:

$$q \xrightarrow{a, g, r, p}_\gamma q', g = [c]^\tau,$$

$$g_{\pi} = \left[c \wedge \neg \bigvee_{a \in \pi} en_q(a') \right]^{\tau}$$

$$q \xrightarrow{a, g_{\pi}, r, p} \perp q'$$

Example of an abstract model with priorities is considered in [8].

Abstract models are platform-independent representations of programs with atomic and instantaneous actions execution. Real (“physical”) models represent the program behavior on a real platform. It accounts the fact that the action execution takes some non-zero time. So we need to break action execution atomicity and introduce execution times. The transition of an action a of an abstract model is replaced by a sequence of two consecutive transitions of the corresponding physical (real world) model – see figure 1. The first transition marks the beginning of the execution of action a , and the second transition marks its completion. These transitions are separated by a partial state denoted by \perp . The execution time of the action corresponds to the waiting time at state \perp

$$q \xrightarrow{a, g, r, p} q' \xrightarrow{\perp_t} \mathbf{q} \xrightarrow{a, g, r, p} \perp_t \xrightarrow{\beta} q'$$

(Corresponding sequence of transitions in M^{\perp}), where

$$t = (q, a, g, r, p, q') \text{ in } M.$$

This denoted the transformation of transitions of the abstract model.

Definiton 5. Physical model. Let $M=(\mathbf{A}, \mathbf{Q}, \mathbf{X}, \mathbf{P}, \rightarrow)$ be an abstract model. We define the associated as the timed automaton $M^{\perp}=(\mathbf{A}, \mathbf{Q} \cup \mathbf{Q}^{\perp}, \mathbf{X}, \mathbf{P}, \rightarrow_{\perp})$ such that:

\mathbf{Q}^{\perp} is the set of partial states such that there is one partial state for each transition of M , that is, $\mathbf{Q}^{\perp}=\{\perp_t \mid t \in \rightarrow\}$

\rightarrow_{\perp} is defined by the rule:

$$\frac{q \xrightarrow{a, g, r, p} q' \quad t=(q, a, g, r, p, q')}{q \xrightarrow{a, g, r, p} \perp_t \quad \perp_t \xrightarrow{\beta, [\text{true}]^1, \emptyset, \text{true}} \perp_t q'}$$

In the physical model M^{\perp} we assume arbitrary execution times for actions, ranging from 0 to $+\infty$, which is modeled by the guard $[\text{true}]^1$ for β -transitions. Notice that M^{\perp} can be further constrained if bounds of the execution

times of actions are unknown. For instance, if an estimate $WCET(a)$ is known for the worst-case execution time of an action a , the associated timing constraint is $[x_a \leq WCET(a)]^d$ instead of $[\text{true}]^1$, where x_a is a clock that is reset whenever a is started. This allows us to statically check the correctness of the application running on the platform but this is beyond the paper scope.

In a physical model M^{\perp} , the execution of the action a by a transition $t=(q, a, g, r, p, q')$ is followed by a lapse of time $\delta(a) \in \mathbb{N}$ at the partial state \perp_t before a β -transition is executed:

$$(q, v) \rightsquigarrow^a (\perp_t, v[r \mapsto 0]) \rightsquigarrow^{\delta(a)} \rightsquigarrow^{\delta(a)} (q', v[r \mapsto 0] + \delta(a)). \quad (1)$$

This corresponds to the following execution sequence in the abstract model M , if such a sequence is feasible:

$$(q, v) \rightsquigarrow^a (q', v[r \mapsto 0]) \rightsquigarrow^{\delta(a)} \rightsquigarrow^{\delta(a)} (q', v[r \mapsto 0] + \delta(a)). \quad (2)$$

It should be noticed that the time stamp $\delta(a)$ of M^{\perp} in (1) may not be the time stamp of M in (2) if $\delta(a) > \text{wait}(q', v[r \mapsto 0])$, meaning that the physical model violates timing constraints defined in the corresponding abstract model. In this case we say that the considered execution sequence is not *time-safe*. (The execution times of abstract and physical models are compared in [9] – considering that if all execution sequences of M^{\perp} are time-safe than M^{\perp} is weakly simulated by M).

Correct model implementation should execute only time-safe sequences, but time-safety violations occur in a physical model when the execution time of an action is larger than what is allowed by the timing and resource constraints of the corresponding abstract model. Correct implementations are obtained for platforms that are sufficiently fast for executing the program without violating time-safety. Here the physical model preserves the semantic of the abstract model as shown in [10]. Otherwise the time-safety violations should be checked in run time.

Physical model composition considerations and correctness considerations can be checked in [8].

So how we can deal with resource limitations? State-of-the-art software packages require simpler solutions, so model interpretation in run-time (such as in [11]) looks complex and superfluous. In case if program is already expressed as a graph of scheduled blocks, it is possible to evaluate model requirements and behavior at the moments between the previous block finish and a the new block start – this also works for moments of spawning new parallel processes and joining parallel processes for one new serial process. This works well for cases enlisted before: LAPACK-based computations [12], neural network inference and media processing in Gstreamer-like pipelines. Even it is possible for interpreted code, such as Java or .Net, where code assemblies can be annotated with resource information.

4. Affected industry cases

Why all these model descriptions are important for us although all they looks to be abstract for real software business?

Before 2005 the parallel programming was the area of academy pundits and small groups of professionals in computer graphics area. After 2005 the integrated circuits making technology enabled so many transistors on single die for hardware engineers so that the single computing units (up to von Neumann definitions) can not effectively use the hardware. The most efficient way was to place several processing units on one silicone crystal, so that the silicone can run multiple processes simultaneously somehow. It looked like “as single CPU can not deal effectively, let’s do multiple CPUs and the programmer should do software in right way”. At practical side a lot of money was invested into teaching parallel programming in colleges and making specialized computer languages for highly parallel hardware – such as Nvidia CUDA [13] or anti-Nvidia industry standard OpenCL [14]. As the efficient parallel software requires more time to invest, more skilled resources and more money for test teams work the new tendency was appeared – to make frameworks which allow the programmer to make parallel programs with simpler (less or more simpler) descriptions. Such concept looks nice but works hard. Two well-known

technology examples – CUDA and OpenCL shows that the main model of writing a parallel program for GPUs is a “producer-consumer” model when a CPU-side program controls GPU threads execution and none of GPU threads are self-sufficient. Practically all memory management employs CPU-GPU memory communications, and memory communications between neighbor GPU cards are exceptions.

Under these conditions any software developer experiences a serious technology limitation, as any parallel program support only its exclusive execution on GPU resource and the resource pool subdivision between different processes are possible only in case if each parallel program allocates some number of GPUs, leaving the other GPUs to counterpart.

However, all technologies similar to CUDA/OpenCL or any specialized parallel programming languages do not decrease the development time significantly. The good ways to decrease programmer effort are infrastructures, which shorten development time for 90 % in simpler cases and for 50 % in hard cases. The important issue is that the most of the infrastructures are upgradable to incorporate the described physical model. Let us check several infrastructure (framework) cases.

1) TensorFlow [3]. At least 60 % of neural network learning and inference market. Tensorflow interprets network model (using Python language) based on a sequence of big code blocks.

2) Caffee[1]. Near 30 % of neural networking segment, written in optimized C++. Caffee (conceptually similar to Tensorflow) interprets an abstract sequence of blocks.

3) Gstreamer [5] and ffmpeg [15] media processing pipelines. Both construct a pipeline using already defined big code blocks (plugins), including spawning parallel processes and internal queue storage. Pipelines also works for GPU-based computations. Both packages are widely used as a base for industry media processing and broadcasting.

4) OpenCV framework [4]. It is a good base for parallel matrix computations

(and 2D/3D processing) at GPU. Still the OpenCV is more low-level library tool than a framework, the packages can benefit from model interpretation.

5) Simulation tools, starting from the old good NS2 (good use example in [16]) and other network simulations. In case of big runs cluster-based modeling benefit from model interpretations.

The proposed abstract and physical models are good universal tools for many frameworks. It can deal with both low level descriptions of CPU-handled subroutines (at 50-100 CPU instructions level) and high-level annotations for framework elements: all timing and resource constraints remain the same.

The next step will be more practical: incorporation of the model interpreter into one of the frameworks and practical test of benefits got because of expressing the software in term of physical model.

Conclusions

In order to meet the modern requirements for software development – less time, more quality, lower expenses – this article proposes the resource constrained model for parallel programs, which allows to run (or model the behavior) of multiple (and different) parallel software runs under resource constraints on real-world hardware systems. In addition, the list of popular frameworks - which can benefit from incorporating the elements of the resource-constrained models interpreter – is presented. The future work includes the extension of one of framework with model interpreter for low-overhead resource checker on the fly (at program run time) and real-world model examples.

References

1. Yangqing J., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T. (2014) Caffe: Convolutional Architecture for Fast Feature Embedding. ArXiv preprint: arXiv:1408.5039
2. Redmon J. (2013) [Online]. Darknet: Open Source Neural Networks in C. – Available from <https://pjreddie.com/darknet/>
3. Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M. & others (2016). TensorFlow: A System for Large-Scale Machine Learning. *OSDI*. P. 265–283.
4. Bradsky G., Kaehler A. Learning OpenCV — O’Reilly, 2008. P. 1.
5. Taymans W., Baker S., Wingo A. (2018) GStreamer Application Development 1.10.1. P. 164. 12th Media Services.
6. DeepStream [Online] – Nvidia DeepStream Software Development Kit – Available at <https://developer.nvidia.com/deepstream-sdk>
7. Peter Hui and Satish Chikkagoudar. (2012) A Formal Model for Real-time Parallel computation. In Proc of FTSCS-2012. P. 39–53.
8. Ahlem Triki, Jacques Combaz. (2013) Model-Based implementation of Parallel Real-Time Systems. Verimag Research Report TR-2013-11
9. Wilhelm R., Altmeyer S., Burguiere C., Grund D., Herter J., Reineke J., Wachter B., Wilhelm S. Static timing analysis for hard real-time systems. In Barthe G. and Hermenegildo M.V., eds., *WMCAI*. 2010. Vol. 5944 of LNCS. P. 3–22. Springer.
10. Abdellatif T., Combaz J., Sifakis J. Model-based implementation of real-time applications. In Carloni L.P. and Stavros Tripakis, eds. *EMSOFT*. 2010. P. 229–238.
11. Basu A., Bogza M., Sifakis J. Modeling heterogeneous real-time components in BIP. In *SEFM*. 2006. P. 3–12. IEEE Computer Society.
12. Baboulin M., Demmel J., Dongarra J., Tomov S., and Volkov V. Enhancing the Performance of Dense Linear Algebra Solvers on GPUs (in the MAGMA Project), Austin, TX, The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC08), Nov. 2008.
13. CUDA [Online] – Available at <https://developer.nvidia.com/cuda-zone>
14. OpenCL [Online] – Available at Khronos Group: <https://www.khronos.org/opencl/>
15. Xu Y.G. and Cao S.X. Real-Time Video Acquisition and Frame Compression Processing Technology Based on FFmpeg, Applied Mechanics and Materials. 2014. Vols. 631–632. P. 494–497.
16. Michael Welzl. Adaptive Multimedia Communication over Satellite Routed IP". In ICC 2000 (International Conference on Communications – IEEE Communications Society), New Orleans, Louisiana, USA, 18–22 June 2000.

Література

1. Yangqing J., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T. (2014) Caffe: Convolutional Architecture for Face Feature Embedding. ArXiv preprint: arXiv:1408.5039
2. Redmon J. (2013) [Online]. Darknet: Open Source Neural Networks in C. – Available from <https://pjreddie.com/darknet/>
3. Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M. & others (2016). TensorFlow: A System for Large-Scale Machine Learning. *OSDI*. P. 265–283.
4. Bradsky G., Kaehler A. Learning OpenCV — O'Reilly, 2008. P. 1.
5. Taymans W., Baker S., Wingo A. (2018) GStreamer Application Development 1.10.1. P. 164. 12th Media Services.
6. DeepStream [Online] – Nvidia DeepStream Software Development Kit – Available at <https://developer.nvidia.com/deepstream-sdk>
7. Peter Hui and Satish Chikkagoudar. (2012) A Formal Model for Real-time Parallel computation. In Proc of FTSCS-2012. P. 39–53.
8. Ahlem Triki, Jacques Combaz. (2013) Model-Based implementation of Parallel Real-Time Systems. Verimag Research Report TR-2013-11
9. Wilhelm R., Altmeyer S., Burguiere C., Grund D., Herter J., Reineke J., Wachter B., Wilhelm S. Static timing analysis for hard real-time systems. In Barthe G. and Hermenegildo M.V., eds., *WMCAI*. 2010. Vol. 5944 of LNCS. P. 3–22. Springer.
10. Abdellatif T., Combaz J., Sifakis J. Model-based implementation of real-time applications. In Carloni L.P. and Stavros Tripakis, eds. *EMSOFT*. 2010. P. 229–238.
11. Basu A., Bogza M., Sifakis J. Modeling heterogeneous real-time components in BIP. In *SEFM*. 2006. P. 3–12. IEEE Computer Society.
12. Baboulin M., Demmel J., Dongarra J., Tomov S., and Volkov V. Enhancing the Performance of Dense Linear Algebra Solvers on GPUs (in the MAGMA Project), Austin, TX, The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC08), Nov. 2008.
13. CUDA [Online] – Available at <https://developer.nvidia.com/cuda-zone>
14. OpenCL [Online] – Available at Khronos Group: <https://www.khronos.org/opencv/>
15. Xu Y.G. and Cao S.X. Real-Time Video Acquisition and Frame Compression Processing Technology Based on FFmpeg, Applied Mechanics and Materials. 2014. Vols. 631–632. P. 494–497.
16. Michael Welzl. Adaptive Multimedia Communication over Satellite Routed IP". In ICC 2000 (International Conference on Communications – IEEE Communications Society), New Orleans, Louisiana, USA, 18–22 June 2000.

Received 01.10.2019

About the author:

Dmytro V. Rahozin,
candidate of tech. sciences (PhD)
More than 10 publication in Ukrainian and
foreign journals.
<https://orcid.org/0000-0002-8445-9921>

Affiliation:

Institute of Software Systems,
NAS of Ukraine
03187, Kyiv-187,
Acad. Hlushkov avenue, 40.
Tel.: +38 068 575 91 25.
E-mail: dmytro.rahozin@gmail.com

ВИКОРИСТАННЯ ОНТОЛОГІЙ ДЛЯ РОЗШИРЕННЯ МОЖЛИВОСТЕЙ ПОШУКУ ТА НАВІГАЦІЇ В ОНЛАЙН-ВЕРСІЇ «ВЕЛИКОЇ УКРАЇНСЬКОЇ ЕНЦИКЛОПЕДІЇ»

Ціль роботи – аналіз засобів семантичного пошуку у Wiki-ресурсах та розробка напрямків розширення його можливостей на основі використання сучасних засобів керування знаннями. В статті аналізуються переваги, які забезпечує користувачам використання семантичної розмітки в інформаційних ресурсах Web для більш зручного та швидкого доступу до контенту. Розглядаються способи використання онтологій для пошуку та навігації у семантизованих Wiki-ресурсах. Наведено визначення Wiki-онтології, що формалізує відомості щодо структури бази знань інформаційного ресурсу (як звичайного, так і семантизованого), який базується на технології Wiki. Проаналізовано теоретичні засади визначення міри семантичної подібності між поняттями на основі онтологій та таксономій. Запропоновано методи побудови засобів навігації у Wiki-ресурсах, які базуються на знаходженні сторінок, що описують змістовно близькі поняття. Продемонстровано реалізацію запропонованого підходу в онлайн-версії «Великої української енциклопедії» (e-BUE), що забезпечує розширення функціоналу енциклопедичного видання шляхом надання засобів доступу до контенту на семантичному рівні. Обґрунтовано методологічні принципи організації бази знань e-BUE та категоризації її гасел, що забезпечують використання оцінок семантичної близькості між поняттями для пошуку подібних гасел.

Ключові слова: семантичні Wiki-ресурси, семантичний пошук, онтологія, семантична близькість, Велика українська енциклопедія.

Вступ

Ефективність використання Web-ресурсів значним чином визначається ступенем їх інтелектуалізації – наявністю семантичної розмітки, можливістю пошуку за змістом, використанням онтологій для моделювання структури бази знань (БЗ) тощо. Це впливає на функціонал інформаційного ресурсу (ІР) та забезпечує можливість використання його контенту іншими інформаційними системами (ІС), що пов'язані з тією самою предметною областю (ПрО).

Значення семантизації збільшується для ІР із складною структурою, гетерогенним контентом та великим обсягом даних, що викликає потребу в застосуванні сучасних технологій керування розподіленими знаннями. Одна з таких поширених на сьогодні технологій є Wiki, яка забезпечує спільне створення та використання інформації у Web.

Інтелектуалізація ІР, що будується на основі Wiki-технології, має свою структуру. В першу чергу вона базується на

семантизації ресурсу, тобто на встановленні змістовних зв'язків між Wiki-сторінками. Такі можливості надає, наприклад, семантичне розширення Wiki – Semantic MediaWiki (SMW) [1], що підтримує використання як семантичних властивостей у розмітці, які визначають зміст відношень між сторінкою ІР та іншими сторінками та даними, так і виконання семантичних запитів, умовами яких можуть бути значення таких властивостей. Уся сукупність семантичних властивостей ІР та їх значень може бути формалізована за допомогою Wiki-онтології ресурсу.

Основні характеристики онлайн-версії «Великої української енциклопедії» (e-BUE ue.gov.ua) дозволяють віднести її саме до таких ІР: ця енциклопедія містить велику кількість гасел з різних галузей знань з різноманітними наборами структурних елементів та мультимедійного ілюстративного матеріалу (рисунки, фотографії, відео, аудіо, карти, діаграми), між якими встановлюються різноманітні відношення.

Використання елементів онтологічного аналізу для створення структури БЗ е-ВУЕ обумовлено високою складністю та гетерогенністю знань, представлених у цьому інформаційному ресурсі: виразна здатність вбудованих можливостей що використовується як основа для створення порталу і підтримує такі механізми, як категорії, семантичні властивості та семантичні запити, достатня для реалізації такої системи знань, але не для її дослідження, вдосконалення та формалізації всіх різноманітних аспектів та характеристик.

Тому цей IP використовується в даній роботі для демонстрації можливостей розширення пошуку та навігації з використанням онтологічних знань. Онтологія ПрО може використовуватися як основа для знаходження семантично пов'язаних гасел е-ВУЕ, що надає ще один засіб навігації в IP: користувач отримує можливість перейти до тих понять ПрО, які змістовно пов'язані з тим гаслом, яке він розглядає.

Щоб автоматизувати використання розподіленого IP, виникає необхідність у побудові онтології його БЗ – формалізованої моделі, що дозволяє фіксувати характеристики його елементів, їх зв'язків, властивостей та відношень у формі, придатній для автоматичного оброблення, логічного виведення та аналізу. Така онтологія може бути основою для семантичної розмітки IP, яка уможлиблює інтелектуалізацію засобів навігації та пошуку відомостей на порталі, задовольняючи інформаційні потреби користувачів із урахуванням їх семантики.

Одним з напрямків використання Wiki-онтології IP є знаходження семантично близьких Wiki-сторінок: це дозволяє створити у ресурсі додаткові засоби навігації для швидкого доступу до сторінок зі спорідненим контентом. При цьому семантична близькість може визначатися відповідно до поточних потреб користувача.

Постановка задачі

Ціль даної роботи – аналіз засобів семантичного пошуку у Wiki-ресурсах та напрямків розширення його можливостей

на основі використання онтологій. Надається теоретичне обґрунтування методологічних принципів використання міри семантичної близькості між поняттями онтології, що формалізує семантичну розмітку е-ВУЕ, для розширення функціоналу цього енциклопедичного видання шляхом створення засобів навігації між сторінками на змістовному рівні.

Онтологічне подання знань

Онтології – БЗ спеціального виду, яка містить семантичну інформацію [2] з певної ПрО – застосовуються для подання знань різноманітних ПрО в Web-орієнтованих ІС [3]. Моделі онтологій зазвичай містять *поняття* (класи та екземпляри), *властивості* понять (атрибути, ролі), *відношення* між поняттями (залежності, функції) та додаткові *обмеження*, що визначаються аксіомами [4].

На формальному рівні онтологія – це система, що складається з наборів понять і тверджень про ці поняття, на основі яких можна будувати класи, об'єкти, відношення, функції та теорії.

Формальна модель онтології O – це впорядкована четвірка

$$O = \langle T, P, R, F \rangle,$$

де T – скінченна множина концептів (термінів) ПрО, яку описує онтологія O ; P – скінченна множина властивостей концептів; R – скінченна множина відношень між концептами заданої ПрО; F – скінченна множина аксіом, заданих на концептах, властивостях і/або відношеннях онтології O . Відношення представляють тип взаємодії між концептами ПрО. Приклад бінарного відношення – "є частиною". Аксіоми використовуються для моделювання тверджень, що завжди є істинними.

Wiki-онтологія – це окремих випадок комп'ютерної онтології, що відображає особливості технології Wiki та її семантичного розширення. Її виразні можливості обмежені в області використання аксіом ПрО та не припускають застосування характеристик для об'єктних властивостей та властивостей даних. Така онтологія

може бути побудована автоматизовано за семантично розміченим Wiki-ресурсом (набором Wiki сторінок, що містять семантичну розмітку) [5], але на практиці зазвичай спочатку розробляють Wiki-онтологію, а вже на її основі здійснюється реалізація самого Wiki-ресурсу [6].

Можна вважати (з певним рівнем спрощення), що Wiki-онтологія та структура БЗ Wiki-ресурсу є різними формами подання тієї самої системи знань. Така онтологія містить тільки ті знання, які можна безпосередньо здобути із семантичної розмітки. Тому в цій онтології відсутні, приміром, такі характеристики класів та властивостей, як еквівалентність, відсутність перетину тощо.

Wiki-онтологія конкретизує основні компоненти онтологічного подання знань $O = \langle X, R, F, T \rangle$ відповідно до технологій Semantic Web [7], розроблених W3C.

Для не семантизованого Wiki-ресурсу відповідна онтологія $O_{\text{wiki_no_semant}}$ містить наступні компоненти:

- $X = X_{\text{cl}} \cup X_{\text{ind}}$ – множина понять онтології, де X_{cl} – множина класів, що співпадає з множиною категорій Wiki-ресурсу P_{categ} ; X_{ind} – множина екземплярів класів – це об'єднання множин Wiki-сторінок: P_{user} – множина сторінок, створених користувачами, P_{template} – множина сторінок, що описують шаблони, P_{spec} – множина інших спеціальних сторінок;

- $R = L \cup \{r_{\text{ier_cl}}\}$ – множина відношень між елементами онтології, де $L = \{\text{"link"}\}$ – множина з одного елемента, що описує посилання однієї Wiki-сторінки цього ресурсу на іншу Wiki-сторінку цього ресурсу (посилання на інші види сторінок у рамках даної моделі не враховуються); $r_{\text{ier_cl}}$ – ієрархічне відношення між категоріями Wiki-ресурсу, яке визначається в процесі створення нових категорій;

- $F = \{f_{\text{equ}}\}$ – множина тих характеристик, що можуть використовуватися

для логічного виведення над онтологією, для Wiki-онтології обмежується єдиним відношенням еквівалентності між Wiki-сторінками (відсильні статті);

- інші елементи онтологічної моделі для цієї Wiki-онтології є порожніми множинами.

Формальна модель семантично збагачених Wiki-ресурсів є більш складною і включає ряд елементів, пов'язаних із семантичними властивостями. Онтологія $O_{\text{wiki_semant}}$ містить більше компонентів:

- множина відношень доповнюється $L_{\text{sem_prop}}$, елементи якого – семантичні властивості Wiki-сторінок, областю значення яких є Wiki-сторінки;

- T – множина типів даних (наприклад, рядок, ціле), значення з яких можуть приймати властивості даних класів онтології, – значення семантичних властивостей, що пов'язують не з іншими сторінками, а зі значеннями даних.

Шаблони типових ІО дозволяють визначати область значення та область визначення об'єктних властивостей Wiki-онтології, та навпаки, характеристики об'єктних властивостей з Wiki-онтології є основою для коректного створення шаблонів типових ІО. Семантична розмітка Wiki-ресурсу дозволяє автоматизувати процес побудови його Wiki-онтології, і тому для ІР на основі Semantic MediaWiki досить просто застосовувати семантичні технології, що базуються на використанні онтологічних знань.

Семантичний пошук на порталі e-VUE

В ефективності використання сучасних розподілених баз знань одним з визначних чинників є організація пошуку, що дозволяє користувачеві отримати доступ до потрібної інформації. При цьому важливі як час доступу, так і зрозумілість і зручність користувацького інтерфейсу. Знання, представлені інформаційними ресурсами, що організовані на основі Wiki-технологій, мають певну специфіку [8].

Організація подання та збереження інформації на таких ресурсах визначає засоби та методи пошуку. Відомості, що представлені за допомогою Wiki-сторінок, з точки зору інформаційно-пошукових систем відносяться до частково структурованих: вони групуються за допомогою категорій Wiki, зазвичай містять посилання на інші Wiki-сторінки того самого ресурсу, та, якщо застосовуються семантичні Wiki, включають семантичну розмітку, тобто властивості інформації, що представлена на сторінці, охарактеризовані на семантичному рівні.

На сторінці категорії у Wiki-ресурсах представлено перелік її підкатегорій та усіх сторінок, що відносяться до цієї категорії. Це забезпечує досить зручну навігацію у ресурсі за змістовно поєднаними у категорії групами сторінок, але не дозволяє отримувати доступ до інформації, що відноситься до групи категорій.

Засоби пошуку у Wiki-ресурсах можна класифікувати на основні наступні групи:

- за *ключовими словами* (для енциклопедій найчастіше – за гаслом статті або за початковими літерами назви);
- за *тематикою предметної області* (для енциклопедій – за категоріями та підкатегоріями статті);
- за *типом інформаційного об'єкта* (для енциклопедій типи ІО теж формалізуються через апарат категорій, конкретні значення яких здобуваються із узагальнення типів статей енциклопедії);
- за *семантикою інформаційного об'єкта* (за його властивостями).

Перший варіант забезпечує найбільш швидкий доступ до інформації, другий базується на загальноновживаних класифікаторах (що дозволяє формалізувати класифікацію, але ускладнює пошук для пересічних користувачів), а третій – враховує семантику інформаційної потреби користувача та дозволяє ознайомитися із семантично близькими статтями.

Пошук за ключовими словами – назвою гасла, словами в природномовному контенті – є традиційним, його реалізують

практично всі електронні довідкові та енциклопедійні видання, але він потребує від користувача знань щодо точного найменування того, що він шукає. Пошук термінів, що вводяться користувачем з помилками, зазвичай не підтримується. Крім того, для такого пошуку має значення порядок слів у пошуковому рядку.

Пошук за тематикою є більш гнучким. Для е-ВУЕ найбільш вживаним є пошук за галузями знань та їх підкласами. Крім того, вводиться багато інших специфічних для окремих галузей категорій, до яких можна легко перейти з поточного гасла. Це забезпечує зручну навігацію для тих користувачів е-ВУЕ, що досліджують певну проблему, а не шукають визначення конкретного терміну.

Третій тип пошуку – за типом інформаційного об'єкта – підтримується в е-ВУЕ за допомогою великої кількості шаблонів для типових ІО. Кожен шаблон пов'язаний із відповідною категорією е-ВУЕ, і користувач може застосовувати ці категорії і їх ієрархію для навігації у ресурсі.

У багатьох випадках для користувачів найзручнішим виявляється пошук за типом інформаційного об'єкта. *Інформаційний об'єкт* (ІО) – це інформаційна модель об'єкта певної предметної області, що визначає структуру, атрибути, обмеження цілісності і, можливо, поведінку цього об'єкта.

Четвертий тип пошуку – це *семантичний пошук*, результатом якого має стати інформаційний об'єкт зі складною структурою, знання про яку використовуються в пошукових процедурах. Його можна розглядати як окремий випадок проблеми розпізнавання об'єктів, якщо у випадку енциклопедій повна реалізація потребує встановлення семантичних плагінів, призначених для оперування із семантичними властивостями статей.

У різних Wiki-ресурсах реалізуються різні підмножини видів пошуку. Це залежить від обсягу Wiki-ресурсу, складності його структури та від того, на яких саме користувачів орієнтовано цей ресурс.

Для типових ІО доцільно виділити найбільш характерні параметри, які можна

відображати за допомогою семантичних властивостей Wiki. Для уніфікованого представлення таких властивостей доцільно розробити відповідні шаблони. Wiki-онтологія дозволяє формалізувати ці параметри та їх властивості, зв'язати їх з категоріями Wiki-ресурсу, що відповідають цим IO. Це дозволяє будувати семантичні запити для пошуку IO за їх властивостями та категоріями.

Саме наявність онтологічної моделі БЗ e-BUE – основа для виконання семантичного пошуку: Wiki-онтологія містить знання про семантичні властивості різних типових IO – їх назви, призначення, можливі значення тощо. У семантичному пошуку під IO будемо розуміти ту інформацію, яку користувач отримує в результаті виконання процесу пошуку. Стосовно Wiki-ресурсів, як окремі IO доцільно розглядати Wiki-сторінки як звичайні, так і спеціальні (приміром, сторінки категорій), які містять однаковий набір семантичних властивостей та відносяться до тих самих категорій.

Найчастіше використовуються запити, що сполучені з функцією *ask*. Ця функція використовується так само як і інші функції синтаксичного аналізатора MediaWiki: її виклик позначається подвійними фігурними дужками, перед іменем ставиться символ #, а після – двокрапка. Спочатку передається сам рядок запиту, що відбирає потрібну інформацію з Wiki, а потім усі параметри запиту, розділені символами вертикальної риси |. В умовах запиту можуть використовуватися назви категорій та семантичних властивостей, а також обмеження щодо значень цих властивостей. Крім того, можуть використовуватися деякі службові параметри, такі як поточний час або дата та назва поточної сторінки. Запити *ask* мають таку форму:

```
{{#ask:Критерії вибору сторінок
|?Запитувана властивість
|параметри
}}
```

Якщо користувачів цікавлять не стільки назви сторінок, що відповідають умовам запитів, але й інформація, яка

представлена на цих сторінках, тоді потрібно отримати значення семантичних властивостей цих сторінок. Наприклад:

```
{{#ask:
[[Категорія:Країна]]
[[Населення::>2000000]]
|?Столиця
|format=broadtable
}}
```

Слід звернути увагу, що у запиті необхідно використовувати точні назви категорій та семантичних властивостей. Саме для цього може бути використана Wiki-онтологія IP, що інтероперабельно формалізує ці відомості і дозволяє бачити зв'язки між властивостями та категоріями, їх характеристики тощо, які явно не подаються у середовищі Semantic Media-Wiki.

За допомогою семантичного пошуку користувач може отримувати доступ до інформації з різних Wiki сторінок e-BUE, а застосування значень семантичних властивостей як параметрів запиту дозволяє автоматизувати відбір саме тієї інформації, що потрібна йому. Це дозволяє економити час та зусилля користувачів, підвищує ефективність використання енциклопедії та забезпечує повніше задоволення персональних інформаційних потреб різних груп користувачів.

Семантичні запити можуть вбудовуватися до існуючих сторінок і розширювати можливості навігації (рис. 1) або викликатися за допомогою шаблонів. Це забезпечує можливість інтеграції відомостей з різних гасел, дозволяє запобігти повторному введенню інформації та гарантує автоматизоване внесення змін до усіх результатів запиту у випадку змін у тих сторінках, серед яких виконується пошук.

Наприклад, для сторінок категорій, що відповідають галузям знань e-BUE та їх підкатегоріям, за допомогою запитів, що вбудовані у шаблони, виконується пошук нових гасел відповідної категорії та гасел, для яких потрібні автори, виводиться інформація про модератора галузі тощо. У шаблоні “Нові_сторінки_категорії” міститься наступний запит:

Технічні науки

Це категорія Технічні науки.
Кількість статей у цій категорії: 300.
Технічні науки — сукупність наук, що досліджують техніку і вивчають явища, пов'язані з її створенням, розвитком і взаємодією з природою та людиною.

Нові статті

В цьому місяці опубліковано такі статті з цієї категорії:

А

- Алмазний інструмент (15 02 2019)
- Алідада (02 10 2019)

Автори категорії

До цієї категорії писали статті такі автори:

А	А (прод.)	А (прод.)
• АН (Снігир Г. В.)	• Автоколивальна система (Смирнов В. П.)	• Агломерат (металургія) (Вергельська Н. В.)
• Абдуллаєв, Гасан Мамедбагир огли (Храмов Ю. А.)	• Автомагістраль (Савенко В. Я.)	• Агломерація матеріалу (Нестеров О. С.)

Рис. 1. Сторінка категорії е-ВУЕ “Технічні науки”, контент якої будується в результаті виконання семантичних запитів

```

{{#ask:
[[Категорія:{{PAGENAME}}]]
[[Дата оприлюднення::>0]]
[[Місяць оприлюднення::{{CURRENTMONTH}}]]
[[Рік оприлюднення::{{CURRENTYEAR}}]]
|?Дата оприлюднення
|?Місяць оприлюднення
|?Рік оприлюднення
|format=category
|link=all
|order=descending
|sep=.
|intro=В цьому місяці опубліковано
|outro=
|default=В цьому місяці не оприлюднено нові
статті.
}}

```

Якщо з'являються нові гасла категорії або в таких гаслах інформація змінюється, контент сторінки категорії оновлюється автоматично.

Такі шаблони та вбудовані запити доцільно створювати лише для тих запитів, які мають виконуватися регулярно та можуть зацікавити велику кількість користувачів. Якщо інформаційні потреби користувача більш специфічні, йому потрібно побудувати семантичний запит самостійно.

Середовище Semantic MediaWiki підтримує просту, але досить потужну мову запитів *SMW-QL* для семантичного пошуку у Wiki-ресурсах, яка дозволяє, по-перше фільтрувати сторінки за заданими критеріями, і по-друге, виводити як результати запиту тільки цікавлячого користувача інформацію, а не весь текст Wiki-сторінки.

Результати таких запитів динамічно інтегрують актуальну інформацію з інших Wiki-сторінок і відображають її у зручному для користувача форматі – у вигляді діаграм, географічних карт, таблиць і схем

тощо. Якщо контент сторінок, з яких здобуваються потрібні дані, змінюється, то результати запитів також автоматично оновлюються, забезпечуючи цим несуперечність і погодженість даних. Це дуже зручно для IP зі складною структурою та розгалуженими зв'язками між сторінками.

На жаль, пошукова мова Semantic MediaWiki має певні обмеження:

- у семантичних запитах не обробляються ієрархічні відношення між категоріями, наприклад, немає засобів пошуку підкатегорій обраної категорії;

- категорії не можуть використовуватися як змінна у запиті (можна вводити тільки фіксоване значення), наприклад, неможливо знайти Wiki-сторінки, що відносяться до тих категорій, що й поточна сторінка;

- категорії Wiki-сторінок, що відповідають умовам запиту, не можуть виводитися в його результатах.

Це обмежує сферу застосування семантичних запитів та ускладнює виконання пошукових процедур, що пов'язані з поточною сторінкою, яку проглядає користувач. Тому виникає потреба у створенні додаткових засобів пошуку та навігації в IP, що базуються на Semantic MediaWiki, які дозволяють усунути ці обмеження. Але їх застосування потребує певного теоретичного обґрунтування.

Методи аналізу подібності понять

Для аналізу семантичної близькості між поняттями ПрО можуть використовуватися методи, що спрямовані на знаходження подібності між елементами різних онтологій. Оцінка подібності понять може базуватися на їх позиціях в ієрархії класів, для яких вже визначено подібність: якщо підкласи та надклас цих понять подібні, то самі такі поняття теж можуть бути подібними. Для кількісної оцінки подібності двох сутностей можуть враховуватися наступні параметри (ознаки):

- оцінка подібності їх прямих надкласів;
- оцінка подібності всіх їх надкласів;
- оцінка подібності підкласів понять;
- оцінка подібності екземплярів понять.

Аналогічно будуються оцінки подібності з урахуванням інших (не ієрархічних) зв'язків між поняттями. Наприклад, якщо поняття $p_{O_{12}}$ та $p_{O_{12}}$ пов'язані відношенням r_{O_1} в онтології O_1 ; поняття $p_{O_{21}}$ та $p_{O_{22}}$ пов'язані відношенням r_{O_2} в онтології O_2 ; поняття $p_{O_{12}}$ подібне до поняття $p_{O_{22}}$ (оцінка подібності вище за порогову); відношення r_{O_1} подібне до відношення r_{O_2} (оцінка подібності вище за порогову), тоді можна припустити, що поняття $p_{O_{11}}$ теж подібне до поняття $p_{O_{21}}$ (оцінка подібності вище за порогову).

Для оцінки близькості понять може застосовуватися також статистичний аналіз, який базується на порівнянні наборів екземплярів двох класів: подібність понять визначається шляхом порівняння екстенціоналів цих понять. Знаходження відповідності між поняттями базується на наступному емпіричному правилі:

- класи p_1 та p_2 онтології O_1 *еквівалентні*, якщо для всіх екземплярів онтології $P_i, \forall i = \overline{1, n}$ виконується $P_i \in p_1 \Rightarrow P_i \in p_2$ і, навпаки, $P_i \in p_2 \Rightarrow P_i \in p_1$;

- клас p_1 онтології O_1 є *підкласом* p_2 цієї онтології, якщо для всіх екземплярів онтології $P_i, \forall i = \overline{1, n}$ виконується $P_i \in p_1 \Rightarrow P_i \in p_2$.

Для Wiki-ресурсів, що розробляються без формалізації структури БЗ (прикладом такого IP є Вікіпедія), необхідно застосовувати саме такі підходи до знаходження СБП. Складність проблеми співставлення понять в онтології пов'язана з її поганим масштабуванням: збільшення кількості понять в онтології та ускладнення її структури значно збільшують простір пошуку. Тому в IP великого обсягу такі методи потребують значних обчислювальних ресурсів.

Якщо потрібно визначати подібність понять в одній онтології, то задача значно спрощується. Тому для тих IP, що мають заздалегідь розроблену систему класів та відношень (прикладом такого IP

є e-BUE), що може бути формалізована як Wiki-онтологія, доцільно застосовувати методи знаходження СБП, що базуються на семантичних мережах.

Використання онтологій у пошуку змістовно близьких понять дозволяє застосовувати знання Про для більш інтелектуальної навігації в IP, надаючи прямі переходи до змістовно пов'язаних сторінок.

Оцінювання семантичної спорідненості понять з використанням семантичних мереж для подання знань має довгу історію [9]. Семантична подібність є особливим випадком семантичної спорідненості [10]. Деякі дослідники припускають, що оцінку подібності в семантичних мережах потрібно розглядати із залученням лише таксономічних зв'язків [11], виключаючи інші типи зв'язків; але зв'язки між частинами також можна розглядати як атрибути, які впливають на визначення подібності [12]. В літературі визначено багато критеріїв подібності, але вони рідко супроводжуються незалежною характеристикою явища, яке вони вимірюють: їх цінність полягає в корисності для конкретного завдання.

Для окремого випадку онтології, де застосовується єдине відношення між поняттями – ієрархічне відношення типу IS-A, – таксономії – близькість двох термінів може бути оцінена за відстанню між поняттями в таксономії вершин, які відповідають цим термінам, в таксономії – окремому випадку онтології.

Семантична відстань між поняттями залежить від довжини найкоротшого шляху між вершинами та загальної специфічності двох вершин. Чим коротший шлях від одного вузла до іншого, тим більше вони подібні. Якщо між елементами існує кілька шляхів, використовують довжину найкоротшого з них [13, 14]. Довжина найкоротшого шляху в цій таксономії між відповідними поняттями, який визначається кількістю вершин (або ребер) в найкоротшому шляху між двома відповідними вершинами таксономії [11], з урахуванням глибини таксономічної ієрархії [15] (чим менше довжина шляху між вершинами, тим вони семантично ближчі):

$$S(a, b) = \log 2N/d(a, b),$$

де N – глибина таксономічного дерева, $d(a, b)$ – довжина найкоротшого шляху між вершинами a та b .

Однак з цим підходом пов'язана складна проблема, спричинена уявленням про те, що всі зв'язки в таксономії являють собою однорідні відстані. На жаль, рівномірну відстань у таксономії важко визначити і ще важче контролювати.

У реальних таксономіях існує велика мінливість «відстані», що охоплюється єдиним таксономічним зв'язком, особливо коли деякі підмножини таксономії (наприклад, біологічні категорії) є набагато більш щільними, ніж інші. Наприклад, у WordNet [16] – широко відомій семантичній мережі для англійської мови – можна знайти прямі зв'язки як між досить подібними поняттями, так і між відносно далекими. Тому доцільно враховувати семантику відношень між поняттями для різних таксономічних відношень та враховувати кількість екземплярів у підкласах.

Деякі міри близькості [17] враховують тільки глибину вершин термінів:

$$S(a, b) = 2 \times N(p(a, b)) / (N(a) + N(b)),$$

де $p(a, b)$ – найближчий спільний предок вершин a та b , $N(x)$ – глибина вершини x .

Подібність понять пов'язана також з їх інформаційним змістом (контентом). Нехай S – множина понять в таксономії $is-a$, що дозволяє множинне успадкування (multiple inheritance).

Одним із ключових факторів у подібності двох понять є ступінь, в якій вони поділяють інформацію, зазначену в таксономії IS-A високо специфічним поняттям, яке відноситься до обох цих понять. Метод підрахунку країв (edge-counting method) враховує це опосередковано, оскільки, якщо мінімальний шлях зв'язків IS-A між двома вузлами довгий, то це означає, що необхідно високо підійматися в таксономії до більш абстрактних понять, щоб знайти найменшу верхню межу – поняття, до якого відносяться обидва поняття, що аналізуються.

Інформаційний контент (*information content*) поняття c можна кількісно визначити як $-\log p(c)$: чим вище ймовірність використання поняття, тим нижче його інформативність. Таким чином, чим вище рівень абстракції поняття (тобто чим вище воно знаходиться в таксономії), тим менше його інформаційний контент. Якщо в таксономії існує унікальна верхня концепція, то її інформаційний контент дорівнює 0.

Така кількісна характеристика інформації забезпечує новий спосіб вимірювання семантичної подібності [18] на основі екстенсіоналу понять.

Чим більше інформації поділяють (сумісно використовують) два поняття, тим більше вони подібні, а інформація, що сумісно використовується двома поняттями, визначається інформаційним контентом понять, що входять до їх складу в таксономії. Формально така семантична подібність визначається наступним чином:

$$\text{sim}(c_1, c_2) = \max_{c \in S(c_1, c_2)} [-\log p(c)],$$

де $S(c_1, c_2)$ – множина понять, що входять до складу як c_1 , так і c_2 .

Хоча подібність обчислюється з урахуванням всіх верхніх меж для двох понять, інформаційна міра дозволяє ідентифікувати мінімальну верхню межу, оскільки жоден клас не є менш інформативним, ніж його надкласи.

На практиці часто виникає потреба у вимірюванні подібності слів, а не понять. Використовуючи для представлення слів з множини W через сукупності понять в таксономії, які є значеннями (змістами) слова w , функцію $s(w)$, таку, що $s: W \rightarrow C$, тобто

$$s(w \in W) = \{c_k \in C, k = \overline{1, m}\},$$

можна визначити

$$\text{sim}_w(w_1, w_2) = \max \text{sim}(c_i, c_j),$$

де $c_i \in s(w_1), c_j \in s(w_2)$.

Це узгоджується з визначенням «диз'юнктивних понять», що використовує підрахунок країв: вони визначають відс-

тань між двома диз'юнктивними наборами понять як мінімальну довжину шляху від будь-якого елемента з першого набору до будь-якого елемента з другого. Подібність слів оцінюється через знаходження максимального інформаційного контенту над усіма поняттями, для яких обидві слова можуть бути екземпляром.

Це дозволяє створювати множини *семантично близьких слів* (СБС), тобто слів, семантична відстань між якими менша за обрану порогову величину.

Семантично близькі поняття (СБП) – це нечітка множина, яка включає набір понять, для яких кількісне значення семантичної близькості з обраним поняттям вище заданого порогу [19]. Міри визначення семантичної близькості понять на основі онтологій використовують різноманітні семантичні характеристики цих понять – їх властивості (атрибути і відношення з іншими поняттями), взаємне положення в онтологічних ієрархіях.

В основу багатьох мір близькості на основі онтологій покладений теоретико-множинний підхід Тверської [20], який визначає міру близькості двох об'єктів шляхом співставлення властивостей (*feature matching*). Міра близькості $S(a, b)$ між об'єктами a і b – це функція трьох наборів властивостей цих об'єктів A і B , їх перетину $A \cap B$, доповнень $A - B$ та $B - A$.

Недоліком більшості мір, які основані на онтологічних структурах, є симетричність (експертні оцінки показують, що міра близькості не завжди симетрична), незалежність від контексту і чутливість до структури ієрархії. Цю проблему дозволяє вирішити застосування асиметричної міри семантичної близькості [21], яка залежить від напрямку руху між ребрами (наприклад, нащадок більш подібний до предка, ніж предок до нащадка).

В інших роботах пропонуються міри близькості, які базуються на неієрархічних («горизонтальних») відношеннях і атрибутах. Оцінювання близькості понять, яке використовує горизонтальні відношення, спирається на припущенні, що якщо два поняття мають одне й те саме відношення з третім поняттям, тоді вони ближче, ніж два поняття, які мають це відношення з різними поняттями, тобто

близькість двох понять залежить від близькості тих понять, з якими вони мають відношення. Така міра близькості обчислюється рекурсивно.

Атрибутивна міра близькості базується на близькості значень спільних атрибутів понять, діапазони яких є літералами, числами, рядками та іншими типами даних. Як міри близькості для рядкових даних можна використовувати нормовану редакторську відстань [22], для чисел – інверсію різниці, що нормована максимальним значенням атрибуту.

Використання цих мір близькості дозволяє оцінювати подібність значень параметрів понять (властивості даних екземплярів класів онтології), яким в семантизованих Wiki-ресурсах відповідають значення семантичних властивостей, що не є посиланнями на інші Wiki-сторінки.

Аналіз існуючих підходів до кількісного оцінювання семантичної подібності понять показує доцільність використання для цього таксономій та відстані в цих таксономіях між поняттями, близькість яких оцінюється, та їх спільним надкласом. Урахування інших типів онтологічних відношень між поняттями та порівняння їх семантичних властивостей дозволяють уточнювати ці оцінки відповідно до специфіки ПрО.

Розглянуті вище методи знаходження СБП можуть використовуватися в е-ВУЕ, де реалізується кілька незалежних таксономій для категоризації гасел, а інші зв'язки між гаслами встановлюються за допомогою семантичних відношень. Ця інформація щодо структури БЗ е-ВУЕ представлена у вигляді Wiki-онтології та може застосовуватися як користувачами енциклопедії, так і іншими інтелектуальними ІС.

Структура БЗ е-ВУЕ

Основний інструмент організації БЗ порталної версії е-ВУЕ – засоби, що надаються Semantic MediaWiki (категорії та семантичні властивості). Для структурованого подання інформації використовується набір незалежних таксономій, з якими пов'язані відповідні набори кате-

горій, між якими встановлені об'єктні відношення типу «клас-підклас» – кожне гасло може бути віднесено до довільного набору існуючих категорій, і наявність ієрархічних зв'язків між цими категоріями не викликає суперечностей в обробці інформації. Це дозволяє відображати різні аспекти, за якими можна класифікувати гасла енциклопедії [23]. Для інтеграції набору семантичних властивостей виділяються типові *інформаційні об'єкти* (ІО) [24] – гасла, що відносяться до визначеного набору категорій та мають фіксований набір характеристик. Більш детально типові ІО для е-ВУЕ проаналізовано в [25].

Онтологічна модель БЗ е-ВУЕ дозволяє автоматизовано знаходити змістовно близькі поняття серед гасел енциклопедії, аналізуючи їх структуру, семантику та зв'язки з іншими гаслами. Це дозволяє пропонувати користувачам нові шляхи навігації у Wiki-ресурсі та розширені функціональні можливості у пошуку.

Для визначення близькості між поняттями можуть використовуватися як знання з відповідних онтологій ПрО, наприклад, кількість посилань-зв'язків між поняттями та рівень значущості цих відношень, так і відомості з таксономій, що фіксують відношення між категоріями, та враховуються як кількість, наприклад, кількість рівнів між категоріями, до яких віднесені поняття, що порівнюються, та обсяг цих категорій.

БЗ е-ВУЕ підтримує довільні змістовні відношення між поняттями, водночас як у Вікіпедії можуть використовуватися лише два типи відношень – ієрархічне відношення «Клас-елемент класу» (категоризація) та відношення зв'язку (посилання між сторінками).

Наявність довільних, семантично визначених відношень значно розширює як виразність подання знань у ресурсі, так і можливості навігації у ньому.

В е-ВУЕ Wiki-онтологія дозволяє явно визначати наявність або відсутність ієрархічних зв'язків між категоріями е-ВУЕ, припустимість або неприпустимість їх перетину; відображати зв'язки

між категоріями та типовими Ю; вказувати, які саме семантичні властивості Wiki-сторінок, що відображають змістовні відношення між різними сторінками енциклопедії, і для яких Ю припустимі ті або інші відношення, притаманні кожному з цих Ю; описувати властивості категоризованих гасел e-ВУЕ та визначати характеристики цих властивостей.

Онтологічна модель структури БЗ e-ВУЕ використовується для підтримки семантичної навігації на порталі. Однією з істотних переваг e-ВУЕ як семантичного порталу є можливість знаходження СБП. Такий пошук базується на наступних припущеннях:

- поняття, що відповідають гаслам, які відносяться до однакового набору категорій, семантично ближчі одне до одного, ніж інші поняття, відображені на порталі;

- поняття, що відповідають гаслам, які мають однакові чи близькі значення семантичних властивостей, семантично ближчі одне до одного, ніж поняття, що відповідають гаслам, які відображені на порталі, в яких семантичні властивості мають інші значення або взагалі не визначені;

- поняття, які визначені як семантично близькі за обома попередніми критеріями, є більш семантично близькими, ніж ті, для яких виконується тільки одна умова близькості або не виконується жодної.

Для e-ВУЕ потребу у знаходженні СБП може викликати нездатність користувача коректно обрати галузь знань, до якої відноситься потрібне йому гасло, або без помилок ввести потрібний термін. В таких випадках користувач може отримати доступ до гасла, що він шукає, не безпосередньо, а через ті семантично близькі гасла, які він спроможний знайти.

Наприклад, користувач хоче знайти відомості про письменника або художника, прізвище якого він точно не пам'ятає, та не здатний точно визначити жанр його творів, але може вказати прізвище його більш відомого сучасника, який працював

у тому ж жанрі. В деяких випадках проблема знаходження СБП вирішується за допомогою знаходження СБС: це дозволяє оцінювати семантичну близькість природно-мовних текстів, що характеризують гасла e-ВУЕ.

Для розширення функціоналу e-ВУЕ, що стосується пошуку та навігації у ресурсі, виникає потреба у засобах знаходження семантично близьких Ю – як *глобально* близьких (за всією сукупністю ознак), так і *локально* близьких (тільки за певною підмножиною ознак).

Для визначення кількісних оцінок семантичної близькості довільних гасел та категорій e-ВУЕ, які при цьому розглядаються як екземпляри класів та класи відповідної БЗ, проведено аналіз наукових досліджень цієї проблематики, розглянуто переваги та недоліки існуючих підходів, враховано можливість їх масштабування для великих даних.

Використання онтологій як основи формалізації структури бази знань є ефективним засобом для встановлення семантичної близькості між елементами цієї БЗ. Особливу увагу викликає обробка окремих випадків онтологічних відношень – таких, як відношення «is-a», синонімії та ієрархічних відношень «Клас-підклас» та «Клас-екземпляр класу», що реалізуються базовими засобами технологічного середовища Semantic MediaWiki.

Для класифікації сторінок e-ВУЕ гасел застосовується набір незалежних таксономій, які дозволяють знаходити семантично подібні гасла енциклопедії та можуть використовуватися для класифікації гасел як окремо, так і інтегровано;

- поділ на три основні групи – Персоналії, Цивілізація та Природа;

- категоризація за тематичною спрямованістю: поділ на 27 базових напрямків знань відповідно з класифікацією ВАК із підкатегоріями різного рівня;

- таксономія типових Ю, для яких розроблено шаблони з основними семантичними властивостями та уніфікованою формою подання на порталі;

– поділ за формою публікації: сторінки, що представлено у паперовій версії, та сторінки, що подано виключно на порталі;

– поділ за ступенем готовності: завершені сторінки; сторінки, що шукають авторів; сторінки, над якими ведеться робота;

– поділ за часом публікації, що враховує такі семантичні властивості Wiki-сторінки, як Рік публікації, Місяць публікації та Дата публікації;

– поділ за територіальною спорідненістю, що враховує семантичні властивості Wiki-сторінки з областю значення типу Місто, Країна, Регіон тощо (наприклад, Місце народження, Місце розташування);

– поділ за часовою спорідненістю, що враховує семантичні властивості Wiki-сторінки з областю значення типу Дата (наприклад, Рік народження, Рік створення);

– поділ за параметрами, специфічними для певної наукової галузі або Про (наприклад, Кількість учасників, Площа).

В е-ВУЕ підтримуються всі основні типи пошуку – як за формальними ознаками (за ключовими словами) та за категоріями, так і на змістовному рівні за семантичними властивостями. Семантичний тип пошуку базується на оригінальній структурі БЗ е-ВУЕ, що розроблена з урахуванням досвіду експертів прикладних галузей та інженерів зі знань, і саме цей засіб доступу до інформації складає одну з основних переваг даного енциклопедійного ресурсу.

Використання СБП для навігації в е-ВУЕ

Для пошуку СБП в е-ВУЕ можуть використовуватися наступні джерела знань щодо понять:

– таксономії категорій, що використовуються в е-ВУЕ;

– природномовні описи понять, що містяться у відповідних Wiki-сторінках;

– гіперпосилання між сторінками;

– семантичні властивості, що пов'язують між собою Wiki-сторінки е-ВУЕ;

– семантичні властивості, що пов'язують сторінки понять із значеннями даних.

Відповідно до специфіки енциклопедичного IP, недоцільно шукати сторінки, що збігаються за усіма доступними параметрами – деякі групи параметрів є унікальними (наприклад, прізвище та ім'я разом з роком народження), а інші є функціонально залежними від інших параметрів, хоча мають і самостійну цінність (наприклад, прізвище особи мовою оригіналу). Тому пропонується знаходити *локальні* СБП, обираючи ті характеристики, які можуть бути корисними для досить широкого кола досліджень.

Для демонстрації можливостей описаного вище підходу пропонується наступні локальні СБП, пошук яких вбудованими засобами Semantic MediaWiki [26] не може бути реалізований:

1) СБП, визначені за *належністю до набору категорій*;

2) СБП, визначені за *значеннями обраних семантичних властивостей*;

3) СБП, визначені за *комбінацією категорій та значень семантичних властивостей*.

Семантична близькість шуканих понять визначається відносно поточної Wiki-сторінки, яку розглядає користувач, тобто категорії та властивості цієї сторінки аналізуються як параметри такого обчислення.

У першому випадку для поточної Wiki-сторінки потрібно знайти гасла, які віднесені одночасно до тих категорій, до яких віднесена поточна сторінка. Нині реалізовано пошук за категоріями та підкатегоріями галузей знань та за типовими Ю, а службові категорії, що пов'язані з формою публікації матеріалу (наприклад, "ВУЕ") не враховуються (рис. 2).

The image shows a screenshot of a web page titled "SemanticSimilarPages" with the main heading "Авіація" and the subtitle "Семантично близькі сторінки". A list of nine related pages is shown, with a callout box pointing to it that says "Семантично близькі сторінки". Below the list are three search method labels: "Semantic Similarity Search", "CountrymanPages Search", and "Persons By Category Search", with a callout box pointing to them that says "Пошук сторінок за набором категорій". To the right, a snippet of the article "Авіація" is visible, followed by a "Література" section with three references. Below that is the "Автор ВУЕ" section listing "В. М. Гребенніков". A callout box points to the category tags "Цивілізація", "Технічні науки", and "Авіаційна та ракетно-космічна техніка" with the text "Категорії гасла". At the bottom left, there are social media sharing buttons for Facebook, Twitter, and another platform.

Рис. 2. Пошук СБП для сторінки е-ВУЕ «Авіація»

Пошук може бути рекурсивно розширено для пошуку груп СБП, для яких спільною є тільки певна підмножина категорій поточної сторінки. Предметом дослідження залишається питання щодо ваги окремих категорій та врахування кількості підкатегорій та екземплярів у кожній з категорій, що досліджуються.

Другий варіант – пошук може здійснюватися не для всього контенту е-ВУЕ, а лише для сторінок, що відносяться до типових Ю одного типу – персоналій, міст, країн тощо. У такому пошуку враховуються значення деяких (обраних при створенні запиту) семантичних властивостей, що характерні для цього типу Ю, але значення цих властивостей залежатимуть від їх значень для поточної сторінки. Наприклад, для типового Ю “Персоналія” можна шукати осіб, що народи-

лися (рис. 3) в тому самому місці, сучасників тощо.

Третій варіант пошуку дозволяє шукати СБП обраної категорії з набором семантичних властивостей або категорій, що притаманні обраній сторінці. Наприклад, можна знайти видатних осіб (категорія “Персоналія”), що спеціалізувалися в тій сфері, до якої відноситься певне поняття або пристрій (рис. 4).

На сьогодні пошук СБП для е-ВУЕ знаходиться на стадії досліджень, і тому певні його елементи реалізовано лише на тестовій версії порталу.

Слід відмітити, що пошук СБП (локально та глобально) неможливо виконувати вбудованими засобами Semantic MediaWiki. Для кожного такого пошуку потрібно писати окремі запити, що аналізують код сторінок.

Семантичні властивості
Ю «Персоналія»
за якими
можна шукати СБП

Абрамович, Всеволод Михайлович (Abramowitsch, Wsewolod)	
Рік народження	1890
Місце народження	Одеса, Україна
Рік смерті	1913
Місце смерті	Берлін, Німеччина
Alma mater	Берлінський технічний університет, Берлін
Напрями діяльності	літакобудування, випробування літальних апаратів

Семантично близькі сторінки

CountrymanPages

Абрамович, Всеволод Михайлович

Земляки

Місце народження=Одеса, Україна

- 1 - Аксентьева, Зінаїда Миколаївна
- 2 - Авдєєнко, Геннадій Валентинович
- 3 - Авілов, Микола Вікторович
- 4 - Азрікан, Арнольд Григорович
- 5 - Алігер, Маргарита Йосипівна
- 6 - Абалакін, Віктор Кузьмич
- 7 - Аблов, Антон Васильович
- 8 - Аганін, Марк Абрамович

Пошук виконується за категорією Персоналія.

Рис. 3. Пошук земляків для сторінок е-ВУЕ категорії «Персоналія»


Search persons by page categories

Авіація


Семантичний пошук персоналій в множині категорій:

Технічні науки
Авіаційна та ракетно-космічна техніка
Проектування, виробництво та випробування літальних апаратів


Антонов, Олег Костянтинович




Сікорський, Ігор Іванович




Адер, Клеман Агнес




Абрамович, Всеволод Михайлович




Авдєєв, Сергій Васильович




Айвінс, Марша Сью




Айзлі, Донн Фултон




Адлер, Георгій Петрович (1886–1965)



Акіяма, Тойохіро



Анатра, Артур Антонович (1919–1943)



Категорії: Цивілізація Технічні науки Авіаційна та ракетно-космічна техніка
Проектування, виробництво та випробування літальних апаратів

f поділитися

tвітнути

поділитися

Semantic Similarity Search

CountrymanPages Search

Persons By Category Search

Пошук сторінок за набором категорій

Категорії гасла

Семантично близькі персоналії

Рис. 4. Пошук спеціалістів (за набором категорій сторінки) для сторінок е-ВУЕ

Висновки

В результаті проведених досліджень розроблено структурну схему БЗ е-ВУЕ та обумовлено наявність у цій структурі окремих елементів та зв'язків між ними. На основі аналізу специфіки знань, що представлені в е-ВУЕ, обґрунтовано потребу в використанні онтологічної моделі Вікі-ресурсу та визначено специфіку Вікі-онтології, що відповідає особливостям е-ВУЕ, визначено сферу застосування цієї моделі.

Використання цієї моделі забезпечує підвищення ефективності пошуку та навігації в інформаційному контенті портальної версії енциклопедії, підтримує семантичний пошук.

Проаналізовано існуючі підходи та методи знаходження СБП, розглянуто роль онтологій в цьому процесі, досліджено специфіку знаходження семантично близьких Вікі-сторінок та можливості застосування для цього онтологічної моделі БЗ е-ВУЕ, наведено практичні приклади застосування запропонованого підходу.

Література

1. Semantic MediaWiki. – https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki
2. Guarino N. Formal Ontology in Information Systems. Formal Ontology in Information Systems. Proc. of FOIS'98. 3–15. 1998.
3. Ushold M., Gruninger M. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*. 1996. Vol. 11, N 2.
4. Asuncion G. Ontological Engineering: with Examples from Areas of Knowledge Management, e-Commerce and the Semantic Web (Advanced Information and Knowledge Processing). – http://www.amazon.com/gp/reader/1852335513/ref=sib_rdr_toc/.
5. Rogushina J. Semantic Wiki resources and their use for the construction of personalized ontologies. CEUR Workshop Proceedings 1631. 2016. P. 188–195.
6. Рогушина Ю.В. Теоретичні засади застосування онтологій для семантизації ресурсів Web. *Проблеми програмування*. 2018. № 2-3. С. 197–203.
7. W3C Semantic Web Activity. <http://www.w3.org/2001/sw/Activity/>.
8. Рогушина Ю.В., Прийма С.М., Строкань О.В. Створення та використання семантичних Вікі-ресурсів: навчальний довідник. Мелітополь, ФОП Однорог Т.В. 2017. 169 с.
9. Quillian M.R. Semantic memory. In Minsky, M. (Ed.), *Semantic Information Processing*. MIT Press, Cambridge, MA, 1968.
10. Collins, A., Loftus, E. A spreading activation theory of semantic processing. *Psychological Review*. 1975. 82. P. 407–428.
11. Rada R., Mili H., Bicknell E., Blettner M. Development and application of a metric on semantic nets. *IEEE Transaction on Systems, Man, and Cybernetics*. 1989. 19(1). P. 17–30.
12. Richardson R., Smeaton A.F., Murphy J. Using WordNet as a knowledge base for measuring semantic similarity between words. Working paper CA-1294, Dublin City University, School of Computer Applications, Dublin, 1994. <ftp://ftp.compapp.dcu.ie/pub/w-papers/1994/CA1294.ps.Z>
13. Lee J.H., Kim M.H., Lee Y.J. Information retrieval based on conceptual distance in IS-A hierarchies. *Journal of Documentation*. 1993. 49(2). P. 188–207.
14. Rada R., Bicknell E. Ranking documents with a thesaurus. *JASIS*. 1989. 10(5). P. 304–310.
15. Leacock C., Chodorow M. WordNet: An electronic lexical database. Cambridge. 1998. P. 265.
16. Miller G. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4). (Special Issue). 1990.
17. Wu Z., Palmer M. Verb semantics and lexical selection. Proc. 32nd Annual Meeting of the Association for Comput. Linguistics. Las Cruces. 1994. P. 133–138.
18. Resnik P. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*. 1999. 11. P. 95–130.
19. Крюков К.В., Панкова Л.А., Пронина В.А., Шипилина Л.Б. Меры семантической близости в онтологиях. Труды научных сессий МИФИ, Научная сессия МИФИ-2010. Т. 5. *Информационно-телекоммуникационные системы. Проблемы информационной безопасности*. С. 75–78.
20. Tversky A. Features of Similarity. *Psychological Rev*. 1977. Vol. 84. P. 327.
21. Bulskov H., Knappe R., Andreasen T. On Measuring Similarity for Conceptual

- Querying. Proc. 5th Int. FQAS Conf. LNCS. V. 2522. Berlin: Springer. 2002. P. 100.
22. Levenshtein I.V. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*. 1966. Vol. 10. P. 707.
 23. Методичні рекомендації з підготовки, редагування та оформлення статей до Великої української енциклопедії / За ред. Киридон А.М. К.: ДНУ «Енциклопедичне видавництво». 2015. 120 с.
 24. Rogushina J.V. The Use of Ontological Knowledge for Semantic Search of Complex Information Objects. Proc. of OSTIS-2017. 2017. P. 127–132.
 25. Рогушина Ю.В. Використання семантичних властивостей вікі-ресурсів для розширення функціональних можливостей «Великої української енциклопедії». Енциклопедичні видання в сучасному інформаційному просторі: колективна монографія / За ред. Киридон А.М. – К.: Державна наукова установа «Енциклопедичне видавництво». 2017. С. 104–115.
 26. Гришанова І.Ю., Рогушина Ю.В. Адаптація технологічних засад semantic mediawiki до потреб онлайн-версії великої української енциклопедії ВУЕ. Енциклопедистика в Україні: люди, ідеї, поступ: колективна монографія / За ред. Киридон А.М. К.: Державна наукова установа «Енциклопедичне видавництво». 2018. С. 240–253.
 - ontologies. CEUR Workshop Proceedings 1631. 2016. P. 188–195.
 6. Rogushina J. (2018) Theoretical means of ontology use for the Web resources semantization. Problems in Programming. N 2-3. P. 197–203. [in Ukrainian]
 7. W3C Semantic Web Activity. – <http://www.w3.org/2001/sw/Activity/>.
 8. Rogushina Y.V., Priyma S.M., Stokan O.V. (2017) Creating and use of the Semantic Wiki resources: tutorial. Melitopol, FOP Odiorog T.V. 169 p. [in Ukrainian]
 9. Quillian M.R. Semantic memory. In Minsky, M. (Ed.), *Semantic Information Processing*. MIT Press, Cambridge, MA, 1968.
 10. Collins, A., Loftus, E. A spreading activation theory of semantic processing. *Psychological Review*. 1975. 82. P. 407–428.
 11. Rada R., Mili H., Bicknell E., Blettner M. Development and application of a metric on semantic nets. *IEEE Transaction on Systems, Man, and Cybernetics*. 1989. 19(1). P. 17–30.
 12. Richardson R., Smeaton A.F., Murphy J. Using WordNet as a knowledge base for measuring semantic similarity between words. Working paper CA-1294, Dublin City University, School of Computer Applications, Dublin, 1994. <ftp://ftp.compapp.dcu.ie/pub/w-papers/1994/CA1294.ps.Z>
 13. Lee J.H., Kim M.H., Lee Y.J. Information retrieval based on conceptual distance in IS-A hierarchies. *Journal of Documentation*. 1993. 49(2). P. 188–207.
 14. Rada R., Bicknell E. Ranking documents with a thesaurus. *JASIS*. 1989. 10(5). P. 304–310.
 15. Leacock C., Chodorow M. WordNet: An electronic lexical database. Cambridge. 1998. P. 265.
 16. Miller G. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4). (Special Issue). 1990.
 17. Wu Z., Palmer M. Verb semantics and lexical selection. Proc. 32nd Annual Meeting of the Association for Comput. Linguistics. Las Cruces. 1994. P. 133–138.
 18. Resnik P. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*. 1999. 11. P. 95–130.
 19. Kriukov K.V., Pankova K.V., Pronina V.A., Shipilina L.B. (2010) Measures of semantic proximity in ontologies. Proc. of MIFI-2010, Vol. 5. Information- telecommunication systems. Problems of information security. P. 75–78. [in Russian]

References

1. Semantic MediaWiki. – https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki
2. Guarino N. Formal Ontology in Information Systems. *Formal Ontology in Information Systems*. Proc. of FOIS'98. 3–15. 1998.
3. Ushold M., Gruninger M. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*. 1996. Vol. 11, N 2.
4. Asuncion G. Ontological Engineering: with Examples from Areas of Knowledge Management, e-Commerce and the Semantic Web (Advanced Information and Knowledge Processing). – http://www.amazon.com/gp/reader/1852335513/ref=sib_rdr_toc/.
5. Rogushina J. Semantic Wiki resources and their use for the construction of personalized

20. Tversky A. Features of Similarity. *Psychological Rev.* 1977. Vol. 84. P. 327.
21. Bulskov H., Knappe R., Andreasen T. On Measuring Similarity for Conceptual Querying. *Proc. 5th Int. FQAS Conf. LNCS*. V. 2522. Berlin: Springer. 2002. P. 100.
22. Levenshtein I.V. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*. 1966. Vol. 10. P. 707.
23. Methodical recommendations for preparing, editing and design of articles for Great Ukrainian Encyclopedia (2015) / Ed. Kyrydon A.M., Kyiv, Kyiv, 120 p. [in Ukrainian]
24. Rogushina J.V. (2017) The Use of Ontological Knowledge for Semantic Search of Complex Information Objects // *Proc. of OSTIS-2017*. P. 127–132.
25. Rogushina J.V. (2017) Use of semantic properties of the Wiki resources for expansion of functional possibilities of “Great Ukrainian Encyclopedia”. *Encyclopaedias in the modern information space: collective monograph* / Ed. Kyrydon A.M., Kyiv. P. 104–115. [in Ukrainian]
26. Grishanova I.Y., Rogushina J.V. (2018) Adaptation of technological means of Semantic Mediawiki for needs of online version of Great Ukrainian Encyclopedia // *Encyclopaedias in Ukraine: people, ideas, steps: collective monograph* / Ed. Kyrydon A.M., Kyiv. P. 240–253. [in Ukrainian]

Про авторів:

Рогушина Юлія Віталіївна,
кандидат фізико-математичних наук,
старший науковий співробітник.
Кількість наукових публікацій в
українських виданнях – 150.
Кількість наукових публікацій в
зарубіжних виданнях – 31.
<http://orcid.org/0000-0001-7958-2557>,

Гришанова Ірина Юріївна,
науковий співробітник.
Кількість наукових публікацій в
українських виданнях – 18.
Кількість наукових публікацій в
зарубіжних виданнях – 3.
<http://orcid.org/0000-0003-4999-6294>.

Місце роботи авторів:

Інститут програмних систем
НАН України,
03181, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: 066 550 1999.
E-mail: ladamandraka2010@gmail.com,
i26031966@gmail.com

Одержано 16.10.2019

Ю.В. Рогушина

ВИКОРИСТАННЯ ТЕЗАУРУСІВ ДЛЯ ПОШУКУ СКЛАДНИХ ІНФОРМАЦІЙНИХ ОБ'ЄКТІВ У WEB НА ОСНОВІ ОНТОЛОГІЙ

Запропоновано онтологічну модель взаємодії між об'єктами та суб'єктами семантичного пошуку у Web, охарактеризовано її основні елементи, розглянуто методи її поповнення та застосування для фільтрації інформації, що відповідає персоніфікованим потребам користувачів. Проаналізовано типи відношень між екземплярами та класами цієї моделі та їх характеристики, що можуть впливати на часову складність обробки знань, що подані на основі цієї моделі. Одним з важливих елементів запропонованої моделі є тезауруси, які відображають знання щодо задач, для розв'язання яких користувачі шукають інформацію, та щодо інформаційних ресурсів, в яких такі відомості можуть міститися. Обґрунтовується доцільність застосування окремих випадків онтології – тезаурусів – для знаходження семантично подібних інформаційних об'єктів. Розглянуто види тезаурусів, які застосовуються для семантичного пошуку, наведено джерела їх поповнення та проаналізовано їх характеристики. В роботі запропоновано алгоритм автоматизованої побудови простого тезаурусу, що утворюється на основі онтології предметної області та природномовного опису задачі користувача, та методи генерації складених тезаурусів, що пертинентні новим задачам користувача, за множиною простих тезаурусів, що побудовані користувачем раніше. Оцінюються виразність та обчислювальна складність запропонованих методів, яка залежить від властивостей онтології предметної області та від обсягу опису задачі. Розглянуто методи використання семантично розмічених Wiki-ресурсів як джерела знань для побудови онтологій предметних областей та пов'язаних з ними типових інформаційних об'єктів.

Ключові слова: семантичний пошук, інформаційний об'єкт, онтологія, тезаурус задачі, семантична розмітка.

Вступ

Сучасні інтелектуальні інформаційні системи (ІС) орієнтовані на роботу в розподіленому середовищі Web, що потребує динамічного отримання актуальних та пертинентних знань з його ресурсів. Великий обсяг та складна структура інформаційних ресурсів (ІР), тенденція поширення великих даних (Big Data) викликають потребу у створенні засобів автоматизованої обробки інформації, які дозволили б аналізувати зміст цих ресурсів та здобувати з них саме ті відомості, що потрібні користувачу для вирішення його поточної задачі.

Один з найбільш перспективних напрямків розв'язання цієї задачі базується на використанні для цього зовнішніх баз знань, тобто із семантизацією як самих ресурсів Web, так і процесу їх пошуку. У використанні ресурсів Web на найбільш високому рівні можна виділити дві основні задачі:

– відбір ІР – пертинентні поточні задачі користувача, що містять інформацію для її розв'язку;

– здобуття з ІР тієї інформації, яка потрібна користувачеві.

Рішення першої задачі є об'єктом для інформаційно-пошукових та рекомендаційних систем і може бути вдосконалено за допомогою застосування фонових знань та інтелектуальних методів їх обробки. Для розв'язання другої задачі використовують Data Mining, Text Mining, методи машинного навчання тощо, що дозволяють здобути з даних неявно представлені в них відомості. Незалежно від того, наскільки досконалим буде розв'язок другої проблеми, вона не буде ефективно вирішена, якщо методи аналізу будуть обробляти не пертинентні дані.

У найбільш широкому розумінні семантизація полягає у встановленні зв'язку між певним інформаційним об'єктом (ІО) та його змістом. Під *семантизацією* ІР будемо надалі розуміти встановлення формалізованих відношень між цими ІР (або їх елементами та мета-описами) та формалізованим поданням знань (наприклад, з онтологією, семанти-

© Ю.В. Рогушина, 2019

чною мережею, фреймом), тобто їх семантичну розмітку на основі обраного подання знань. Така розмітка – це основа для більш ефективної навігації та пошуку в Web.

Під семантичним пошуком [1] зазвичай розуміють такий пошук інформації, коли для задоволення інформаційних потреб користувача, що виникають у процесі розв'язання певної задачі, використовуються зовнішні знання щодо суб'єктів і об'єктів пошукової процедури й методів аналізу цих знань. Це викликає потребу в застосуванні формально представлених зовнішніх відносно пошукової процедури знань щодо основних елементів цієї процедури. Такі фонові знання можуть стосуватися користувача та специфіки його інформаційних потреб (персоніфікація пошуку), структури IP, серед яких здійснюється пошук, тієї предметної області (ПрО), до якої відносяться ці IP. В процесі семантичного пошуку співставлення запиту користувача з контентом IP здійснюється не безпосередньо, а з урахуванням фонових знань співставляються їх формалізовані інформаційні моделі.

Використання типових IO дозволяє чіткіше визначити інформаційну потребу користувача на змістовному рівні. Це дозволяє категоризувати вміст IP та пов'язувати елементи контенту з певними поняттями ПрО, які є типовими – мають однакові властивості, відносяться до однакової групи класів, містять подібні за структурою та вмістом елементи. Наприклад, за тими самими умовами користувач може шукати людину, організацію або документ. Якщо пошук здійснюється серед структурованих та класифікованих IP, визначення типу IO дозволяє обрати категорію або набір категорій, до якої має відноситися шуканий IO. Для пошуку серед неструктурованої інформації виникає потреба спочатку отримати з фонових знань ПрО інформацію щодо структури шуканого IP (його властивості, їх типи та можливі значення, надкласи та підкласи тощо), а потім застосовувати цю інформацію для фільтрації результатів пошуку, отриманих за запитом користувача.

Джерелом таких знань можуть бути як онтології ПрО, до яких відносяться шукані IP, так і довільні семантично структуровані IP (наприклад, семантичні Wiki-ресурси). Важливо, що інформацію щодо структури та властивостей таких типових IO користувач може отримувати із зовнішніх джерел знань, а не формулювати самостійно. Це значно спрощує пошук IO зі складною структурою та дозволяє відфільтровувати необхідну інформацію серед великої кількості IP, але користувачеві потрібно самостійно обирати з таких наборів знань ту підмножину, яка пертинентна його задачі.

Наприклад, якщо таким IO є людина, то різним користувачам можуть бути необхідні різні аспекти відомостей – щодо освіти, кваліфікації, здоров'я, сімейного стану тощо.

Щоб використовувати онтологічні знання в процесі семантичного пошуку, потрібно забезпечити: 1) механізми створення онтологічних моделей інформаційних потреб користувачів та IP, серед яких здійснюється пошук; 2) методи зіставлення таких моделей. Перша проблема пов'язана з формалізацією властивостей основних елементів пошукової процедури, яка виконується із застосуванням фонових знань, а друга може розглядатися як окремих випадок співставлення незалежно створених онтологій, на які накладено деякі специфічні обмеження.

Онтологічна модель взаємодії користувачів та IP у Web

Щоб проаналізувати методи знаходження в Web IO зі складною структурою, що відповідають персональним інформаційним потребам користувачів, необхідно побудувати модель пошуку, яка дозволяє чітко та однозначно відобразити властивості основних компонентів пошукової процедури та зв'язки між ними. Такий опис має визначити всі базові терміни, що використовуються для опису задачі семантичного пошуку та характеризують його учасників, вхідні та вихідні дані, а також критерії, за якими оцінюються результати пошукового процесу.

Сьогодні для моделювання різноманітних ПрО широко застосовуються онтологічні моделі. Онтологічна модель семантичного пошуку (ОМСП) – це онтологічна модель, яка формалізує відношення між основними суб'єктами пошуку, до яких можна віднести користувачів, експертів, авторів IP тощо, і його об'єктами (такими, як IP, Ю, запити та результати їх виконання, описи ПрО тощо). Така модель дозволяє однозначно описати ті взаємини між користувачами та IP, які виникають в процесі використання знань для задоволення інформаційних потреб користувачів. Для подання моделі може бути використана мова OWL, що дозволяє застосовувати її в різних ПС, які функціонують в Web і використовують його IP [2].

Використання ОМСП у задачі семантичного пошуку є основою для інтелектуальної обробки ресурсів Web з використанням онтологічного аналізу. Основна ідея запропонованого підходу полягає у тому, що застосовуються два типи онтологій – зовнішні та внутрішні, відмінність між якими полягає у наступному:

- внутрішня онтологія створюється самими розробниками ПС відповідно до специфіки тих задач, що вирішуються системою, та формалізує структуру та відношення між основними суб'єктами та об'єктами цієї ПС, і тому всі характеристики цієї онтології відомі ще до початку роботи з ПС і дозволяють чітко та однозначно визначити її виразні можливості, обсяг та методи обробки;

- внутрішні онтології здобуваються з ресурсів Web у процесі функціонування ПС (їх знаходять у зовнішніх репозиторіях, будують відповідно до потреб користувачів, експортують із різноманітних семантичних представлень даних тощо), і тому неможливо оцінити до початку роботи їх властивості та виразну здатність, що безпосередньо визначають складність обробки.

Для задачі семантичного пошуку ОМСП є внутрішньою онтологією, тоді як отримані з різних джерел онтології ПрО, IP та Ю є зовнішніми.

Цей підхід може застосовуватися для розв'язку інших інтелектуальних задач, пов'язаних з аналізом інформаційних ресурсів Web. Прикладами таких задач є проактивне надання рекомендацій, машинне навчання, створення семантичних порталів. В таких випадках потрібно побудувати відмінну від ОМСП модель взаємодії елементів такої системи (слід зазначити, що багато класів ОМСП – такі, як користувач та ПрО – є досить універсальними, і їх можна переносити до нової моделі тільки з певними доповненнями), доповнену специфічними для задачі класами.

Крім того, ОМСП може бути використана для окремих випадків семантичного пошуку, приклади яких будуть розглянуті далі, – для пошуку фіксованих підмножин Ю (пошук вакансій та навчальних закладів, Web-сервісів) та для пошуку в інформаційному середовищі, що є підмножиною Web (пошук у репозиторіях RDF та OWL, у Wiki-ресурсах).

Основні *суб'єкти* інформаційного пошуку – сутності, які своїми діями можуть ініціювати пошуковий процес або впливати на його результати:

- *користувачі* – ті особи (люди або програмні сутності), які прагнуть за допомогою пошуку (наприклад, за допомогою певної ПС) отримати доступ до певної інформації;

- *експерти* – ті особи, які здатні певним чином оцінювати об'єкти і суб'єкти пошуку (приміром, надавати кількісну оцінку якості IP, його відповідності певному запиту, визначати зв'язок між онтологією ПрО та задачею користувача тощо);

- *власники IP* – особи, що створюють або публікують певну інформацію в Web та можуть визначати її тематику, якість, умови доступу тощо.

У семантичному пошуку додатково можуть використовуватися такі суб'єкти, як *група користувачів* – скінчена неперевірена множина користувачів, що поєднана за певними спільними властивостями. Приміром, у деяких рекомендуючих системах кожен користувач може визначити склад співтовариства, думки якого в по-

точній ситуації для нього мають певну цінність.

Основні *об'єкти* інформаційного пошуку – сутності, що використовуються в процесі виконання пошукових процедур: IP; IO; інформаційне середовище; інформаційно-пошукові системи (ІПС); інформаційні потреби користувачів (ІП); запити, що формалізують ІП користувачів; результати виконання запитів; зовнішні бази знань (БЗ).

IP – це сукупність даних (документів, файлів тощо), засобів доступу та користування цими даними (бібліотека, архів, база даних тощо). В даній роботі основна увага приділяється IP, що представлені в електронній формі та доступні за допомогою Web, тобто мають унікальні ідентифікатори (адреси) та характеризуються як за допомогою формальних властивостей (розмір, час створення модифікації, мова подання тощо), так і через їх контент. Також для опису IP можуть використовуватися метадані, що описують ці властивості певною формальною мовою (приміром, RDF).

IO – модель певного об'єкту Про в інформаційному просторі, що визначає структуру, атрибути, обмеження цілісності і, можливо, поведінку цього об'єкта через контент інформаційних ресурсів. До складу одного IP може входити кілька IO. З іншого боку, один IO може бути описаний за допомогою кількох IP. Приклади IO – Web-сервіс, організація, особа, документ. Приміром, сайт організації може складатися з набору окремих Web-сторінок, але на одній з цих сторінок можуть описуватися кілька осіб.

Інформаційне середовище – сукупність усіх доступних IP, їх властивостей (включаючи їх оцінки користувачами) і зв'язків між ними. У даній роботі під інформаційним середовищем будемо розуміти Web, якому характерні гетерогенність, динамічність та великий обсяг інформації, що визначають вимоги та обмеження до методів пошуку інформації, що розробляються. Інші приклади інформаційного середовища, що задають інші специфічні вимоги до пошуку, – корпоративні мережі, сховища даних різних типів,

інформаційний вміст локального обчислювального пристрою.

ІПС – засіб, що встановлює за певними критеріями кількісну міру відповідності між запитом користувача та інформацією щодо певної множини IP або IO та знаходить серед них підмножину найбільш відповідних.

ІП – усвідомлена необхідність в інформації для розв'язання поставленого завдання за розробленим планом. ІП, для задоволення якої і виконується пошук інформації, може бути формалізована за допомогою запиту (та його контексту), який характеризує поточні інтереси користувача, його задачу та здатність до сприйняття інформації тощо. У більшості випадків інформаційна потреба користувача є надто складною, щоб її формалізація відображала її повністю.

Запит – представлена за допомогою якоїсь мови формалізація інформаційної потреби користувача. Це може бути набір ключових слів – можливо, пов'язаних логічними операторами (такі запити застосовуються найчастіше), природномовне речення або перелік значень властивостей того IO, який має задовольнити інформаційну потребу (приміром, вхідні та вихідні дані Web-сервісу або адреса організації, назва якої потрібна користувачу).

Результат запиту – це скінчена впорядкована множина IP або IO, які ІПС відібрала серед усіх доступних джерел інформації шляхом співставлення інформаційної потреби користувача з інформацією щодо цих IP або IO. Результати виконання того самого запиту у різний час можуть різнитися як через зміни в оточуючому середовищі, так і через зміни у профілі користувача.

Зовнішня БЗ – сукупність формалізовано поданих знань, що створена та функціонує незалежно від дій користувачів та розробників пошукової системи, але може бути використана в процесі пошуку.

Крім основних об'єктів процесу інформаційного пошуку, ОСМП описує також додаткові об'єкти, що пов'язані із семантизацією та персоніфікацією пошукових процедур та з підтримкою колабора-

тивного пошуку. Додаткові об'єкти дозволяють більш точно охарактеризувати основні об'єкти цього процесу. До таких об'єктів належать:

- предметна область (ПрО);
- онтології ПрО та ІО;
- тезауруси;
- лексичні онтології;
- теми запитів.

ПрО – деяка підмножина реального світу, що відповідно до якогось набору ознак цікавить користувача у певний час. Це може бути галузь знань, сукупність територіально поєднаних сутностей тощо. ПрО може бути формально представлена через множину понять, їх властивостей, відношень між ними та різноманітних обмежень. Нині у Web-орієнтованих інтелектуальних системах для формалізації опису ПрО часто використовуються її онтології.

Онтологія ПрО – це довільна онтологія [3], представлена на одному з діалектів OWL [4] та придатна для комп'ютерної обробки. Класи цієї онтології відповідають поняттям обраної ПрО, її екземпляри пов'язані з окремими випадками цих понять, а властивості дозволяють визначити зв'язки між поняттями та їх екземплярами. Онтології дозволяють формально описувати як семантику ПрО, що цікавить користувача, і задачі, яку він прагне вирішити, так і семантику тих ІР та ІО, які містять потрібні користувачеві відомості. Слід зазначити, що ці онтології, на відміну від ОМСП, є зовнішніми для задачі пошуку: на відміну від ОМСП, що може в процесі функціонування системи семантичного пошуку тільки поповнюватися новими екземплярами класів та значеннями їх властивостей, ці онтології можуть змінюватися довільним чином – як внаслідок змін у тих ресурсах, за якими вони будуються, так і внаслідок безпосередніх вказівок користувача.

Онтологія ІО – онтологія (часто – таксономія), що формалізує структуру групи ІО, що є суб'єктами пошуку, та їх відношення як одного з одним, так і з іншими об'єктами ПрО, що впливають на обмеження та умови у пошуковому запиті щодо того, які саме типи та екземпляри ІО задовольняють потребам користувача.

Задача користувача – поточна задача, для розв'язку якої користувач потребує отримати певну інформацію з зовнішніх ІР. Може бути описана через природномовне (неструктуроване) або структуроване визначення, приклади, елементи метаданих.

Тезаурус задачі – це окремий випадок онтології ПрО, який містить тільки онтологічні терміни (класи та екземпляри), але не описує (або обмежено описує) семантику відношень між ними з метою аналізу природномовних текстів. Може автоматизовано генеруватися за онтологією ПрО та природномовним описом задачі. Це окремий випадок онтології. *Простий* тезаурус задачі – тезаурус, який базується на термінах однієї онтології ПрО. *Складений* тезаурус задачі – тезаурус, який базується на термінах двох або більш онтологій ПрО.

Тезаурус ІР – це підмножина тезаурусу задачі, який містить тільки ті його терміни, для яких знайдено відповідні фрагменти у контенті цього ІР. Таким чином, склад тезаурусу ІР залежить як від тезаурусу задачі, для якої він будується, так і від методу співставлення контенту ІР із термінами цього тезаурусу.

Лексична онтологія ПрО – онтологія, яка містить формалізовані знання щодо зв'язків між поняттями певної онтології ПрО та пертинентними їм елементами природномовних текстів.

Тема запитів – це скінчена невпорядкована множина запитів одного або кількох різних користувачів, які дозволяють згрупувати їх за певними спільними властивостями або шляхом перерахування для того, щоб спільно обробляти їх параметри або отримані за цими запитами результати. Теми запитів дозволяють структурувати колаборативний пошук та організувати обмін інформацією за визначеними напрямками.

На основі ОМСП створюється інтероперабельний *профіль користувача*, який базується на класі ОМСП “Користувач” та як об'єктні властивості використовує екземпляри інших класів цієї онтологічної моделі. Відомості в цьому профілі можна поділити на кілька груп:

- Реєстраційна інформація:
 - ідентифікатор користувача;
 - пароль для доступу до ІПС.
- Досвід взаємодії ІПС з користувачем:
 - список онтологій, які користувач застосовував для опису своїх інформаційних інтересів;
 - список тезаурусів, що користувач застосовував у пошукових запитах;
 - список раніше виконаних запитів;
 - список результатів виконаних запитів з оцінками користувача для знайдених результатів.
- Відомості, імпортовані з зовнішніх джерел (необов'язкові відомості, їх може й не бути):
 - ідентифікатори користувача в соціальних мережах, що дають змогу динамічно оновлювати відомості про нього;
 - рейтинги користувача в соціальних мережах;
 - адреса користувача у Вікіпедії та інших Wiki-ресурсах;
 - адреса сайту користувача;
 - сфера компетенцій користувача (ключові слова, імпортовані з соціальних мереж);
 - посилання на публікації користувача.

Власні характеристики користувача: сфера компетенцій користувача (список ключових слів, що вводяться користувачем безпосередньо).

– Формальні дані про користувача (необов'язкові відомості, що дають змогу ІПС формувати групи користувачів зі схожими інформаційними потребами): місце проживання; вік; професія, освіта тощо.

Для опису ОМСП пропонується використовувати наступну формальну модель онтології:

$$O = \langle X, R, F, T \rangle,$$

яка більш детально описана в [5]. Ця модель дозволяє формалізувати відношення між елементами процесу пошуку інформації в Web, вона досить добре співставля-

ється з технологічними елементами редактора онтологій Protégé та засобами семантичної розмітки Semantic MediaWiki, використання яких для поповнення онтологій розглядатиметься далі.

ОМСП містить такі основні класи, що пов'язані із типами об'єктів та суб'єктів семантичного пошуку:

– *користувач* – клас, екземпляри якого відповідають описам окремих користувачів, а властивості відповідають параметрам профілю користувача, який описано вище, та зв'язують екземпляри цього класу із екземплярами інших класів ОМСП та константами, що визначають значення певних параметрів із цього профілю;

– *онтологія ПрО*, що містить опис області, до якої належать інформаційні потреби користувача

$$O_{PrO_i} = \langle X_{PrO_i}, R_{PrO_i}, F_{PrO_i} \rangle, i = \overline{1, n};$$

– *лексична онтологія ПрО* – база знань щодо лексики ПрО, що містить відомості про лексеми природних мов, які відповідають термінам онтології ПрО

$$L_{PrO_i} = \langle X_{lex_i}, R_{lex_i} = \{r_{lex}\}, \emptyset \rangle, i = \overline{1, n},$$

де

$$X_{lex_i} = X_{PrO_i} \cup T_{PrO_i}$$

тобто

$$\forall x_{ij} \in X_{PrO_i}, j = \overline{1, m_i}$$

існує набір фрагментів ПМ

$$\{s_{ij_p} \in T_{PrO_i}\}, p = \overline{1, q_{ij}}, r_{lex}(s_{ij_p}) = x_{ij} -$$

така онтологія використовується для встановлення зв'язків між елементами природномовних документів і термінами онтології ПрО;

– *тезаурус* – множина термінів Th, що разом із своїми властивостями характеризують певний суб'єкт пошуку, дозволяючи співставляти його з іншими суб'єктами; цей клас у рамках ОМСП має наступні підкласи, екземпляри яких мають додаткові властивості:

– *тезаурус онтології* – множина термінів онтології

$$\text{Th}_O = \{x_k \in X\}, k = \overline{1, n};$$

– *тезаурус множини онтологій* – об'єднання тезаурусів множини онтологій

$$O^* = \{X_m\}, m = \overline{1, p},$$

такого, що містить p онтологій, $p \geq 1$,

$$\text{Th}_{O^*} = \{x_{k_m} \in X_m\}, k_m = \overline{1, n_m}, m = \overline{1, p},$$

таке, що

$$\text{Th}_{O^*} = \bigcup_{m=1}^p \text{Th}_{O_m};$$

– *тезаурус задачі* – множина термінів з множини X онтології O , сукупність яких характеризує ту конкретну задачу з $\text{Pr}O$, що в цей час розв'язує користувач (визначається шляхом співставлення онтології O з описом задачі),

$$\text{Th}_{i_j} = \{th_{k_{i_j}} \in X\}, k = \overline{1, s_{i_j}}, j = \overline{1, m_i};$$

– *зважений тезаурус задачі* – множина пар, першим елементом яких є термін з тезаурусу задачі, сукупність яких характеризує конкретну задачу з $\text{Pr}O$, а другим – вага (позитивна чи негативна) цього терміна для цієї задачі

$$\text{Tw}_{i_j} = \{ \langle th_{k_{i_j}} \in X_{i_j}, w_{k_{i_j}} \rangle \},$$

$$k = \overline{1, s_{i_j}}, j = \overline{1, m_i};$$

– *тезаурус IP* – підмножина термінів тезаурусу задачі, яким відповідають певні фрагменти контенту або метаопису IP

$$\text{Th}_{IR_q} = \{th_k \in \text{Th}_{i_j}\}, k = \overline{1, s_{i_j}}, q = \overline{1, z}$$

(слід відмітити, що для різних задач тезауруси того самого IP можуть значно відрізнятися);

– *зважений тезаурус IP* – множина пар, першим елементом яких є термін тезаурусу задачі, що містяться в контенті IP або в його метаописі, а другим – вага цього терміну для документа, яка визначається (за різними критеріями) як функція від кількості появ цього терміну в IP, місць його появи та від довжини документа

$$\text{Tw}_{IR_q} = \{th_k \in \text{Th}_{i_j}, w_k \rangle\},$$

$$k = \overline{1, s_{i_j}}, q = \overline{1, z};$$

– *тезаурус IO* – множина термінів тезаурусу задачі, що містяться в контенті IO або в його метаописі

$$\text{Th}_{IO_q} = \{th_k \in \text{Th}_{i_j}\}, k = \overline{1, s_{i_j}} -$$

такий опис дозволяє коректно співставляти різні типи IO та IO одного типу, але з різною семантикою із урахуванням їх структури (приміром, розрізняти Web-сервіси, якщо вхідні дані одного подібні до вихідних даних іншого);

– *зважений тезаурус IO* – множина пар, першим елементом яких є термін тезаурусу задачі, що містяться в контенті IO або в його метаописі, а другим – назва того елементу даного IO (з онтологічного опису IO), в якій зустрічається даний термін

$$\text{Tw}_{IO_q} = \{ \langle th_k \in \text{Th}_{i_j}, d_w \rangle \},$$

$$k = \overline{1, s_{i_j}}, k = \overline{1, s_{i_j}}, w = \overline{1, x_{IO}};$$

– *зважений тезаурус задачі користувача* – множина пар, першими елементами яких є терміни однієї або різних онтологій, сукупність яких характеризує інформаційні інтереси користувача, а другим – вага цього терміна для опису інтересів користувача

$$\text{Tw}_{user_j} = \{ \langle th_{k_{user_j}} \in \text{Th}_{PrO_i}, w_{k_{user_j}} \rangle \},$$

$$k = \overline{1, s_{user_j}},$$

де вага терміну визначається як функція (як правило, як сума добутків) від ваги певного ресурсу для користувача та кількості термінів у цьому ресурсі;

– *запит* – множина ключових слів, що характеризують одну з інформаційних потреб користувача, пов'язану з конкретною задачею, за допомогою тезауруса;

– *тема* – множина запитів, пов'язаних з однією інформаційною потребою, що дає змогу поєднувати семантично по-

в'язані запити різних користувачів, які базуються на різних онтологіях і тезаурусах;

– *результат запиту* – множина пар, першим елементом яких є посилання на IP, а другим – оцінки цих IP користувачем;

– *група користувачів* – клас, властивостями якого є ідентифікатор групи і список користувачів, які з певних причин об'єднані в одну групу (групи можуть формуватися шляхом вибору користувача безпосередньо чи автоматично на основі відповідності яким-небудь умовам, наприклад, групи користувачів з подібними формальними даними або таких, що виконують схожі запити);

– *IP* – клас, що описує відомості про відомі ППС ресурси (ідентифікатор ресурсу, запити, за якими він був виявлений, оцінку користувача, якому він був наданий, і його рівень читабельності для цього користувача) та оцінки цих ресурсів, надані різними користувачами

$$\langle U_{url}, \{ \langle z_i, m_i, q_i \rangle, i = \overline{1, n} \} \rangle;$$

– *IO* – клас, що описує відомості про відомі ППС IO з певною структурою, визначеною користувачем, що містяться в одному чи декількох IP (ідентифікатор IO, запити, за якими він був виявлений, оцінку користувача, якому він був наданий, і онтологію, що визначає структуру даного IO) та оцінки цих IO, надані різними користувачами

$$\langle IO_{url}, \{ \langle z_i, m_i, O_i \rangle, i = \overline{1, n} \} \rangle;$$

– *рекомендація* – інформація, що надається користувачеві ППС проактивно, як наслідок аналізу і персональних відомостей про цього користувача, і колаборативного досвіду системи.

– *агент користувача* – це інтелектуальний програмний агент, що презентує інтереси користувача у взаємодії з ППС та виконує певні дії в його інтересах.

Застосування такого формалізму, як агент користувача, дасть змогу, з одного боку, уникнути приписування людині-користувачу штучно обмеженої і формально схарактеризованої сфери інтересів, а

з іншого – забезпечить засоби та методи прогнозування його вчинків у межах моделі взаємодії користувача та ресурсів у відкритому інформаційному середовищі. Для опису поведінки такого агента використовуються інтенціональні відношення, за допомогою яких можна формалізувати цілі, наміри й бажання користувача. Таким чином, у ОМСП проводиться відмінність між самим користувачем: клас “користувач” відображає інформацію щодо фактів, пов'язаних з діями користувача, а клас “агент користувача” містить припущення щодо мотивації цих дій.

Для того, щоб описати екземпляри класів ОМСП з X_{ind} , необхідно спочатку формалізувати ті відношення, які для цього використовуються, та задати їх область значення та визначення.

Відношення між елементами ОМСП

Однією з основних переваг, яку забезпечує наявність використання онтологічного підходу до моделювання процесу пошуку, є можливість явно визначити семантику відношень між його основними елементами, тобто задати не тільки імена та визначення цих відношень, але й їх властивості. ОМСП визначає набір таких властивостей та їх характеристики, що впливають на складність даної моделі і визначають ту дескриптивну логіку, що дозволяє описати ОМСП.

Відповідно до специфіки проблеми пошуку, між суб'єктами та об'єктами цієї сфери існують наступні значущі для проблеми зв'язки:

- між IP і ПрО;
- між IP і IO;
- між інформаційними потребами й ПрО;
- між ПрО й задачами користувачів;
- між користувачами та ПрО;
- між користувачами ППС.

В ОМСП зв'язки відображаються за допомогою відношень з

$$R = r_{ier_cl} \cup \{r_i\} \cup \{p_j\},$$

які дозволяють визначити семантику, область значення та її визначення кожного такого зв'язку. Проаналізувавши властиво-

сті цих зв'язків, можна визначити, яка саме дескриптивна логіка лежить в основі ОМСП та, відповідно, наскільки складну мову для подання такої онтології необхідно використовувати. Це, в свою чергу, дозволяє визначити обчислювальну складність задач, які можна вирішувати з використанням такої онтології.

Відношення «клас-підклас»

Ієрархічні відношення «клас-підклас» r_{ier_cl} (приміром, «експерт» є підкласом класу «користувач», «мультимедійний інформаційний об'єкт» є підкласом класу «ІО») є транзитивними та антисиметричними:

- якщо X належить до класу A , а A є підкласом B , то X належить до B ;
- якщо A є підкласом B , B є підкласом C , то A є підкласом C ;
- якщо A є підкласом B , то B не є підкласом A .

За допомогою таких відношень не відображаються мереологічні зв'язки різних типів – приміром, відношення “є членом групи” не можна відображати таким чином, тому що екземпляр класу “користувач” не є екземпляром класу “група користувачів”. Це викликає потребу включити до ОМСП інші ієрархічні відношення із специфічною для ПрО специфікою.

Відношенням «клас-підклас» в ОМСП пов'язані такі класи (табл. 1):

Таблиця 1. Ієрархічні властивості в ОМСП

Надклас	Підклас
Онтологія	Онтологія ПрО; Таксономія ІО; Лексична онтологія; Тезаурус; Wiki-онтологія
Тезаурус	Тезаурус задачі; Тезаурус користувача; Тезаурус ІР; Тезаурус ІО
Користувач	Експерт; Член групи
ІР	Природномовний ІР; Мультимедійний ІР; Семантично розмічений ІР
ІО	Людина; Організація; Документ; Web-сервіс

Об'єктні властивості ОМСП

Відношення, специфічні для цієї предметної області – відношення семантичного пошуку, що виражаються через властивості класів, значеннями яких є екземпляри інших класів (приміром, клас «тезаурус» має властивість «побудований на основі», значення якого належить до класу «онтологія ПрО», а клас «тема» має властивість «містить», значення якого належить до класу «запит». В даній ПрО не визначені специфічні відношення, які мають властивості, що можуть застосовуватися для логічного виведення (транзитивність, рефлексивність, симетричність тощо). Такі відношення, в яких і область значення, і область визначення є екземплярами класів ОМСП, з точки зору онтологічного аналізу відповідають об'єктним властивостям $\{r_i\}$ відповідної онтології (табл. 2).

Таблиця 2. Об'єктні властивості в ОМСП

Область значень	Відношення	Область визначення
Користувач	Використовує	Онтологія ПрО; Онтологія ІО; Тезаурус задачі; Тезаурус онтології; Тезаурус користувача
Тезаурус ІР; Тезаурус ІО; Тезаурус задачі; Лексична онтологія ПрО; Тезаурус користувача; Запит; Рекомендація	Базується на	Онтологія ПрО; Онтологія ІО; Тезаурус задачі; Запит
Тема; Користувач; Результат запиту	Є об'єднанням	Запит; Група користувачів; ІО; ІР
Результат запиту; Рекомендація	Є результатом	Запит
Агент користувача	Є представником	Користувач; Група користувачів

Через те, що ОМСП створюється для формалізації вже відомих відношень, а не для впорядкування термінології, то недоцільно створювати класи-синоніми: альтернативні назви понять можна вказувати тільки у поясненні або у визначенні класу.

Мереологічні відношення в ОМСП, що відображають різні види специфічних для ПрО зв'язків типу “частина-ціле” (приміром, відношення «входить до складу» пов'язує екземпляри класу «Р» з екземплярами класу «результати пошуку», а екземпляри класу «користувач» з екземплярами класу «група користувачів») в загальному випадку не є транзитивними.

Таким чином, для сфери семантичного пошуку не виявлено важливих транзитивних або симетричних відношень між екземплярами одного класу. Приміром, якщо користувач А вважає експертом користувача В, а користувач В вважає експертом користувача С, то з цього не випливає, що користувач А вважає експертом користувача С. Це пов'язано з тим, що, як правило, екземпляри одного класу не взаємодіють безпосередньо один з одним в процесі пошуку, а їх відношення можуть встановлюватися тільки через відношення з екземплярами інших класів. Приміром, екземпляри класу “користувач” можуть бути пов'язані через екземпляри класу “тезаурус”, що використовуються у запиті, або через екземпляри класу “Р”, що є результатами пошукової процедури.

Властивості даних ОМСП

Такі відношення, в яких область значення є екземплярами класів ОМСП, а область визначення – іншими типами даних, з точки зору онтологічного аналізу відповідають даних властивостям $\{p_i\}$ відповідної онтології (табл. 3).

Властивості даних в ОМСП дозволяють встановити конкретні значення властивостей екземплярів класів, явно вказавши їх семантику та характеристики. Вказуючи тип значення властивості, можна не тільки задавати стандартні типи даних (число, рядок тощо), але й задати значення із скінченної множини, описавши таким чином всі припустимі варіанти та вказав-

ши відношення між цими значеннями (приміром, часткову впорядкованість або синонімію). Прикладами таких множин можуть бути різні варіанти подання дати або часу, що надалі будуть інтерпретуватися однаково.

Ці відношення не мають додаткових властивостей, які можуть враховуватися в процесі обробки онтології, і тому не впливають на складність ОМСП.

На основі обробки типів цих значень будується множина T для ОМСП.

Онтологічна модель задачі користувача

Формально проблема побудови онтології задачі користувача полягає у наступному: за онтологією ПрО O_{domain} ,

$$O_{domain} = \langle X_{domain}, R_{domain}, F_{domain}, T_{domain} \rangle,$$

та набором Wiki-сторінок W_{user} , семантична розмітка яких базується на O_{domain} , побудувати “легковажну” онтологію задачі користувача O_{user} , знання якої є підмножиною знань з O_{domain} . Слід зазначити, що джерела та методи побудови цієї онтології ПрО знаходяться поза сферою розгляду даної роботи – вона може мати довільну структуру та бути сформована як безпосередньо експертами ПрО, так і за допомогою різноманітних засобів здобуття онтологічних знань [6].

$$O_{user} = \langle X_{user}, R_{user}, F_{user}, T_{user} \rangle,$$

така, що

$$X_{user} \subseteq X_{domain},$$

тобто

$$X_{cl_{user}} \subseteq X_{cl_{domain}},$$

$$X_{ind_{user}} \subseteq X_{ind_{domain}}; R_{user} \subseteq R_{domain},$$

тобто

$$r_{ier_cl_{user}} = r_{ier_cl_{domain}},$$

$$\{r_{user_j}\} \subseteq \{r_{domain_i}\}, i = \overline{0, n}, j = \overline{0, m}, m \leq n;$$

$$\{p_{user_k}\} \subseteq \{p_{domain_l}\}, l = \overline{0, q}, k = \overline{0, t}, t \leq q,$$

$$F_{user} = \emptyset; T_{user} \subseteq T_{domain}.$$

Таблиця 3. Властивості даних в ОМСП

Область значень	Відношення	Область визначення		
		Назва	Тип	Кількість значень
Користувач	Використовує	Ключове слово	Рядок	Обов'язкове, єдине
Користувач	Має ідентифікатор	Ідентифікатор	Рядок	Обов'язкове, єдине
Користувач	Має пароль	Пароль	Рядок	Обов'язкове, єдине
Користувач	Має сайт	Сайт	Рядок	Не обов'язкове, єдине або кілька
Користувач	Має сферу інтересів	Ключове слово	Рядок	Не обов'язкове, єдине або кілька
Користувач	Має професію	Професія	Рядок	Не обов'язкове, єдине або кілька
Користувач	Має освіту	Освітній рівень	Рядок, значення з множини {неповна середня, середня, вища, вчений ступінь}	Не обов'язкове, єдине
Користувач	Рік народження	Рік	Число з 4 знаків	Не обов'язкове, єдине
Користувач	Живе у	Країна Місто Населений пункт	Рядок	Не обов'язкове, єдине або кілька
Користувач	Народився у	Країна Місто Населений пункт	Рядок	Не обов'язкове, єдине або кілька
Онтологія ПрО	Містить	Термін	Рядок	Обов'язкове, єдине або кілька
Тезаурус	Містить	Термін	Рядок	Обов'язкове, єдине або кілька
Запит	Містить	Ключове слово	Рядок	Обов'язкове, єдине або кілька

Така робота має виконуватися в тому разі, якщо користувач починає працювати над великою та досить складною проблемою, рішення якої буде потребувати інформації протягом досить значного часу, значно більшого, ніж час, потрібний на побудову власної онтології (приміром, плануючи дослідження на кілька років,

доцільно витратити кілька годин на те, щоб надалі отримувати семантично відфільтровані відомості).

Множина екземплярів класів ОМСП

ОМСП поповнюється екземплярами класів у процесі функціонування системи пошуку. Наприклад, екземпляри кори-

стувачів створюються внаслідок реєстрації користувачів у системі та можуть доповнюватися новими значеннями властивостей у процесі виконання користувачами пошукових запитів, тоді як екземпляри груп користувачів створюються самими користувачами відповідно до їх власних інформаційних потреб.

Використання онтологічних знань у персоніфікованому семантичному пошуку

У загальному випадку співставлення двох незалежних онтологій, які знаходяться в репозиторії онтологій [11], є надзвичайно складною задачею, що потребує багато часу та додаткової обробки. Але в інформаційному пошуку використовуються онтологічні моделі, які мають достатню для задачі, але досить обмежену складність. Такі моделі можуть використовувати знання з довільних онтологій ПрО, але містити тільки обмежену їх підмножину та не застосовувати складний набір відношень між класами та атрибутами (але сам алгоритм побудови таких спрощених моделей за довільними онтологіями може бути досить складним та знання-містким).

Застосування ОМСП у процесі семантичного пошуку

Відомості, що представлені в ОМСП, використовується на різних етапах семантичного пошуку для (табл. 4):

- переформулювання запитів користувачів відповідно до їх реальних інформаційних потреб;
- фільтрації результатів пошуку, отриманих від зовнішніх ПС, відповідно до їх пертинентності поточним інформаційним потребам користувача;
- використання досвіду співтовариства користувачів з областями інформаційних потреб, що перетинаються, для проактивного пошуку та надання рекомендацій;
- оцінка відповідності рівня складності контенту знадених ІР здатностям користувача до сприйняття інформації в обраній ПрО.

Таблиця 4. Екземпляри класів в ОМСП

Тип операції	Екземпляри класів ОМСП
Переформулювання запитів користувачів	Користувач, Онтологія ПрО, Тезаурус онтології, Лексична онтологія ПрО, Тезаурус задачі, ІО, Тезаурус ІО, Запит, Група запитів
Фільтрація результатів	Користувач, Онтологія ПрО, Зважений тезаурус онтології, Лексична онтологія ПрО, Зважений тезаурус задачі, Зважений тезаурус ІР, Зважений тезаурус ІО
Використання досвіду співтовариства користувачів	Користувач, Група користувачів, Тема, Онтологія ПрО, Тезаурус онтології, Лексична онтологія ПрО, Тезаурус задачі, Тезаурус множини онтологій, Запит, Група запитів, Результат запити, Агент користувача
Оцінка рівня складності контенту	Тезаурус ІО Тезаурус користувача

Слід враховувати, що така модель інформаційного пошуку орієнтована на користувачів із сталими та досить глибоко усвідомленими інформаційними потребами, тому процес пошуку інформації може розглядатися як циклічний процес, різні етапи якого повторюються у певній послідовності, а наповнення ОМСП екземплярами класів продовжується протягом усієї взаємодії користувача з системою.

Алгоритми такого застосування елементів онтології та їх властивості розглянуті у наступних розділах.

Семантичний пошук на основі зіставлення тезаурусів

Будемо вважати, що сферу інтересів користувача в цілому формально характеризує онтологія відповідної ПрО (або набір таких онтологій, що відповідають різним аспектам діяльності однієї особи), а його поточні інтереси – природномовний опис задачі.

Природномовний опис задачі – це неструктуровані або слабо структуровані дані, аналіз яких потребує попередньої обробки, а онтологія ПрО у загальному випадку має довільний розмір та структуру, що надзвичайно ускладнює її безпосереднє використання у пошуку. Тому за обома цими об'єктами пропонується будувати *тезаурус задачі*, що поєднає їх переваги та дозволяє позбутися недоліків.

Для того, щоб відфільтрувати результати роботи зовнішньої ПС і отримати тільки ті ІР, що пертинентні інформаційним потребам користувача, необхідно попередньо сформувавши тезаурус задачі користувача та ПрО, що цікавить цього користувача, і тезауруси цих ІР, а потім порівняти ці тезауруси.

Алгоритм побудови простого тезаурусу задачі

Простий тезаурус задачі

$$Th = \langle T, r_{ier_cl}, \emptyset, \emptyset \rangle$$

будується за обраною користувачем онтологією ПрО та за описом поточної задачі (рисунок).

Опис задачі може бути подано через ПМ-текст, який містить елементи, пов'язані з елементами онтології, або через умови, яким мають задовольняти терміни ПрО, що стосуються цієї задачі. Розглянемо два етапи побудови такого тезаурусу.

Етап 1. Автоматизована генерація простого тезаурусу задачі за описом задачі.

Етап 2. Розширена генерація простого тезаурусу задачі за набором умов, що використовують інші елементи онтології О, крім екземплярів та класів.

На етапі 1 теж можна виокремити два кроки (на практиці може застосовуватися їх поєднання). Етап 1.1 полягає у тому, що користувач явно та вручну з автоматично побудованого переліку класів та екземплярів X обирає ті, які він вважає пертинентними задачі:

$$T = \{x_{t_1}, \dots, x_{t_p}\}, 1 \leq t_k \leq n, \forall x_{t_k} \in X.$$

В найпростіших випадках на цьому кроці побудова тезаурусу може завершуватися, але це потребує від користувача більше зусиль.

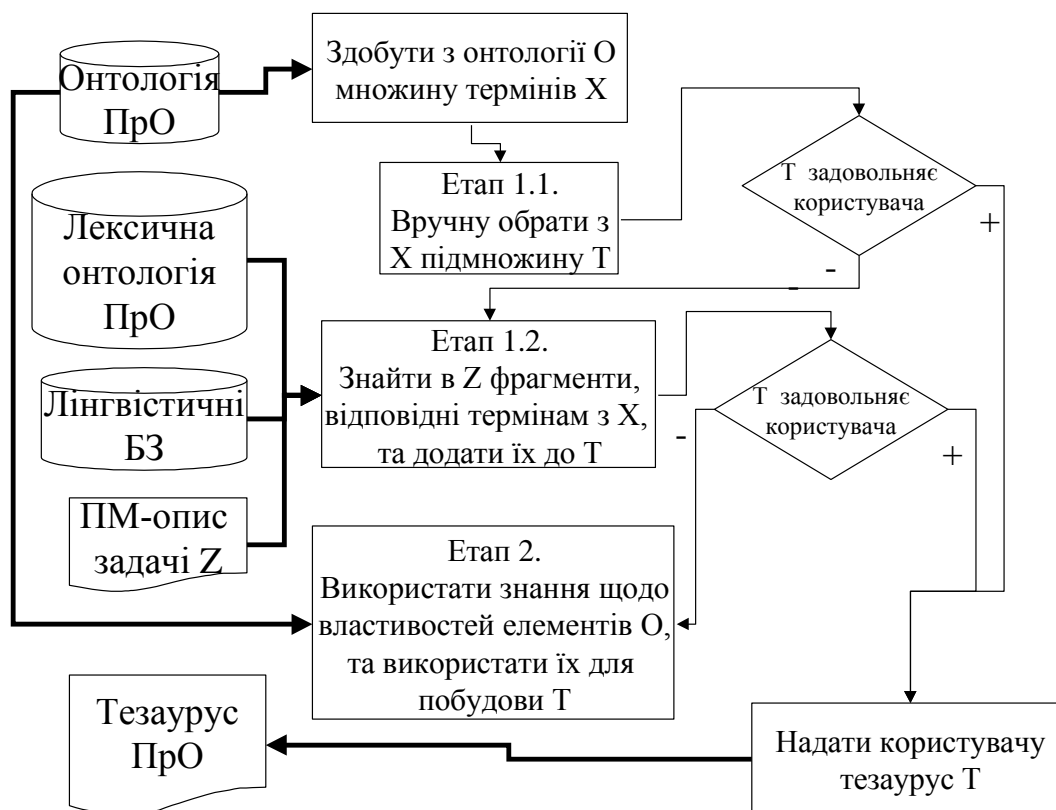


Рисунок. Алгоритм побудови тезаурусу задачі

Етап 1.2 використовує різноманітні методи обробки природномовного опису задачі (лінгвістичний аналіз, статистична обробка, аналіз семантичної розмітки), які дозволяють виявити фрагменти ПМ-тексту, пов'язані з термінами O .

Ті терміни, для яких в описі задачі знайдено відповідні фрагменти, заносяться до простого тезаурусу задачі. На цьому етапі використовується множина X онтології O . Один з методів обробки ПМ-опису задачі Z базується на лексичній онтології

$$L_{PrO} = \langle X_{lex} = X_{PrO} \cup T_{PrO}, \{r_{lex}\}, \emptyset \rangle:$$

якщо

$$s_{jp} \in T_{PrO}, p = \overline{1, q_j}$$

є фрагментом ПМ Z , та

$$r_{lex}(s_{jp}) = x_j \in X,$$

тоді x_j додається до множини T тезаурусу задачі.

Такий метод високо ефективний у тому разі, якщо вже накопичено великий обсяг лексичної онтології. Для цього можуть використовуватися безпосереднє поповнення лексичної онтології користувачами, експорт лінгвістичних знань з відповідних словників та баз знань, а також аналіз семантично розмічених текстів, що буде розглянуто далі.

Але у багатьох випадках доцільно використовувати у побудові тезаурусу інформацію про інші елементи онтології, які дозволяють враховувати властивості окремих термінів та їх відношення з іншими термінами. В такому разі застосовується етап 2, який спрямований на вдосконалення початково сформованого тезаурусу відповідно до явно сформульованих умов користувача. Ці умови обумовлені специфікою задачі, але не є похідними від її опису. Їх можна розглядати як набір метаправил для опису інформації, яку користувач прагне отримати.

Етап 2 можна представити як функцію, що перетворює онтологію O на простий тезаурус

$$f_{Th}(O) \rightarrow Th,$$

є відображенням набору умов, які формулює користувач щодо тих класів та екземплярів класів онтології O , що потрібно включити до тезаурусу задачі.

Набір цих умов можна розглядати як об'єднання (диз'юнкцію) d множин властивостей елементів онтології (класів та об'єктів)

$$f_{Th}(O) = \cup_{i=1}^d f_{Th_i}(O),$$

що можуть пов'язувати кожен елемент, що задовольняє цим вимогам, їх з фіксованими значеннями або іншими елементами онтології, – кон'юнкція вимог

$$f_{Th_i}(O) = \cap_{j=1}^{d_i} f_{Th_{ij}}(O).$$

Потрібно спочатку здобути з онтології набори для всіх

$$f_{Th_i}(O), i = \overline{1, d},$$

а потім побудувати їх об'єднання. Усі $f_{Th_i}(O), i = \overline{1, d}$ будуються наступним чином: оброблюється окремо кожна умова, за нею генеруються d_i наборів елементів (класів та екземплярів) онтології O , після цього побудувати перетин цих множин.

Побудова множини елементів тезаурусу для кожної умови

$$f_{Th_{ij}}(O), i, j = \overline{1, d_{ij}}$$

складається з наступних дій, що послідовно враховують інформацію з усіх елементів онтології O .

Множина класів X_{cl} є джерелом тих термінів PrO , що стосуються поточного набору задачі користувача, і кожен з цих термінів може бути доданий до $Th_{ij}(O)$ користувачем безпосередньо або через перерахування в умовах. Окремо оброблюються негативні умови щодо класів – кожен з цих термінів може бути видалений з $Th_{ij}(O)$ користувачем безпосередньо або через перерахування в умовах.

Множина класів X_{ind} є джерелом тих екземплярів об'єктів PrO , що стосуються поточного набору задачі користувача, і кожен з цих термінів може бути доданий до $Th_{ij}(O)$ користувачем безпосередньо або через перерахування в умовах. Можуть також додаватися до тезаурусу всі екземпляри певного класу. Окремо обробляються негативні умови щодо екземплярів класів – кожен з цих термінів може бути видалений з $Th_{ij}(O)$ користувачем безпосередньо або через перерахування в умовах.

Множина відношень між елементами онтології

$R = r_{ier_cl} \cup \{r_i\} \cup r_{ier_prop} \cup \{p_j\} \cup r_{ier_prop}$ може бути використана для побудови $Th_{ij}(O)$ наступним чином:

- ієрархічні відношення r_{ier_cl} дозволяють користувачу включити (або видалити – для негативних умов) до тезаурусу $Th_{ij}(O)$ надкласи та підкласи обраних класів, задавши глибину обробки, наприклад, всі підкласи терміну “наукова публікація” на глибину q ;

- об'єктні властивості $\{r_i\}$, що встановлюють відношення між екземплярами класів, дозволяють користувачу включити (або видалити – для негативних умов) до тезаурусу $Th_{ij}(O)$ ті терміни, що пов'язані з обраними термінами і відношеннями з $\{r_i\}$, наприклад, для терміну x включити до $Th_{ij}(O)$ ті терміни, з якими в онтології O цей термін пов'язаний об'єктним відношенням “співпрацює з”;

- ієрархічні відношення між об'єктними властивостями r_{ier_prop} дозволяють використовувати підкласи відношень замість самих відношень з $\{r_i\}$, наприклад, якщо для терміну x потрібно включити (або видалити – для негативних умов) до $Th_{ij}(O)$ ті терміни, з якими в

онтології O цей термін пов'язаний об'єктним відношенням “співпрацює з” з множини $\{r_i\}$, то потрібно включити до $Th_{ij}(O)$ також ті терміни, з якими в онтології O цей термін пов'язаний об'єктним відношенням “працює в одному відділі”, яке є підкласом відношенням “співпрацює з”;

- властивості даних $\{p_j\}$, що встановлюють відношення між екземплярами класів та даними, дозволяють користувачу включити (або видалити – для негативних умов) до тезаурусу $Th_{ij}(O)$ ті терміни, що задовольняють певним умовам, в яких задаються значення властивостей даних з $\{p_j\}$, наприклад, включити (або видалити – для негативних умов) до $Th_{ij}(O)$ ті терміни, які в онтології пов'язані відношенням даних “Рік народження” більшим за 1900;

- ієрархічні відношення між властивостями даних r_{ier_prop} дозволяють використовувати підкласи відношень замість самих відношень з $\{p_j\}$, наприклад, якщо для терміну x потрібно включити (або видалити – для негативних умов) до $Th_{ij}(O)$ ті терміни, що задовольняють певним умовам, в яких задаються значення властивостей даних з $\{p_j\}$, потрібно включити (або видалити – для негативних умов) до $Th_{ij}(O)$ ті терміни, які в онтології

задовольняють умовам щодо підкласів цих властивостей, наприклад, якщо є умова щодо відношення даних “Кількість публікацій” більше за 10, то потрібно включити до тезаурусу елементи, для яких “Кількість публікацій Scopus” більше за 10;

- множина характеристик класів онтології F_{cl} , що можуть застосовуватися для логічного виводу, обробляються в процесі побудови тезаурусів відповідно до того, яку саме властивість вони фіксують: якщо два класи x_1 та x_2 в онтології O еквівалентні, і клас x_1 занесено до тезаурусу

урусу $Th_{ij}(O)$, тоді треба занести до тезаурусу й клас x_2 ;

– множина характеристик об'єктних властивостей екземплярів класів онтології F_{prop} , що можуть застосовуватися для логічного виводу, обробляються в процесі побудови тезаурусів відповідно до того, яку саме властивість вони фіксують (на практиці це зазвичай не застосовується);

– множина нелогічних правил PrO використовується для побудови тезаурусу наступним чином: якщо об'єкт з онтології O (клас або екземпляр класу) o_1 належить до тезаурусу $o_1 \in Th$ та в онтології в M міститься правило “якщо $o_1 \in O$, тоді $o_1 \in O$ ”, треба додати до тезаурусу елемент o_2 . Приклад: в тезаурус треба додати усі екземпляри осіб пенсійного віку (тобто тих, вік яких більше певної константи). Якщо вік особи невідомий, але відомо, що вона має дитину пенсійного віку, а за нелогічними правилами PrO вік батьків більше за вік дитини, тоді треба додати до тезаурусу цей екземпляр класу.

Інші елементи онтології O не використовуються безпосередньо для побудови Th_O , але вони можуть застосовуватися для поповнення та вдосконалення самої онтології O .

Алгоритм побудови складеного тезаурусу задачі

Складений тезаурус задачі будується на основі простих або складних тезаурусів задач. Для цього застосовуються теоретико-множинні операції перетину, об'єднання та різниці.

Якщо

$$Th_1 = \langle T_1, r_{ier_cl}, \emptyset, \emptyset \rangle$$

та

$$Th_2 = \langle T_2, r_{ier_cl}, \emptyset, \emptyset \rangle$$

– прості тезауруси задачі, і $Th_1 \neq Th_2$, тоді їх об'єднання

$$Th_{об} = \langle T_1 \cup T_2, r_{ier_cl}, \emptyset, \emptyset \rangle,$$

перетин

$$Th_{перет} = \langle T_1 \cap T_2, r_{ier_cl}, \emptyset, \emptyset \rangle$$

та різниця

$$Th_{різ} = \langle T_1 / T_2, r_{ier_cl}, \emptyset, \emptyset \rangle$$

– складені тезауруси.

Якщо $Th_1 = \langle T_1, r_{ier_cl}, \emptyset, \emptyset \rangle$ та

$Th_2 = \langle T_2, r_{ier_cl}, \emptyset, \emptyset \rangle$ – складені тезауруси задачі, і $Th_1 \neq Th_2$, тоді їх об'єднання $Th_{об} = \langle T_1 \cup T_2, r_{ier_cl}, \emptyset, \emptyset \rangle$,

перетин $Th_{перет} = \langle T_1 \cap T_2, r_{ier_cl}, \emptyset, \emptyset \rangle$ та різниця

$Th_{різ} = \langle T_1 / T_2, r_{ier_cl}, \emptyset, \emptyset \rangle$ теж є складеними тезаурусами.

Для того, щоб побудувати складений тезаурус, користувачеві потрібно обрати два раніше створених тезауруси та визначити, яку саме теоретико-множинну операцію треба до них застосувати.

Використання складених тезаурусів дозволяє застосовувати тезаурусне представлення знань PrO , що цікавить певного користувача, у багатьох задачах, пов'язаних з пошуком, аналізом та структуруванням ресурсів Web, відображаючи персональні уявлення окремого користувача щодо сфери його інформаційних потреб.

На змістовному рівні такий тезаурус – це сукупність термінів PrO , відомих користувачеві, тобто користувач обирає лише ту підмножину онтологічних термінів різних онтологій, які відповідають його особистим інтересам та уявленням.

Такий тезаурус може застосовуватися не тільки безпосередньо в процесі пошуку, але він є зручним інструментом для розширення функціоналу семантичної ПС. Наприклад, тезаурус PrO дозволяє виконувати наступні операції, алгоритми здійснення яких більш детально описані в наступних розділах:

– оцінка складності природномовного тексту для сприйняття конкретним користувачем;

- побудова груп користувачів з подібними інформаційними потребами для рекомендуєчих систем;
- виконання теоретико-множинних операцій над тезаурусами, що забезпечують повторне використання тезаурусів для нових задач;
- побудова та використання лексичних онтологій.

*Алгоритм побудови
зваженого тезаурусу задачі*
Зважений тезаурус задачі

$$T_w = \{ \langle t_j, w_j \rangle, t_j \in T, j = \overline{1, s} \}$$

будується за множиною

$$T = \{ x_j \in X, j = \overline{1, s} \}$$

тезаурусу задачі T_h (простим або складеним) наступним чином: кожному елементу з T ставиться у відповідність вага (позитивна чи негативна) w_j – кількісна характеристика важливості цього терміну для поточної задачі користувача. Ця оцінка може задаватися користувачем явно (якщо зважений тезаурус будується за простим або складеним тезаурусом) або обчислюватися за раніше заданими оцінками (якщо зважений тезаурус будується як об'єднання двох раніше побудованих тезаурусів, тоді значення оцінок термінів підсумовуються).

*Аналіз виразної здатності
тезаурусу задачі*

Для того, щоб використовувати запропонований підхід, необхідно довести, що виразна здатність таких тезаурусів є задовільною для виконання семантичного пошуку. Незважаючи на досить просту структуру самого тезаурусу задачі, його виразна здатність визначається методом його побудови, який використовує всі ті знання PrO , що містяться у відповідній онтології та можуть бути застосовані для пошуку у тому випадку, якщо б у співставленні задачі користувача та IP використовувалися б довільні онтології.

Слід зазначити, що в інших моделях пошуку можуть застосовуватися інші аспекти онтологічних знань, що не вико-

ристовуються у цій моделі. Крім того, деякі моделі пошуку дозволяють користувачеві явно керувати тим, які саме знання треба враховувати в процесі пошуку (і запропонована модель належить саме до цього класу), тоді як інші моделі не дозволяють користувачу впливати на такий вибір. Переваги та недоліки окремих моделей знаходяться поза розглядом даної роботи.

Можна стверджувати, що певні знання з онтології *зафіксовані* у тезаурусі, якщо їх відсутність в онтології призвела б до таких змін у тезаурусі, побудованому за цією онтологією, які вплинули б на результати пошуку. Але таке визначення не дозволяє оцінити виразну здатність тезаурусу. В процесі пошуку виконується співставлення моделі задачі та тезаурусу задачі. В цьому співставленні аналізуються тільки ті елементи тезаурусу задачі, для яких знайдені певні відповідності у моделі задачі. Тому ті елементи тезаурусів онтологій, які не входять до тезаурусу задачі, не впливають на впорядкування результатів пошуку. Якщо у тезаурусі онтології можна відобразити певний елемент знань онтології, то ці знання можуть бути відображені в тезаурусі задачі.

Тому будемо вважати, що якщо певний елемент онтології може вплинути на вміст тезаурусу, то виразна здатність тезаурусу є достатньою для його відображення стосовно пошукової процедури.

Твердження 1. Алгоритм побудови простого тезаурусу задачі дозволяє використовувати знання щодо структури PrO з онтології, яку користувач вважає пертинентною його поточної задачі, відповідно до тих умов, які користувач вважає доцільним застосовувати в обраній пошуковій моделі.

Доказ. Проаналізуємо окремо кожен компонент онтології O , його використання для пошуку та те, як цей компонент відображено в простому тезаурусі задачі

$$Th_O = \langle Th_{cl} \cup Th_{ind} \rangle.$$

Розглянемо дане для обох етапів алгоритму побудови простого тезаурусу задачі.

Слід зазначити, що на етапі 1 використовується лише множина X онтології

О. Завдяки цьому обчислювальна складність даного алгоритму не вище, ніж лінійна залежність від кількості класів та екземплярів класів в онтології О.

На етапі 2, крім класів та екземплярів онтології О, інші елементи використовуються не безпосередньо, а для поповнення та вдосконалення самої онтології О, тому обчислювальна складність побудови тезаурусу залежить від кількості та типу умов (наприклад, умов “включити всі екземпляри зі значенням “рік створення” більшим за 2000”), до яких входять ці елементи, та від розміру відповідних множин онтології О (наприклад, в онтології ПрО може бути 3 або 33 властивості даних).

Відповідно до алгоритму побудови простого тезаурусу задачі, для побудови тезаурусу можуть бути застосовані:

- ієрархічні відношення для розширення тезаурусу надкласами та підкласами обраних класів на обрану глибину обробки;
- об’єктні властивості для розширення тезаурусу екземплярами класів, що пов’язані з вже обраними екземплярами класів певними об’єктними відношеннями;
- ієрархічні відношення між об’єктними властивостями для розширення тезаурусу, використовувати підкласи об’єктних відношень замість відношень, визначених у попередньому пункті;
- властивості даних екземплярів класів для розширення тезаурусу тими термінами, що задовольняють певним умовам;
- ієрархічні відношення між властивостями даних розширювати тезаурус, використовуючи підкласи відношень даних замість визначених користувачем відношень даних;
- характеристики класів онтології для використання еквівалентних класи замість класів, обраних користувачем;
- характеристики об’єктних властивостей екземплярів класів для логічного виводу в процесі побудови тезаурусів відповідно до того, яку саме властивість вони фіксують;

– нелогічні правила ПрО для розширення тезаурусу, якщо задовольняються умови цих правил.

Слід відмітити, що для довільної онтології та довільного набору умов користувача досить складно оцінити час побудови тезаурусу, точніше, можна спрогнозувати найгірший варіант, але на практиці час обробки значно менший. Доцільніше аналізувати окремі випадки як онтологій, так і умов. Надалі більш детально буде оцінено обчислювальна складність побудови простого тезаурусу задачі за Wiki-онтологією – онтологією, яка будується на основі семантичної розмітки Wiki-сторінок.

Твердження 2. Алгоритм побудови складеного тезаурусу задачі дозволяє використовувати ту частину знання щодо структури ПрО, які застосовувалися в алгоритмі побудови тезаурусів онтологій, що безпосередньо пов’язані з поточною задачею користувача та можуть бути співставленні з фрагментами її природномовного опису (безпосередньо, з використанням логічного виведення або специфічних для ПрО правил) у тому випадку, коли складений тезаурус будується за простими тезаурусами суттєво різних ПрО.

Доказ. На відміну від операцій перетину, об’єднання та різниці онтологій, алгоритм побудови складеного тезаурусу працює дуже швидко, а його обчислювальна складність залежить лише від розміру простих тезаурусів.

Можна вважати, що простий тезаурус, побудований за об’єднанням онтологій, містить ту саму інформацію для пошуку, що й об’єднання простих тезаурусів, що побудовані за кількома незалежними онтологіями (це визначається алгоритмом побудови простого тезаурусу задачі), але побудова об’єднання простих тезаурусів потребує значно менше часу.

Для випадку, коли поєднуються незалежні онтології, це впливає з алгоритму побудови простого тезаурусу: та частина об’єднаної онтології, що була побудована внаслідок об’єднання, не використовується в операціях поповнення тезаурусу, тому що вона не містить відповідних

термінів і не задовольняє пов'язаних з ними умовам.

Якщо поєднуються пов'язані онтології, то можливі три основні варіанти, суттєві для побудови тезаурусу:

– деякі онтологічні знання дублюються в різних онтологіях, і тому це не впливає на побудову тезаурусу;

– деякі онтологічні знання суперечать і не можуть бути поєднані, тобто в об'єднаній онтології обирається лише один з кількох можливих варіантів зв'язку між термінами, і тоді у побудові тезаурусу використовуються ті знання, які користувач вважає більш пертинентними. Якщо будуватиметься об'єднання тезаурусів таких онтологій, то такий спосіб виведення, що привів до додання певного терміну до результуючого тезаурусу, буде присутній хоча б одній з тих онтологій, що об'єднуються;

– деякі онтологічні знання не суперечать між собою, але їх об'єднання надає нові зв'язки між термінами онтології. В такому випадку тезаурус задачі, побудований за об'єднаною онтологією, може містити деякі терміни, що відсутні в усіх тезаурусах, побудованих за тими онтологіями, що об'єднуються. Але через те, що обробка менших онтологій на етапі 1.1 значно простіша, на практиці ці терміни досить часто користувач додає вручну.

Таким чином, запропонований підхід є ефективним тільки для онтологій, що описують різні ПрО (або суттєво різні аспекти ПрО) і тому мають набори термінів (класів та екземплярів), що не перетинаються. Аналогічно оцінюються перетин та різниця онтологій та побудованих за ними тезаурусів.

Твердження 3. Обчислювальна складність першого етапу алгоритму побудови простого тезаурусу за онтологією O лінійно залежить від кількості термінів у множині X онтології O , обсягу лексичної онтології та розміру опису задачі.

Доказ. Відповідно алгоритму побудови тезаурусу, на етапі 1.1. користувач проглядає весь перелік термінів онтології O та для кожного з n елементів у множині

X онтології O приймає рішення щодо того, чи занести цей елемент до тезаурусу. Від інших параметрів онтології та задачі цей етап не залежить явно. Тому, якщо вважати швидкість прийняття рішення користувачем за постійну величину, то обчислювальна складність етапу 1.1. не перевершує n .

На етапі 1.2 виконується співставлення елементів множини X з природним описом задачі за допомогою

$$L_{\text{ПрО}} = \langle X_{\text{lex}} = X \cup T, \{r_{\text{lex}}\}, \emptyset \rangle$$

– лексичної онтології, в якій кожному елементу (класу або екземпляру класу) з X $\forall x_j \in X, j = \overline{1, m}$ відповідає скінчена кількість фрагментів ПМ

$$s_{jp} \in T_{\text{ПрО}_i}, p = \overline{1, q_j}, r_{\text{lex}}(s_{jp}) = x_j,$$

що співвідносяться з цим елементом.

Для елементів множини X виконується $\sum_{j=1}^n q_j$ перевірок-співставлень для

кожного фрагмента, час виконання яких залежить від розміру опису задачі l , якщо

$q = \max_{j=1}^n q_j$, то обчислювальна складність

алгоритму для етапу 1.2 не більш як $n \cdot q \cdot l$.

Твердження 4. Алгоритм другого етапу побудови простого тезаурусу за онтологією O є скінченим, і його обчислювальна складність залежить лінійно від кількості термінів у множинах X , M та R онтології O , кількості вимог та обмежень, за якими елемент з X може бути додано до тезаурусу, рівня вкладеності вимог та кількості елементів.

Доказ. Скінченність алгоритму впливає із скінченності множин елементів та умов щодо них, що перевіряються. Виникненню циклів запобігає те, що кожна перевірка для кожного елемента не виконується більше одного разу: на кожному кроці виконання спочатку визначається множина елементів, які потрібно перевірити, – як об'єднання всіх множин, що задовольняють початковим умовам, а потім

з цієї множини видаляються елементи, що не задовольняють обмеженням.

Для кожного елемента множини X виконується перевірка для кожної з q вимог, що відображають переконання користувача щодо цікавлячої його ПрО відповідно до алгоритму виконання етапу 2.

Більш точно оцінювати обчислювальну складність алгоритму побудови тезаурусу доцільно для окремих випадків онтологій ПрО, що використовуються у практичних задачах, наприклад, для Wiki-онтологій, що будуть розглянуті далі.

Алгоритм побудови тезаурусів IP

Побудова тезаурусів природномовних IP дозволяє здобути з неструктурованих текстів відомості, що стосуються тієї задачі, яка цікавить користувача.

Для цього можуть використовуватися лексичні онтології або різноманітні інші засоби лінгвістичного аналізу. Слід зазначити, що лексична онтологія містить відносно невелику підмножину знань щодо ПМ-представлення термінів, пертинентних задачі користувача, і тому час аналізу тексту на її основі має обчислювальну складність, що залежить від розміру лексичної онтології, побудованої для тезаурусу задачі (простого або складеного).

Тезаурус IP

$$Th_{IR} = \langle X_{IR} \subseteq X_{Th}, \emptyset, \emptyset, \emptyset \rangle$$

це підмножина тезаурусу задачі

$$Th = \langle X_{Th} \subseteq X, r_{ier_cl} \in R, \emptyset, \emptyset \rangle,$$

який містить тільки ті його терміни, для яких знайдено відповідні фрагменти у контенті цього IP. Таким чином, склад тезаурусу IP залежить як від тезаурусу задачі, для якої він будується, так і від методу співставлення контенту IP із термінами цього тезаурусу.

Алгоритм побудови тезаурусу IP з використанням лексичної онтології складається з наступних кроків:

$\forall x_j \in X_{Th}, j = \overline{1, q}$ у лексичній онтології

$$L_{PrO} = \left\langle \begin{array}{l} X_{lex} = X_{PrO_i} \cup T_{PrO_i} \\ R_{lex} = \{r_{lex}\}, \emptyset \end{array} \right\rangle$$

шукати відповідні фрагменти ПМ

$$s_{j_p} \in T_{PrO}, p = \overline{1, q_j}, r_{lex}(s_{j_p}) = x_j,$$

якщо в контенті IP знайдено хоча б один з $s_{j_p} p = \overline{1, q_j}$, тоді додати до множини X_{IR} :

$$\forall x_j \in X_{IR} \exists s_{j_p} : r_{lex}(s_{j_p}) = x_j.$$

Для необхідності аналізу великої кількості IP для виконання кожного пошукового запиту виникає необхідність використовувати такий алгоритм побудови їх тезаурусу, обчислювальна складність якого лінійно залежить від обсягу IP та від обсягу опису задачі, для якої він будується.

Цей алгоритм застосовується тільки до тих природномовних IP, що не супроводжуються метаописами. За наявності метаописів (у форматі RDF [12] чи OWL [13]) для довільних IP (природномовних, мультимедійних, структурованих тощо) аналогічний алгоритм застосовується до цих метаданих: аналізуються елементи метаопису.

Слід зазначити, що запропонований алгоритм виконує співставлення не для всіх елементів лексичної онтології, а лише для тих, що відповідають тезаурусу задачі, що значно зменшує час його виконання через порівняно невелику кількість співставлень. Безпосередньо онтологія ПрО та опис задачі користувача в ньому не використовуються, але відповідні знання з них містяться в тезаурусі задачі, що будується за ними.

Алгоритм побудови зваженого тезаурусу IP

Зважений тезаурус IP

$$Tw_{IR} = \{ \langle t_j, w_j \rangle \}, t_j \in T, j = \overline{1, s}$$

будується за множиною X_{IR} тезаурусу IP Th (простим або складеним) наступним чином: кожному елементу з T ставиться у відповідність вага w_{IR_j} – позитивна кількісна характеристика важливості цього терміну для IP, що аналізується.

Ця оцінка обчислюється з урахуванням кількості успішних співставлень контенту IP з тими елементами лексичної онтології, що відповідають цьому терміну. Знаходження відповідностей до терміну в заголовку або метаописі IP може мати більше значення і тому оцінюється більш високо.

Побудова онтології задачі

Алгоритм побудови онтології задачі наведено в [7]. Для побудови онтології задачі доцільно застосовувати семантично розмічені IP, що використовують поширені стандарти для такої розмітки. На сьогодні найбільш відомим та вживаним засобом для цього є семантичні Wiki, наприклад, такі IP, що базуються на Semantic MediaWiki [8]. Використання семантичних Wiki-технологій для створення розподілених інформаційних ресурсів не тільки дозволяє досить легко додавати структурування до неструктурованих даних (НСД), але й є джерелом фонових знань для аналізу довільних природномовних текстів відповідної предметної області. Створення е-ВУЕ як семантизованого Wiki-ресурсу дозволяє вдосконалити процес генерації таких знань. Використання онтологічного аналізу – основа для переходу від неструктурованого контенту [9] до розподіленої бази знань, придатної для повторного використання.

Найпростіше використовувати неспеціалізовані енциклопедії та довідники (такі, як електронна версія Великої української енциклопедії [10]), але, якщо користувач має відомості до більш спеціалізованих ресурсів, то їх застосування може збільшити ефективність роботи.

Етапи побудови онтології за Wiki-ресурсом

Якщо користувач явно визначив множину понять ПрО, що його цікавлять, за допомогою множини Wiki-сторінок, тоді алгоритм побудови онтології ПрО має наступне.

Етап 1. Обрати множину Wiki-сторінок X, що пертинентні ПрО.

Етап 2. Здобути з цих сторінок всі категорії та відібрати ті, що пертинентні

ПрО (відкинути службові категорії, зайві для задачі категорії тощо). За множиною цих категорій побудувати множину класів ПрО K.

Етап 3. Проаналізувати множину K та за її структурою додати в онтологію ієрархічні відношення між класами.

Етап 4. Порівняти множини K та X і додати в онтологію ПрО екземпляри класів, що відповідають Wiki-сторінкам з X.

Етап 5. Проаналізувати семантичні властивості сторінок з X, обрати з них ті, що стосуються інших сторінок з X та додати відповідні об'єктні властивості до онтології ПрО.

Етап 6. Проаналізувати посилання (семантичні та звичайні) між сторінками з X та додати до онтології ПрО відповідні відношення між екземплярами класів.

Етап 7. Проаналізувати семантичні властивості, що пов'язують сторінки з X з даними. Додати ці властивості до властивостей даних онтології ПрО, а їх значення – до значень цих властивостей відповідних екземплярів онтології.

Рекурсивне розширення онтології ПрО на основі семантизованого Wiki-ресурсу

У тому випадку, коли користувач задає не всю множину термінів ПрО, що його цікавить, а тільки їх початковий набір, алгоритм побудови онтології розширюється наступними етапами:

Етап 2а. Здобути з множини K всі їх підкатегорії та відібрати ті, що пертинентні ПрО (відкинути службові категорії, зайві для задачі категорії тощо). За множиною цих категорій розширити множину класів ПрО.

Етап 2б. Проаналізувати інші екземпляри категорій з K та запропонувати користувачеві додати їх до X.

Повторювати етапи 2а та 2б доти, поки користувач не буде задоволений термінологічним складом ПрО.

Етап 5а. Проаналізувати семантичні властивості, що пов'язують сторінки з X з іншими Wiki-сторінками. За необхідності додати ці сторінки до X, а самі властивості додати до множини об'єктних властивостей онтології ПрО.

Повторювати етап 5а доти, поки користувач не буде задоволений термінологічним складом ПрО.

Алгоритм фільтрації IP на основі тезаурусів

Як було вказано вище, через велику кількість IP, доступ до яких забезпечує Web, основна проблема у пошуку інформації пов'язана не із знаходженням усієї множини IP I, пертинентних (більше або менше) потребам користувача, а у відборі з цієї множини I тих IP, що найбільш відповідають цій потребі. В даній роботі розглядається та підзадача пошуку в Web, що стосується фільтрації результатів пошуку за набором ключових слів, отриманих від довільної зовнішньої ППС.

Алгоритм фільтрації результатів запиту користувача до зовнішнього ППС:

– користувач обирає ППС, які забезпечують доступ до IP (у Web, корпоративній мережі, сховищі даних);

– користувач формулює запит, ідентифікуючи свою інформаційну потребу: за допомогою набору ключових слів, умов запиту, документів-зразків тощо – відповідно до можливостей, що надає обрана ППС;

– користувач обирає онтологію ПрО та за нею створює (формує або обирає зі вже існуючих) зважений тезаурус задачі

$$Tw_{user} = \{ \langle x_k \in T_{PrO}, w_k \rangle, k = \overline{1, s} \};$$

– запит передається до зовнішньої ППС, від якої отримують відповідні до запиту результати його виконання – посилань на IP та їхні короткі описи

$$I = \{ \langle ref_j, d_j \rangle, j = \overline{0, m} \},$$

де Ref_j – http-адреса відповідного IP, знайденого ППС, а d_j – коротка інформація про цей IP, що зовнішня ППС надає користувачеві у відповідь на запит;

– якщо множина I не порожня, тобто ППС знайшла у відповідь на запит хоча б один IP ($m \geq 1$), то потрібно встановити порядок, в якому пропонувати користувачеві відомості про знайдені IP.

Тоді для всіх IP з цієї множини I формуються їх зважені тезауруси $Tw_{IR_j}, j = \overline{1, m}$ та відповідні їм словники термінів X_{IR_j} .

Елементи цієї множини з k елементів $x_{jk} \in X_{IR_j}$ – це терміни ПрО, яким відповідають певні фрагменти з d_j – опису j-го IP з множини I, запропонованої ППС. У зваженому тезаурусі IP фіксується також w_{jk} – вага кожного терміну з X_{IR_j} , що кількісно характеризує його важливість у цьому IP (відповідно до місця, де знайдено відповідний фрагмент, та залежно від кількості таких успішних співставлень – відповідно до способу оцінювання, обраного користувачем);

За наявності зважених тезаурусів задачі та IP визначення семантичної близькості між цими об'єктами вирішується за допомогою обчислення коефіцієнту їх близькості:

$$K(Tw_{IR_j}, Tw_{user}) = \sum_{i=1}^n w_{user_i} * w_{IR_{j_i}} * f(x_i).$$

Функція $f(x)$ виконує співставлення термінів тезаурусів задачі та IP:

$$f(x) = \begin{cases} 1: x \in Tw_{IR_j} \\ 0: x \notin Tw_{IR_j} \end{cases}.$$

Коефіцієнт близькості враховує кількість термінів тезаурусу задачі, що знайдено у тезаурусі IP, так і в тезаурусі ПрО, важливість цих термінів для задачі користувача та важливість цих термінів для контенту IP. На практиці можуть застосовуватися різні варіанти цього критерію, наприклад, нормовані значення w_{IR_j} , що дозволяють обробляти IP різного обсягу, але для аналізу невеликих фрагментів ПМ-тексту приблизно однакового розміру, які надають зовнішні ППС, достатньо використовувати таку оцінку.

Отримавши оцінки для всіх знайдених IP, можна виконати впорядкування їх списку за цими оцінками, використовуючи довільний алгоритм сортування масивів. Знайдені IP впорядковуються залежно від значень K_j .

Часова складність алгоритмів сортування різниться від $O(n)$ до $O(n^2)$. Доцільно застосовувати такі алгоритми сортування з часовою складністю $O(n \log n)$, як сортування злиттям, швидке сортування, пірамідальне сортування. Але слід враховувати, що стабільні алгоритми сортування, що працюють за час $O(n \log n)$, потребують $O(n)$ додаткової пам'яті. Якщо використовуються алгоритми сортування з часовою складністю $O(n)$, такі як сортування комірками, сортування підрахунком, сортування за розрядами, то вони потребують використання додаткової інформації про елементи, приміром, діапазон значень ключа.

В даному випадку елементи множини, що впорядковуються, містять не тільки ключ, за яким здійснюється сортування, але й інформацію про місцезнаходження відповідного файлу, тобто впорядкування відбувалось не у самому масиві елементів, а в масиві ключів, що є посиленнями на інші дані. Такий підхід не спрямований на аналіз повного контенту IP, але забезпечує той самий рівень аналізу, який використовують користувачі, вручну проглядаючи такі описи: короткий опис може некоректно відображати вміст самого IP, але це залежить не від засобів аналізу, а від самого IP.

Користувачеві надають насамперед ті IP, що мають найбільші значення K_j – коефіцієнтів близькості до ПрО. Можна обмежити множину IP, що надаються користувачеві, за двома параметрами, – кількістю IP (наприклад, перші 20 найближчих IP) та значенням K_j (наприклад, надавати IP) з $K_j \geq 5$.

Доцільно використовувати одночасно як позитивні (відомі, бажані, релевантні терміни), так і негативні (незнайомі, незрозумілі, нерелевантні проблеми терміни) тезауруси. Якщо в IP трапляються терміни з негативною вагою, то зменшує семантичну близькість IP до задачі.

Запропонований підхід доцільно використовувати, якщо:

- користувач досить глибоко обізнаний у ПрО та близьких до неї областях;

- користувач виконує велику кількість запитів з однієї ПрО, пов'язаних з різними задачами;

- користувач виконує велику кількість запитів, пов'язаних з однією ПрО, що відповідають різним її аспектам або етапам і тому потребують різної інформації;

- користувач досить довго займається пошуком інформації, і тому час, який він витрачує на вибір онтології ПрО та побудову тезаурусу задачі, значно менше за той час, який він витрачував на ручний прогляд результатів кожного запиту у цій сфері.

Таким умовам відповідають наукова діяльність (наприклад, моніторинг публікацій в обраній сфері, пошук аналогів), навчальний процес, аналітичні дослідження тощо.

Висновки

Запропонований у роботі підхід до застосування онтологічної моделі взаємодії між користувачами та IP у процесі семантичного пошуку забезпечує знаходження IO із складною структурою, формалізований опис яких міститься у зовнішніх онтологіях. Це дозволяє використовувати фонові знання, що подаються у вигляді тезаурусів, для персоніфікованої фільтрації потрібного користувачам контенту, що особливо актуально із зростанням кількості, обсягу та структурної складності IP, що оброблюються. Перехід від онтологій до їх окремого випадку – тезаурусів – зменшує обчислювальну складність співставлення IO. Використання семантичних Wiki-ресурсів як джерела онтологічних знань дозволяє значно точніше описувати ПрО, що цікавлять конкретних користувачів, і внаслідок цього отримувати більш точні результати пошуку за менший час.

Література

1. Baeza-Yates R., A. Raghavan R. Next generation Web search. S. Ceri and M. Brambilla, editors, Search Computing, Springer. 2010. P. 11–23.

2. Рогушина Ю.В. Семантический поиск у Web на основе онтологий: разработка моделей, средств и методов. Мелітополь: МДПУ ім. Богдана Хмельницького. 2015. 291 с.
3. Ushold M., Gruninger M. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*. 1996. Vol. 11. N 2.
4. Antoniou G., Van Harmelen F. Web ontology language: Owl. *Handbook on ontologies*. Springer Berlin Heidelberg. 2004. P. 67–92.
5. Рогушина Ю.В. Теоретичні засади застосування онтологій для семантизації ресурсів Web. *Проблеми програмування*. 2018. № 2-3. С. 197–203.
6. Mitchell, T.M. Machine learning. Burr Ridge, IL: McGraw Hill, 45(37). 1997. P. 870–877.
7. Rogushina J.V. Models and Methods of Ontology Use for the Web Semantic search. Proc. of the 11th International Conference of Programming UkrPROG 2018, P. 197–203. <http://ceur-ws.org/Vol-2139/197-203.pdf>.
8. Semantic MediaWiki. https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki.
9. Grimes S. Unstructured Data and the 80 Percent Rule, 2008, Clarabridge, Bridgepoints. <http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/>.
10. Рогушина Ю.В. Використання семантичних властивостей вікі-ресурсів для розширення функціональних можливостей «Великої української енциклопедії». Енциклопедичні видання в сучасному інформаційному просторі: колективна монографія / За ред. Киридон А.М. К.: Державна наукова установа «Енциклопедичне видавництво», 2017. С. 104–115.
11. Гладун А.Я., Рогушина Ю.В. Репозитории онтологии как средство повторного использования знаний для распознавания информационных объектов. *Онтология проектирования*. 2013. № 1 (7).
12. Resource Description Framework (RDF) Model and Syntax Specification. W3C Proposed Recommendation, 1999. <http://www.w3.org/TR/PR-rdf-syntax>
13. OWL 2 Web Ontology Language Document Overview. W3C. 2009. <http://www.w3.org/TR/owl2-overview/>.
2. Rogushina J.V. (2015) Semantic retrieval in the Web on base of ontologies: design of methods, means and methods. Melitopol, MDPU. [in Ukrainian].
3. Ushold M., Gruninger M. (1996) Ontologies: Principles, Methods and Applications, // Knowledge Engineering Review, V.11, N 2.
4. Antoniou G., Van Harmelen F. (2004) Web ontology language: Owl. Handbook on ontologies. Springer Berlin Heidelberg, P. 67-92.
5. Rogushina J.V. Theoretical principles of use of ontologies for semantization of the Web resources. Problems in programming. 2018. N 2-3. P. 197–203.
6. Mitchell, T. M. (1997) Machine learning. Burr Ridge, IL: McGraw Hill, 45(37). P. 870–877.
7. Rogushina J.V. Models and Methods of Ontology Use for the Web Semantic search. Proc. of the 11th International Conference of Programming UkrPROG 2018, P.197-203. – <http://ceur-ws.org/Vol-2139/197-203.pdf>.
8. Semantic MediaWiki. https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki.
9. Grimes S. (2008) Unstructured Data and the 80 Percent Rule, , Clarabridge, Bridgepoints. <http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/>.
10. Rogushina J.V. (2017) Use of semantic properties of the Wiki resources for expansion of functional possibilities of “Great Ukrainian Encyclopedia” // Encyclopaedias in the modern information space: collective monograph / Ed. Kyrydon A.M., Kyiv. P. 104–115. [in Ukrainian]
11. Gladun A., Rogushina J. (2013) Ontology repositories as a means of knowledge reuse for recognizing of information objects // Ontology of Design, № 1 (7). [in Russian]
12. Resource Description Framework (RDF) Model and Syntax Specification. W3C Proposed Recommendation, 1999. <http://www.w3.org/TR/PR-rdf-syntax>.
13. OWL 2 Web Ontology Language Document Overview. W3C. 2009. <http://www.w3.org/TR/owl2-overview/>.

References

1. Baeza-Yates R., A. Raghavan R. (2010) Next generation Web search // S. Ceri and M. Brambilla, editors, Search Computing, Springer, P.11-23.

Одержано 24.10.2019

Про автора:

Рогущина Юлія Віталіївна,
кандидат фізико-математичних наук,
старший науковий співробітник.
Кількість наукових публікацій в
українських виданнях – 150.
Кількість наукових публікацій в
зарубіжних виданнях – 31.
<http://orcid.org/0000-0001-7958-2557>.

Місце роботи автора:

Інститут програмних систем
НАН України,
03181, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: 066 550 1999.
E-mail: ladamandraka2010@gmail.com

КЛАСИФІКАЦІЯ МЕТАДАНИХ ВЕЛИКИХ ДАНИХ

Насьогодні накопичені величезні обсяги даних різної структури (або в загалі не структуровані) та походження, їх обсяги зростають експоненційно. Проблема полягає у тому, що існуюче програмне та апаратне забезпечення, яке не може обробити таку кількість різноманітних типів даних, що створюються з великою швидкістю. Великі дані стали надто складними та динамічними, щоб їх можна було обробляти, зберігати, аналізувати та управляти ними за допомогою традиційних засобів. Це обумовило виникнення нових платформ та підходів для роботи з даними, а разом з цим чітке розуміння того, що для вирішення задач великих даних ці необроблені дані повинні бути доповнені метаданими. Насьогодні метадані є засобом класифікації, впорядкування та характеристики даних, та їх вмісту. Їх найважливішою особливістю є впорядкована структура. Завдяки структурованому вигляду, метадані доступні для читання не лише для людини, але й для комп'ютера. Таким чином, вони можуть оброблятися автоматизовано та використовуватися для різних цілей: для індексації, пошуку, об'єднання, автоматичної обробки, класифікації великих даних тощо. Побудова ефективних систем керування метаданими, перш за все, вимагає їх узгодженої загальної класифікації з урахуванням типів джерел (способів отримання) даних, що формують контент задач, що мають вирішуватися на різних етапах життєвого циклу, існуючих форматів представлення відповідних даних, принципів розумної ефективності, так як часто розміри метаданих значно перевищують обсяг самих даних (навіть великих), які вони описують. Тому, мета даної роботи полягає в аналізі існуючих джерел великих даних, способів створення та обробки відповідних метаданих, а також програмних засобів, що дозволяють опрацювати метадані певним чином, та побудові класифікації метаданих на основі проведеного аналізу.

Ключові слова: джерела великих даних, керування метаданими, Nadoop, класифікація метаданих, аналіз метаданих, сервіси обробки метаданих, створення метаданих, перегляд метаданих, редагування метаданих, метадані зображень, метадані аудіо-файлів, метадані відео-файлів, метадані сховищ, метадані в соціальних мережах.

Вступ

При впровадженні кожного нового проекту великих даних має бути можливість їх ідентифікувати. Важлива можливість для розробки та розвитку сервісів обробки великих даних – створення комплексної програми керування метаданими. Метадані можуть значно спростити та вдосконалити процеси збору, інтеграції та аналізу великих даних. За відсутністю метаданих підприємства можуть втратити глибоке розуміння того, що саме можуть дати великі дані. Метадані можуть керувати всім життєвим циклом даних, процесами, процедурами, а також клієнтами або користувачами, які впливають на певну бізнес-інформацію. Вони є основою для збору величезних обсягів даних з різних джерел та інформаційних сховищ, перш ніж вони стануть некерованими.

Слід також зазначити, що зростання ролі метаданих значною мірою обумовлене розвитком Веб. Це стосується не лише виникнення самих великих даних. Веб зві-

льняє від обмежень фізичного світу, а саме контент, що формують великі дані, вже не повинен знаходитись в один момент часу в одному місці, як стілець чи шафа. Веб контент може існувати в багатьох місцях одночасно. Метадані дозволяють помітити контент термінами з таксономії, що описує його мету. Для цього можуть бути використані півдюжини різних словників. Деякі можуть описувати семантику контенту, інші – функції/задачі контенту та аудиторію, для якої він призначений, формат контенту, його структуру, API платформу чи компоненти, що використовуються та інше. Потім ці метадані можуть використовуватися багатьма організаціями відповідно до різних цілей. Правила відповідної системи керування сайтом (CMS) визначають, де буде з'являтися контент.

Метадані можуть бути застосовані як для опису контенту, так й бути представлені як фасети у результатах пошуку, щоб визначити фільтрацію результату.

Окрім цього, можна створювати складні фільтри, які комбінуватимуть метадані різними способами, наприклад, якщо потрібно знайти весь контент, що відповідає (інструменту) АСМЕ АРІ та використовується для налагодження віджетів (функція), а також призначений для розробників (персонал). Інформацію можна отримати миттєво за допомогою фасетних фільтрів або завчасно побудованих запитів.

Звільнення контенту від статичної, єдиної позиції у змісті, ймовірно, є найбільш актуальною та явною перевагою, яка з'явилася в результаті переміщення контенту до Інтернету. Але доки контент не збагачений метаданими та семантичними мітками, його складно знаходити, витягувати, просувати та обробляти різними способами. Це вимагає спеціальних інструментів для створення та анотування контенту метаданими. Слід зазначити, що контент має бути таким, що легко переноситься. Звісно, що використання сторонніх схем метаданих вимагає додаткових зусиль щодо оформлення контенту, але це може забезпечити кращу взаємодію з користувачами на різних платформах у пристроях у сьогоденні та майбутньому. Тому, створення добре структурованого контенту та забезпечення його гнучкості, та можливості легкого перенесення є ключем для вирішення багатьох задач.

Впорядкована структура саме є найважливішою особливістю метаданих. Інформація категоризована та має визначену форму/формат. Наприклад, категорію *час створення* можна визначити лише за допомогою формату запису дати та часу. Завдяки структурованому вигляду, метадані доступні для читання не лише для людини, а й для комп'ютера. Таким чином, вони можуть оброблятися автоматизовано та використовуватися для різних цілей: для індексації, пошуку, об'єднання, автоматичної обробки тощо.

Побудова ефективних систем управління метаданими, перш за все, вимагає їх узгодженої загальної класифікації з урахуванням типів джерел (способів отримання) даних, що формують контент, задач, що мають вирішуватися на різних етапах життєвого циклу, існуючих форматів

представлення відповідних даних, принципів розумної ефективності, так як часто розміри метаданих значно перевищують обсяг самих даних (навіть великих), які вони описують.

Джерела великих даних

Класифікація метаданих неможлива без усвідомлення типів джерел надходження великих даних, різновидів самих великих даних, потреб користувачів у їх використанні та способів обробки.

На найвищому рівні відповідно до способів отримання даних можна виділити внутрішні та зовнішні джерела (рис. 1).

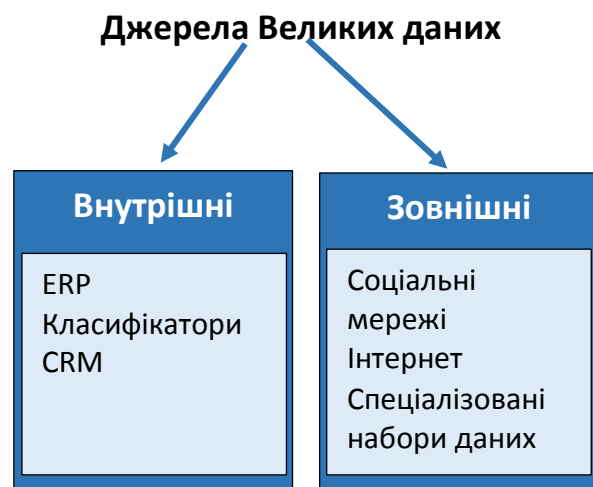


Рис. 1. Внутрішні та зовнішні джерела даних

Найбільш традиційним типом джерел даних є бази даних (БД). Існує багато різних БД, які мають власну архітектуру та властивості. Але використання транзакційних БД на сьогодні вже не є оптимальним рішенням для вирішення задач у бізнес-аналізі. На це існує багато причин, зокрема, вимоги щодо попередньої оптимізації даних для звітності та аналізу, вимоги до структурованості контенту, певні обмеження обсягів контенту, низька швидкість виконання запитів до даних тощо. У деяких випадках компанії використовують ETL засоби для збору даних з транзакційних БД, перетворення їх певним чином, щоб вони були оптимізовані для бізнес-аналізу, та завантаження їх у сховище та інші вітрини даних. Але, на сьогодні, в умовах відкритого світу та інформаційного

буму, треба мати можливість обробляти дані з не структурованих або слабко структурованих інформаційних джерел. На рис. 2 показано порівняльну характеристику можливих джерел інформації різного рівня структурованості. Аналіз джерел проводиться за основними властивостями великих даних, а саме 3V характеристиками: швидкість (Velocity), різноманітність (Variety) та обсяг (Volume).

Як основні джерела отримання інформації можна виділити.

1. *Архіви відсканованих документів*, заяв, страхових форм, медичних записів, кореспонденції, архіви паперових документів, друковані файли потоку, які містять вихідні системи записів між організаціями та їх користувачами.

2. *Документи*: файли різних форматів xls, word, html, html 5, pdf, csv, ppt, txt, xml, json тощо.

3. *Сховища даних* (SQL або NoSQL), файлові системи тощо.

Сьогодні підприємства вважають за краще використовувати як традиційні, так й сучасні БД (разом), щоб отримати відповідні великі дані. Ця інтеграція відкриває шлях для гібридної моделі даних і вимагає низьких інвестиційних витрат та витрат на ІТ-інфраструктуру. Крім того, такі гібридні БД розгортаються також й одночасно для кількох цілей бізнес-аналітики, та потім можуть забезпечувати вилучення інформації, яка використовується для отримання прибутку. Процес вилучення та аналізу даних з джерел великих даних є складним процесом. Ці проблеми можна вирішити, якщо організації охоплюють всі необхідні міркування великих даних, беруть до уваги відповідні джерела даних і розгортають їх у спосіб, який добре налаштований на цілі організації.

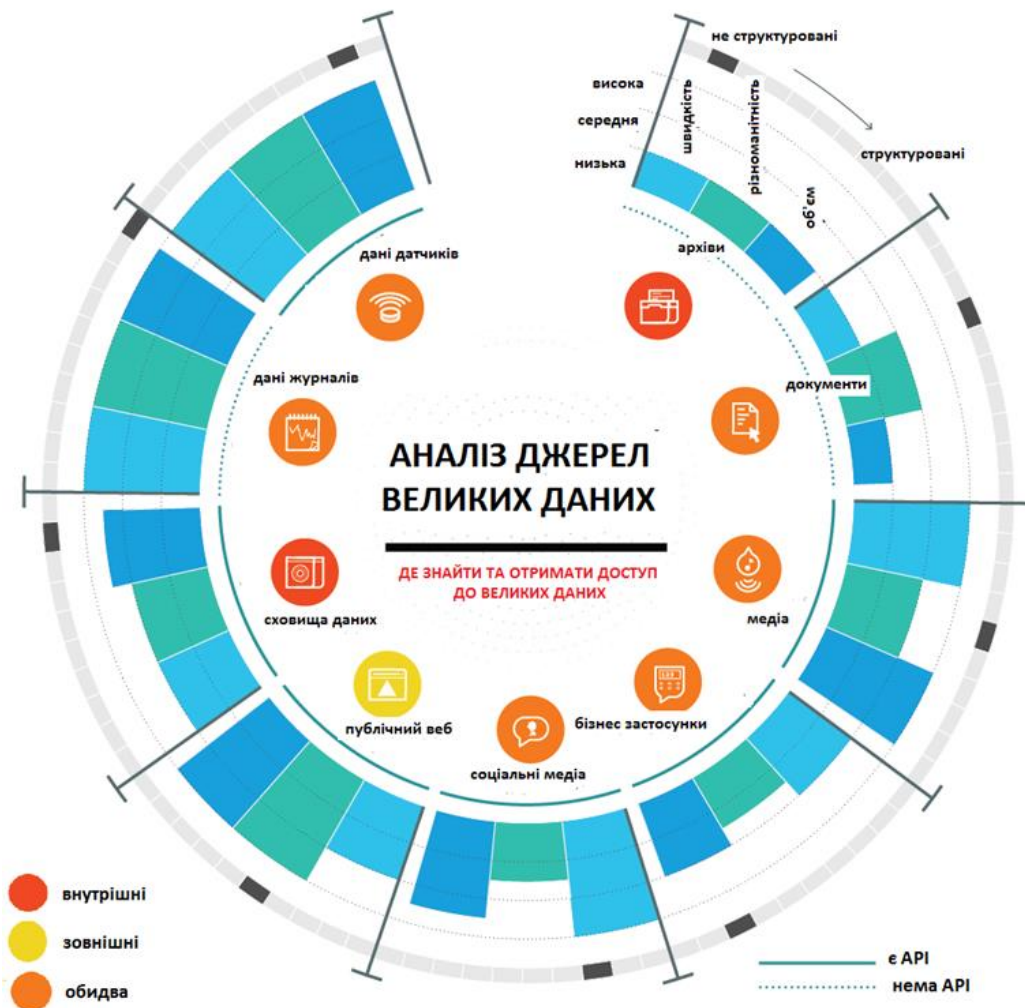


Рис. 2. Аналіз джерел великих даних

4. *Бізнес-застосунки*: системи керування проектами, системи автоматизації маркетингу, продуктивності, CRM системи, керування ERP контентом, HR, керування талантами, системи закупівель, керування витратами, портали, Інтернет системи тощо.

5. *Публічний веб*. Цей веб містить дуже поширені та легко доступні великі дані. Дані в Інтернеті є загально доступними як для фізичних осіб, так і для компаній. Різноманітні веб сервіси (сервіси охорони здоров'я, загальнодоступні фінансові сервіси, Вікіпедія, сервіс перепису населення, економічні сервіси, встановлення відповідності тощо) надають всім безкоштовну та швидко інформацію. Величезні розміри мережі забезпечують зручність її використання у різних аспектах. Веб є особливо корисним для стартапів та малих і середніх підприємств, оскільки звільняє їх від необхідності розробки власної інфраструктури великих даних й сховища даних, перш ніж вони зможуть використовувати ці великі дані.

6. *Засоби масової інформації* (ЗМІ) є найпопулярнішим джерелом великих даних, оскільки надають цінну інформацію про переваги споживачів і змінюють тенденції. Оскільки це інформація, що транслюється самостійно і перетинає всі фізичні та демографічні бар'єри, медіа джерела є для підприємств найшвидшим способом отримати глибокий огляд своєї цільової аудиторії, побудувати закономірності, зробити висновки та підвищити рівень прийняття рішень. ЗМІ включають соціальні медіа та інтерактивні платформи, такі як: Google, Facebook, Twitter, YouTube, Instagram, а також загальні засоби масової інформації (медіа), такі як: зображення, відео, аудіо, записи прямих ефірів та підкасти (цифрові аудіо файли в Інтернеті, готові до завантаження), які забезпечують кількісні та якісні висновки щодо кожного аспекту взаємодії з користувачем.

7. *Дані журналів*: дані серверів, журнали подій, журнали бізнес-процесів, журнали прикладних систем, дані поточкових каналів, детальні записи викликів, мобільні локації, використання мобільних застосунків тощо.

8. *Автоматично генерований контент* (ІОТ) – це значиме джерело великих даних. Це дані, які зазвичай генеруються датчиками, що пов'язані з електронними пристроями. Це не лише інформація, що генерується комп'ютерами та смартфонами, але й дані з медичних пристроїв, датчиків автомобілів, розумних електричних лічильників, дорожніх камер, супутників, пристроїв запису дорожнього руху, процесорів у транспортних засобах, відеоігр тощо.

Метадані різних типів даних

Контент, що описується метаданими, може бути представлений у різних форматах. І, в загальному випадку, кожний тип файлу має власний стандарт для визначення метаданих. Але, насправді існує не так вже й багато схем, протоколів та форматів представлення метаданих. Розглянемо далі можливі схеми метаданих відповідно до типів джерел великих даних, які вони описують.

Метадані зображень. Зображення, як й будь-який інший контент, також мають різні формати та *схеми метаданих* [1]. У випадку графічних файлів, таких як фотографії з цифрової камери або смартфона (формати JPEG, TIFF та RAW), ми переважно маємо справу з метаданими формату Exif, які є досить розвиненими порівняно з метаданими інших типів файлів.

EXIF (Exchangeable Image File Format) – стандарт зберігання метаданих у зображенні, що використовується цифровими камерами для зберігання інформації про витримку, діафрагму та інші параметри зйомки. Метадані у форматі EXIF можуть зберігатися у файлах форматів JPEG, TIFF та RIFF WAV.

EXIF формат був розроблений з виникненням цифрових камер та, перш за все, для опису зроблених ними зображень, і, тому, ці дані є у кожній фотографії, не залежно від того, з якого пристрою вона була зроблена. EXIF – це не лише параметри фотоапарата/смартфона, з якого було зроблено фотографію, але й багато іншого: дата створення, геолокація, інформація про власника кадра тощо. За стандартом з

користувачьких описових метаданих в EXIF може зберігатися лише опис (тег Description) та коментар (тег User Comment), але Windows Explorer використовує також декілька додаткових тегів (XPTitle, XPSubject, XPAuthor, XPComment, XPKeywords). Windows Explorer ігнорує тег XPTitle за наявності стандартного тега Description. Таким чином, це в основному відомості про характеристики, структуру, розміщення зображення, умови та способи його отримання, автора, дату та час його змінення, та дуже мало справжньої семантики. З повним переліком елементів метаданих можна ознайомитись у специфікації стандарту [2].

EXIF є частиною більш широкого стандарту DCF (Design rule for Camera File system). DCF [3] дозволяє визначити метадані не лише для файлу зображення, але й пов'язаного з ним файлу, наприклад, аудіофайлу. DCF також містить досить детальну, але технічну інформацію.

Якщо виникає необхідність визначення більш детального опису змісту фотографії, Exif може бути розширений набором *метаданих у стандарті IPTC*, який, окрім полів, що пов'язані з темою фотографії, має розділ для контактних даних фотографа. Це стандартний додаток графічних файлів, що доступні через банки фотографій.

IPTC (International Press Telecommunications Council) – це скоріше назва організації, що розробила даний стандарт, Міжнародна Рада з питань преси та телекомунікації. Назва самого стандарту – ПМ (Information Interchange Model). На відміну від EXIF, який спрямований на технічну інформацію, ПМ дозволяє зберігати різну детальну інформацію. У метаданих даного стандарту можуть зберігатися такі описові поля, як ObjectName (заголовок), Keywords (ключові слова), Caption (опис, існує декілька варіацій тега).

У початкових версіях стандарту метадані зберігалися таким чином, що програмне забезпечення (ПЗ), яке не знало про існування IPTC, не могло працювати з файлами зображень з такими метаданими. Але, згодом стандарт був розширений Adobe, та метадані були перенесені до бло-

ку APP13 JPEG-файлу, що дозволило такому ПЗ успішно читати JPEG-файл, ігноруючи невідомі метадані. IPTC інформацію підтримують фотобанки, пошукові сервіси тощо.

XMP (eXtensible Metadata Platform) [4] – стандарт, розроблений Adobe, який, починаючи з 2012 року, є також стандартом ISO. Метадані зберігаються у моделі RDF, яка представлена в XML форматі. Це дозволяє включити будь-яку необхідну інформацію до файлу зображення. Це стандарт з відкритим кодом. XMP метадані можуть бути додані до графічних файлів багатьох різних типів.

Властивості метаданих групуються в схеми. Кожна схема ідентифікується унікальним URI простору імен та містить довільну кількість властивостей. Хоча URI просторів імен виглядає досить схожим на веб-адреси (в дійсності, вони часто виглядають однаково), важливо зазначити, що вони не ідентифікують конкретну веб сторінку. Це просто унікальні ідентифікатори для деякої сутності, що використовується в XMP. Остання специфікація включає більше десятка передвизначених схем з сотнями властивостей для документа загалом та характеристик зображення. Більшість таких схем називається Дублінським ядром (DC) [5] та включає загальні властивості такі як: Назва, Тема, Автор та Опис. Окрім передвизначених схем, можуть бути визначені спеціальні (власні) схеми, щоб задовільнити специфічні вимоги до метаданих організації. Власні формати метаданих можна імпортувати та розповсюджувати разом з іншими XML файлами.

Все це дозволяє розробникам досить просто адоптувати специфікацію стандарту до стороннього ПЗ. Дана технологія забезпечує також обмін метаданими між прикладними системами. Метадані, що визначені в інших форматах, таких як Exif, IPTC (ПМ), GPS, та TIFF, можна синхронізувати з XMP, що полегшує їх використання та керування ними. Даний стандарт, реалізований у всіх Adobe продуктах підтримується десятками незалежних розробників ПЗ та груп користувачів.

Насьогодні специфікація XMP

складається з трьох частин:

– «*Модель даних, серіалізація та базові властивості*» [6] охоплює модель представлення основних метаданих, що є базисом формату XMP стандарту. Модель даних прописує, як можуть бути організовані метадані XMP, незалежно від формату файлу чи специфіки використання. Модель серіалізації визначає, як модель даних представляється в XML, зокрема, в RDF.

– Друга частина [7] забезпечує детальний перелік властивостей для схем метаданих XMP стандарту; вони включають схеми загального призначення, такі як: Дублінське ядро, та схеми спеціального призначення для застосунків Adobe, таких як Photoshop. Також забезпечує інформацію з розширення існуючих схем та створення нових.

– «*Зберігання у файлах*» забезпечує інформацію про те, як серіалізовані XMP метадані пакуються у XMP пакети та вбудовуються у файли різних форматів. Включає інформацію щодо відношення XMP з іншими форматами метаданих, та узгоджує значення, що представлені в інших форматах.

Існує також стандарт ISO 16684-2:2014, *Graphic technology – Extensible metadata platform (XMP) – Part 2: Description of XMP schemas using RELAX NG* (опис схеми XMP за допомогою RELAX NG), який специфікує використання RELAX NG для опису серіалізованих XMP метаданих. Стандарт визначає, як відповідні схеми можуть використовувати функції RELAX NG.

Метадані аудіофайлів. Завдяки приєднанню метаданих до звукових файлів виникає можливість доповнити звук (чи музику) будь-якою необхідною інформацією. Інформація може бути дуже різноманітною, виходячи з вимог та бажань автора чи власника запису. Метадані – це не лише назва трека чи рік випуску музичного альбому, але й ім'я композитора, автора аранжировки, тексту слів до пісні, адреси сайтів, електронні адреси, все, що пов'язане з художнім оформленням пісні або альбому, якщо це музичний файл та багато іншого. У сучасних мультимедій-

них пристроях пошук композицій здійснюється не за назвою файлів чи папок, а за метаданими, які там містяться, та лише за умови їх відсутності, за назвою файлів. Виробники мобільних пристроїв таких відомих марок, як наприклад, Nokia, Sony Ericsson, iPod, і та інші слідуєть цьому ж принципу.

Першим запропонував додавати до MP3-файла невеличкий блок з даними програміст Ерик Кемп (проект «Studio3»). Цей блок був названий ID3tag (tag – ярлик, мітка, ID3 – Identification Data for Studio3). В наслідок цього назва TAG надійно закріпилася за метаданими інших форматів, таких як: WMA, OGG, MP4 та інших.

Щоб не викликати не сумісності з плеєрами, тег розміщувався наприкінці файлу, в результаті чого, міг просто ігноруватися без будь-яких наслідків. Це дало можливість додавати потрібну текстову інформацію до будь-якого MP3 файлу.

Серед існуючих насьогодні форматів метаданих аудіофайлів можна виділити: ID3tag, Lyrics3 tag, APE tag, WM metadata (частина стандарту Windows Media), Vorbis comments, MP4/iTunes metadata, ATRAC metadata.

Перелік можливих властивостей, що визначаються метаданими, досить великий [8], тому, наведемо деякі з них на прикладі фреймів стандарту ID3v2 [9]:

- назва альбому, фільму або шоу, якому належить уривок;
- головний виконавець;
- група, оркестр, супровід;
- кількість ударів в секунду;
- коментар;
- композитор;
- уточнення до назви;
- інформація про авторські права;
- ім'я людини, що закодувала файл;
- жанр музики;
- мова тексту;
- назва твору;
- номер твору в альбомі;
- текст;
- синхронізований текст;
- рік;

- обкладинка;
- точки початку/кінцівки;
- мітки синхронізації з аудіо-потокком для тексту пісні.

Повний перелік фреймів стандарту ID3v2 та їх значень можна знайти на офіційному сайті [10]. Окрім фреймів, наданих в цій специфікації, користувачі можуть створювати власні фрейми з власною структурою. Інші відомі стандарти метаданих аудіо – iTunes, XSPF [11], також містять в основному інформацію про альбом, трек, авторів, тривалість запису і т. і. Це дуже корисна інформація, яка може бути використана для пошуку, але вона не розкриває семантичної сутності самого запису. Єдиний елемент, що може характеризувати зміст запису, є назва (Title).

Метадані відеоматеріалів. Як і зображення, відео також містять додаткову описову інформацію – метадані про технічні подробиці зйомки, характеристики камери, місце зйомки, назву та опис відео тощо. Теги Назва (Title) та Опис (Description) дозволяють задати семантичну інформацію, яка може бути використана для пошуку. Окрім цього, як правило, можна задати додаткові теги, або метатеги, де визначити слова або фрази, що описують відео.

Теги відрізняються від ключових слів тим, що вони не відображаються користувачам у результатах пошуку. Але, пошукові системи впливають на них при певній мірі індексації сторінки чи відео. Як правило, метадані асоціюють з текстовою інформацією на сторінці (назви, описи та ін.), але вони можуть використовуватися й для надання більш глибокої інформації. Наприклад, "прямі тимчасові метадані" або метадані на основі часу – це інформація, яка прив'язана до часової шкали в межах відео. Вдалим прикладом однієї з форм часових метаданих є закриті підписи. Приєднаний текст може також включати інформацію про те, коли та яка музика відтворюється, сцену, коли є сміх, хто говорить тощо.

Різні засоби відеозйомки можуть генерувати власний набір метаданих, залежно від його цілей, призначення тощо. Так,

IBM® Intelligent Video Analytics [12] аналізує будь-який об'єкт, який рухається у відеопотоці, та створює метадані для опису ідентифікованих дій та подій. Компонент Smart Surveillance Engine (SSE) для Intelligent Video Analytics генерує метадані при обробці відеофідів. Компонент Middleware for Large Scale Surveillance витягує й зберігає дані, та керує ними.

Кожний екземпляр метаданих, що згенеровані Intelligent Video Analytics:

- описує єдиний об'єкт відеоспостережень;
- включає у себе відмітку часу, що використовується для перегляду відео;
- містить ключовий кадр, який надає зведення події або сповіщення;
- містить ідентифікатор представлення, що використовується для посилання на камеру цього відеозапису.

Метадані, які отримані у Intelligent Video Analytics, залежать від аналітичного профіля, що обраний при створенні цього каналу. Операційний аналітичний механізм може генерувати різні формати метаданих. Ці формати описані визначеннями типу контенту (CTD), що сконфігуровані в межах аналітичного профілю.

До складу кожної події або сповіщення включається декілька повідомлень метаданих, які містять різні атрибути для виявлення сегменту змінення або руху у джерелі відеосигналу. За потребою до Intelligent Video Analytics можна додати потрібні типи контенту, включивши аналітичні механізми для відправлення метаданих датчиків та подій. Можливе також виконання перехресної кореляції метаданих від усіх аналітичних механізмів, дозволяючи користувачам пошук за модальностями. Такі розширені можливості індексації забезпечують унікальний та потужний диференціатор порівняно з іншими існуючими рішеннями для спостережень.

Ступенем деталізації метаданих можна керувати. Більша рухливість об'єктів у полі зору призводить до більшої кількості метаданих, що зберігаються в БД системи для обраного каналу. Існує також кореляція між об'ємом метаданих та продуктивністю.

Враховуючи особливості складу та використання відео файлів, окрім технічних та загальних описових властивостей, що визначаються у метаданих, та спеціальних описових властивостей, притаманних відповідній предметній області, цілям бізнесу, чи застосунку, можна виділити ще дві групи елементів метаданих (які, зокрема, генеруються й Intelligent Video Analytics): метадані подій та метадані оповіщень.

Метадані подій:

- описують об'єкти або дії, що спостерігаються у відеоматеріалі;
- містять атрибути, які описують об'єкти. Атрибути змінюються залежно від аналітичного профілю;
- зазвичай використовуються для криміналістичного пошуку або статистичного аналізу.

Метадані оповіщень:

- ініціюються, коли для об'єкта на відео виконані попередньо визначені умови;
- базуються на поведінці та атрибутах;
- використовуються для моніторингу в реальному часі та можуть бути використані також для статистичного аналізу.
- За замовченням метадані оповіщень включають наступні атрибути:
 - час, коли воно було ініційоване;
 - канал, по якому це відбулося;
 - передвизначені ім'я та тип оповіщення;
 - пріоритет оповіщення;
 - затримка перед відтворенням (час в секундах перед ініціацією оповіщення).

Технічні метадані можуть визначати інформацію про файл (назва та розмір, формат, тривалість запису, формат аудіо даних, роздільна здатність тощо) у CVS-документі чи таблиці [13]. Щодо стандартів метаданих, кожний формат відеофайлу має власний метод зберігання метаданих, але метадані можна також зберігати у зовнішньому файлі (чи БД) або використовувати XML/ XSD комбінацію.

MPEG7 [14] – стандарт опису мультимедійного вмісту, який набув статусу стандарту в ISO/IEC 15938 («*Інтерфейс опису мультимедійного вмісту*»). Для формалізації/зберігання метаданих він використовує XML та може бути приєднаний до тимчасового коду, наприклад, щоб синхронізувати текст з піснею. Але стандарт не набув широкого розповсюдження.

Досить вдала реалізація намірів щодо узагальнення структурованої інформації для опису великих даних була зроблена спільними зусиллями компаній Google, Microsoft, Yahoo та Yandex, які створили словниковий ресурс схем для структурованих даних Schema.org [15], який детально, розглядатимемо далі. Разом з іншими, даний словник визначає перелік характеристик для опису відеооб'єктів.

Окрім цього, набори метаданих для декількох передвизначених типів контенту, таких як: відеокліпи, аудіофайли, вебсторінки і т. і. були запропоновані розробниками Open Graph, що також, розглядатимемо далі.

Словники Schema.org – спільна робота спільноти з місією створення, підтримки та просування схем для структурованих даних в Інтернеті, на вебсторінках, у повідомленнях електронної пошти та за її межами. Словник Schema.org можна використовувати з багатьма різними кодуваннями, включаючи RDFs, Microdata та JSON-LD. Ці словники охоплюють сутності, відносини між сутностями та діями і можуть бути легко розширені за допомогою добре документованої моделі розширення. Понад 10 мільйонів сайтів використовують Schema.org для розмітки своїх вебсторінок і повідомлень електронної пошти. Багато додатків від Google, Microsoft, Pinterest, Yandex та інших вже використовують ці словники для енергійного, розширюваного досвіду.

Заснована компаніями Google, Microsoft, Yahoo і Yandex, словники Schema.org розробляються в рамках спільноти, використовуючи список розсилки public-schemaorg@w3.org і через GitHub. Спільний словник (колекція словників) спрощує веб-майстрам та розробникам вирішення

питань щодо схеми метаданих та максимізує ефективність їх роботи.

Так, для відеоданих у словнику пропонується наступна схема метаданих, яка приведена у таблиці.

Таблиця

Елементи <i>VideoObject</i>		
actor	Person	Актор, наприклад, у TV, радіо, фільмі, відео грі тощо, або в події. Актори пов'язані з індивідуальними елементами або серією, епізодом, кліпом
caption	MediaObject або Text	Підпис об'єкта (наприклад, субтитри). Вказує формат кодування
director	Person	Директор, наприклад, у TV, радіо, фільмі, відео грі або події. Директори пов'язані з індивідуальними елементами або серією, епізодом, кліпом
musicBy	MusicGroup or Person	Композитор звукового треку
thumbnail	ImageObject	Мініатюра для зображення або відео
transcript	Text	Чи цей MediaObject є AudioObject або VideoObject, транскрипція цього об'єкту
videoFrameSize	Text	Розмір фрейму
videoQuality	Text	Якість відео

Метадані документів та веб-контенту. Поняття метаданих для документів визначене у стандарті ISO 15489-2001 «Інформація та документація. Управління документами» [16]. Відповідно до цього стандарту документ окрім свого інформаційного змісту повинен містити метадані або бути постійно зв'язаним з метаданими, необхідними для виконання дій з документами. Це потрібно для забезпечення незмінності структури документа (його формату та взаємозв'язків між складовими елементами) при виконанні різних операцій з електронним документом. Наявність метаданих робить зрозумілими об-

ставини, за яких документ був створений, отриманий та використаний, – діловий контекст (включаючи відомості про те, частиною якого ділового процесу є виконана операція, про дату, час та учасників ділової операції).

Міжнародний стандарт ISO 15836-2003 («*Information and documentation – The Dublin Core metadata element set*») визначає універсальний набір метаданих для будь-яких інформаційних ресурсів. Стандарт покладений в основу практично всіх наборів метаданих, що описують текстові контенти.

Будь-який документ характеризується змістом (контентом) та структурою.

Контент – інформація документа, що фіксує управлінську діяльність.

Структура – зовнішній вигляд та розміщення частин контенту (наприклад, носій, формат, організація даних, розміщення реквізитів, шрифти, примітки і т. і.), та наявність у документі зв'язків з іншими документами (гіперпосилань).

На базі Дублінського ядра було розроблено ГОСТ 7.70-2003 «*Опис баз даних та машиночитаних інформаційних масивів. Склад та позначення характеристик*», який вводить нескладну систему опису інформаційних, зокрема, мережевих, ресурсів. Для опису інформаційного ресурсу стандарт пропонує використовувати 29 характеристик, обов'язковими з яких є для всіх типів ресурсів:

- 1) ідентифікатор інформаційного ресурсу (посилання на ресурс);
- 2) назва ресурсу;
- 3) власник;
- 4) опис (зміст ресурсу, влючаючи анотацію або реферат, або опис контенту візуальних, аудіо- або мультимедійних ресурсів);
- 5) коди рубрикатора – тематика ресурсу, що виражена кодами стандартного переліку тематичних рубрик;
- 6) ключові слова;
- 7) мова (для текстового ресурсу);
- 8) період оновлення;
- 9) фінансування – форма фінансування при створенні та введенні ресурсу;
- 10) дата останнього оновлення вмісту або дата створення.

Умовно обов'язкові:

- 1) дата останнього оновлення метаданих інформаційного ресурсу;
- 2) адреса в мережі;
- 3) консультант – особа, до якої треба звертатися за додатковою інформацією;
- 4) дата реєстрації інформаційного ресурсу;
- 5) служба, що реєструє інформаційний ресурс.

В стандарті визначені також факультативні реквізити, які можуть надати додаткову інформацію про інформаційний ресурс, зробити його зручнішим: «Творець», «Учасник», «Права», «Ресурс-джерело», «Обмеження доступу», «Об'єм» тощо.

Більшість комп'ютерних програм автоматично створюють деякі метадані та асоціюють їх з файлами. Наприклад, кожний документ MS Word має певний набір властивостей файлу (назва, автор, розмір файлу тощо), який визначається автоматично або вручну. Автоматично формується особливий тип метаданих, так звані системні аудиторські перевірки, які фіксують дії, які виконуються з окремим документом (наприклад, дату дії, вид дії, особа, що її ініціювала).

Великий відсоток веб-контента генерується спеціальними ПЗ, наприклад, генераторами сайтів. Такі програмні засоби дозволяють генерувати пости, тобто веб-контент, фактично, дані, які супроводжуються метаданими. Кожний такий «генератор» може мати власні вимоги до метаданих та використовувати власні механізми створення метаданих та протоколи їх зберігання. Так, у Wintersmith [17] обов'язковими вважають три види метаданих: *template* визначає шаблон, що використовується для рендерінга, *title* – назву поста, *date* – його дату. Але існує можливість додавання будь-яких метаданих.

Метадані, як правило, мають власний формат представлення (окремий від формату даних). Одним з найбільш розповсюджених є YAML формат [18, 19]. Широке використання YAML отримав завдяки тому, що він орієнтований на зручність введення/виведення типових струк-

тур даних багатьох мов програмування, тобто даних багатьох форматів, включаючи Json об'єкти. Він є дружнім до людини форматом серіалізації даних, концептуально близький до мов розмітки. Дозволяє визначити будь-які метадані відповідно до вимог бізнес – застосунку. Json об'єкти, в свою чергу, довели свою зручність та доцільність з розвитком великих даних та мережевих технологій, як стандартний текстовий формат для представлення структурованих даних на основі синтаксису об'єкта JavaScript, що, зазвичай, використовується для передачі даних до веб-застосунків.

Метадані у соціальних мережах

Найрозповсюдженішими схемами метаданих, що використовуються у соціальних мережах, є Twitter Card, розроблений Twitter та Open Graph (OG), запропонований Facebook. Ці протоколи метаданих виконують одну й ту саму функцію – забезпечення найкращого досвіду взаємодії з користувачем при розповсюдженні інформації через соціальні платформи.

Twitter Cards та OG [20] це два окремі набори метаданих. Багато соціальних платформ зчитують їх частину та відображають у відповідному вигляді. У Twitter можна побачити результат у вигляді анотацій до новин чи зображень у стрічці. Результат використання OG можна побачити, наприклад, у Facebook у вигляді блока-анотації з заголовком, описом та зображенням, що характеризують інформаційний блок.

За основу протоколу OG, при його створенні, було покладено стандарт Dublin Core Metadata Element Set. Спочатку Facebook створив цей стандарт для власного використання. Тому, він є досить складним та містить функції, які потрібні лише Facebook. Але, інші платформи, такі як LinkedIn та Google+, також успішно використовують метадані OG.

Twitter Cards служить лише одній меті: наповненню інформаційних блоків, які відображаються у Twitter та застосунках, що працюють з цим сервісом. Невеличкі відмінності між Twitter Cards та OG обумовлені лише специфічними особливо-

стями Twitter. Унікальність Twitter як соціальної платформи полягає у розповсюдженні інформації через ретвіти, що створює певні проблеми. Обмеження твіту у 140 символів не завжди дозволяє додати коментар або інформацію про автора чи власника контенту. Twitter Cards дозволяє офіційно визначити у метаданих інформацію про автора та правовласника. Twitter досить вміло сформували власну схему метаданих, щоб не відставати від інших соціальних платформ. Автори та власники сайтів, які реалізували підтримку схеми Twitter Cards, тепер можуть надавати більше інформації про контент, що розповсюджується через Twitter.

Будь-які застосунки, що забезпечують функції соціальної мережі, підтримують відправлення повідомлень наступних типів [21]: текстові повідомлення; текстові повідомлення з приєднаним зображенням, аудіо, відео чи просто будь-яким файлом; повідомлення з приєднаним шаблоном; повідомлення з прикріпленим відступом. Відповідно, для кожного повідомлення зберігаються загальні властивості (відправник, отримувач, властивості, що ідентифікують повідомлення) та властивості, специфічні для обраного типу повідомлення (текст повідомлення, тип приєданого файлу, посилання на нього тощо).

Обидві схеми також спрямовані на надання інформації про контент найпопулярніших типів:

- для статей формується текстова анотація і зображення для попереднього перегляду;
- для аудіо-записів додається аудіоплеєр;
- для відеокліпів – відеоплеєр;
- для зображення – можливість їх попереднього перегляду.

Протоколи Twitter Cards та OG є досить схожими та мають специфічні властивості, що визначаються для передвизначених типів контенту. Розробники визначили декілька типів контенту, таких як відеокліпи, аудіофайли, веб-сторінки та інші, та запропонували для кожного з них власний набір мета-властивостей. Коли користувач ділиться аудіофайлом або ві-

деокліпом, для їх відтворення на основі метаданих обирається відповідний плеєр.

Далі наводиться перелік мета-тегів, що надають детальну інформацію про статтю в OG та Twitter Cards.

OG:

```
<meta property="og:type"
content="article">
<meta property="og:url"
content="URL об'єкта">
<meta property="og:site_name"
content="Назва ресурсу, де розміщена
стаття">
<meta property="og:image"
content="URL зображення для статті">
<meta property="og:title"
content="Заголовок статті">
<meta
property="og:description"
content="Опис статті">
<meta
property="article:author"
content="URL сторінки автора статті">
<meta
property="article:section"
content="Розділ, до якого відноситься
стаття">
<meta property="article:tag"
content="Ключові слова">
```

Twitter Cards:

```
<meta name="twitter:card"
content="summary">
<meta name="twitter:url"
content="URL статті">
<meta name="twitter:title"
content="Заголовок статті">
<meta
name="twitter:description"
content="Опис статті">
<meta name="twitter:image"
content="URL зображення для статті">
```

Наступні опціональні елементи Twitter Card дозволяють визначити ідентифікатор автора або організації автора контенту у Twitter:

```
<meta name="twitter:site"
content="@username">
<meta name="twitter:site:id"
content="Twitter ID">
<meta name="twitter:creator"
content="@username">
<meta
name="twitter:creator:id"
content="Twitter ID">
```

Протокол Open Graph [22]. Визначає метадані для опису веб-сторінок, та перетворює їх у вузли соціальної мережі. Хоча існує багато різних технологій і схем, які можна було б об'єднати разом, не існує єдиної технології, яка надає достатньо інформації для повномасштабного представлення будь-якої веб-сторінки у соціальній мережі. Протокол OG побудований на цих існуючих технологіях. Його головною метою є простота для розробників. Протокол містить базові елементи метаданих, опціональні елементи метаданих, структуровані властивості для аудіо, відео та зображень, спеціальні властивості для музичних записів та визначення типів об'єктів.

До базових елементів метаданих відносяться:

- назва об'єкта;
- тип об'єкта;
- URL на зображення, що представляє об'єкт у графі соціальної мережі;
- URL на об'єкт, що використовується як ідентифікатор;
- опціональні та загально рекомендовані елементи метаданих:
 - URL на аудіофайл, що супроводжує об'єкт;
 - стислий опис об'єкта;
 - слово, що виводиться перед назвою об'єкта (за замовченням пуста);
 - місцева мова;
 - масив інших доступних на сторінці мов;
- назва сайту, якому належить об'єкт;
- URL на відеофайл, що доповнює даний об'єкт.

Деякі властивості також можуть мати додаткові метадані [23].

Протокол Twitter Card (TC). Зміст протоколу визначається, перш за все, потребами користувачів Twitter. Користувачі Twitter організують людей, яким вони слідують, у списки, постять текст та відео, використовують хеш-теги для визначення коментарів у твіті та пов'язують їх один з одним, ретвітять чужий контент з/без коментарів, виділяють “улюблені” твіти, керують властивостями, такими як «список популярних тем». Глибина даних про сус-

пільство, що представлена контентом у Twitter, та його метадані призвели у 2010 році до заключення угоди про те, що Бібліотека Конгресу США буде архівувати цей цінний матеріал для досліджень [24].

Перелік елементів метаданих TC за складом дуже близький до OG, що обумовлюється, як було вже зазначено вище, єдиним набором типів контентів, які вони намагаються структурувати та описати. Існує 5 різних типів карт, відповідно до різних об'єктів, які вони описують [25]:

- зведена карта;
- зведена карта з великим зображенням;
- карта програвача;
- карта застосунку;
- карта Lead Generation.

Зведена карта може використовуватися постами блогів, новинами, сторінками продуктів та іншими бізнес-новинами. Їх мета полягає у наданні такої інформації, як назва, опис та зображення, що супроводжує даний пост.

Зведена карта з великим зображенням відрізняється лише допустимими розмірами зображення.

Карта застосунку може використовуватися мобільними застосунками. Вона надає інформацію про назву, опис, іконку, рейтинг та ціну. Щоб витягнути цю інформацію Twitter може використовувати ідентифікатори застосунків.

Карта програвача була створена, щоб рекламувати мультимедійні потокові мультимедіа, такі як аудіо- чи відео-програвач у Twitter. Містить такі дані як: опис, зображення, програвач.

Карта *Lead Generation* підходить для спілкування з потенційними клієнтами. Вона дозволяє зібрати інформацію про перспективи. Дані потенційних клієнтів (ім'я, контакти), зазвичай, вже введені та їх не треба заповнювати ще раз. Карта містить таку інформацію як: назва, повідомлення посту, зображення, заклик до дії.

Щоб глибше усвідомити значимість та обсяги метаданих у соціальних мережах, розглянемо невеличкий приклад [26] метаданих Twitter (рис. 3), що пов'язані з твітом лише у 140 символів.

Останні не представляють великих обсягів даних, однак об'єми їх вибухають, якщо зв'язати твіт з усіма метаданими, що необхідні для розуміння цих 140 символів у контексті розмови.

Наведений далі приклад (рис. 3) демонструє підмножину повного переліку елементів метаданих протоколу ТС, а саме містить наступні елементи:

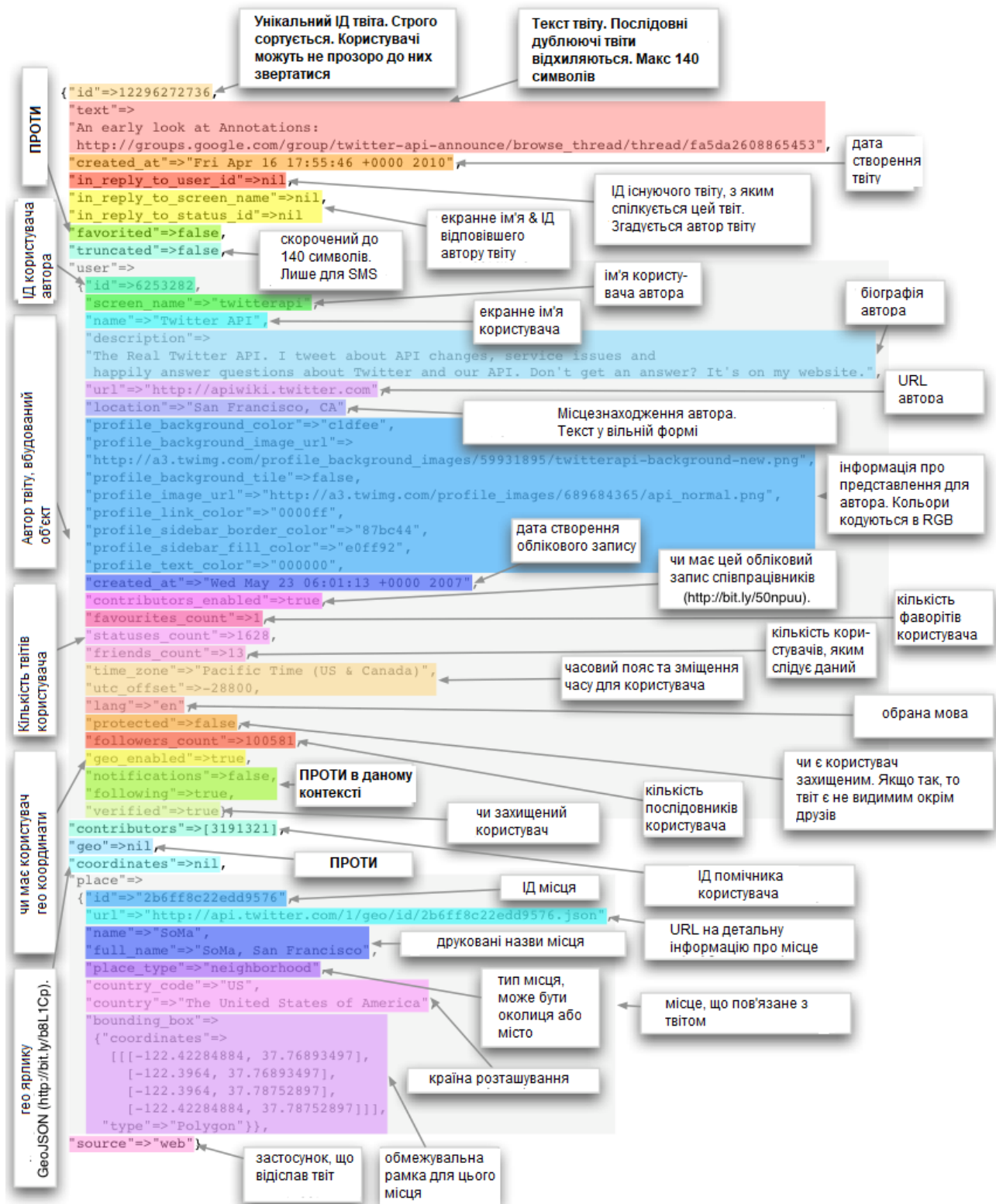


Рис. 3. Метадані, пов'язані з твітом

- ім'я та ідентифікатор користувача, що відповів автору твіта;
- дата та час створення твіту;
- ім'я автора;
- ім'я користувача;
- біографія автора;
- URL автора;
- місцезнаходження автора;
- надання інформації для автора;
- дата створення облікового запису;
- кількість обраних, що має користувач;
- кількість користувачів, на яких підписаний даний користувач;
- часовий пояс та зміщення часу для даного користувача;
- мова, обрана користувачем;
- чи є користувач захищеним;
- кількість підписників користувача;
- ідентифікатор місця;
- друкована назва цього місця;
- тип місця;
- країна;
- застосунок, який відіслав твіт.

Особливістю даного прикладу є те, що всі метадані у деякій мірі структуровані. Вони збираються таким чином, щоб виконувати корисну задачу, та сортуються за відомими категоріями. Саме це поняття структури перетворює необроблену інформацію у реальні метадані.

Метадані сховищ даних

Склад та зміст цього класу метаданих визначається цілями їх використання у сховищах даних. А саме, метадані, у даному випадку, повинні сприяти мінімізації робіт з розробки та адміністрування сховища даних та забезпечувати більш ефективне витягнення інформації із сховища. Метадані містять всю інформацію, що необхідна для витягнення, перетворення та завантаження даних з вихідних систем, а також для наступного використання та інтерпретації вмісту сховища.

Ральф Кимболл виділив наступні типи метаданих:

- метадані вихідної системи:

- специфікації джерел даних, таких як репозиторії;
- описова інформація (частота оновлення, юридичні обмеження, методи доступу);
- інформація про процеси, наприклад, графік завдань та коди витягнення.
- метадані перетворення даних:
 - інформація про отримання даних (планування передачі даних, відомості про використання файлів);
 - керування таблицями вимірювань, (визначення вимірювань та присвоєння сурогатних ключів);
 - перетворення та агрегування, (розширення та відображення даних, програми завантаження СКБД, визначення агрегатів даних);
 - документування перевірок, робіт та журналів (журналів перетворення даних й записів стеження за походженням даних.
- метадані СКБД:
 - зміст системних таблиць СКБД;
 - рекомендації з обробки.

Ці метадані можуть використовуватися трьома способами:

- *пасивно*, забезпечуючи чітку документацію щодо структури, процесу розробки та використання системи Сховища даних. Документація необхідна як кінцевим користувачам, так і системним адміністраторам та розробникам застосунків;

- *активно*, шляхом збереження конкретних аспектів (наприклад, правил перетворення) у вигляді метаданих, які можна інтерпретувати та використовувати під час виконання. У даному випадку, метадані керують процесами сховища даних. Як наслідок, код (активні метадані) та додаткова документація погоджено та уніфіковано керуються в одному репозиторії, при цьому зростає актуальність документації;

- *напівактивно*, за рахунок зберігання статичної інформації (наприклад, визначень структур, специфікацій конфігурацій), яку буде зчитувати інший програмний компонент під час виконання.

Наприклад, обробникам запитів необхідні метадані для перевірки існування атрибутів. На відміну від активного використання, тут метадані лише читаються, але не виконуються.

Метадані сховищ дозволяють вирішувати наступні задачі відповідно до вище визначених цілей [27]:

- *підтримка інтеграції систем.* Схеми та інтеграція даних залежать від метаданих, що описують структуру та сенс окремих джерел даних та цільових систем. Правила перетворення можна застосувати до вихідних даних та зберігати як метадані. Більше того, інтеграція різних інструментів можлива лише, якщо вони розділяють «дані», які є метаданими системи сховища;

- *підтримка аналізу та проектування нових застосунків.* Метадані підвищують контрольованість та надійність процесу розробки застосунків, забезпечуючи інформацію про зміст даних, їх структуру та джерела. Більше того, метадані, що стосуються рішень з проектування застосунків, можна використовувати повторно;

- *підвищення гнучкості системи та можливості повторного використання існуючих програмних модулів.* Це можливо лише для активного та напівактивного використання даних. Семантичні аспекти, які швидко змінюються, зберігаються явним чином у вигляді метаданих прикладних програм. Тому, підтримка є суттєво простішою. Систему можна розширити та адаптувати без будь-яких труднощів. Даний підхід дає можливість повторного використання «фрагментів коду»;

- *автоматизація адміністративних процесів.* Метадані керують запуском різних процесів Сховища даних (наприклад, завантаження та оновлення). Інформація про їх виконання (журнали доступу, кількість записів та ін.) також міститься у репозиторії, доступному адміністратору;

- *підсилення механізмів безпеки.* Метадані повинні забезпечити правила доступу та права користувачів для всієї системи сховища. Керування доступом у сховищі іноді вимагає застосування складних методів. Наприклад, оперативне дже-

рело може містити нешкідливу інформацію про окремі показники роботи компанії, але сумарні значення у сховищі іноді виявляються важливим секретом. З іншого боку, персональні доходи кожного співробітника є тайною, але при цьому підсумок грошової одиниці у сховищі може взагалі не бути критичною інформацією;

- *підвищення якості даних.* Якість даних визначається наступними характеристиками:

- погодженість (однорідність представлення, відсутність дублікатів, даних з визначеннями, що конфліктують);

- повнота (всі дані присутні)

- точність (співпадання збережених та фактичних значень);

- своєчасність (актуальність значення, що зберігається);

- правила перевірки якості даних необхідно визначати, зберігати у вигляді метаданих та перевіряти при кожному оновленні сховища. Окрім цього, висока якість вимагає підтримки контролю даних. Метадані забезпечують інформацію про час створення, про автора даних, про джерело, значення даних в момент отримання (про спадковість даних), й про подальший шлях від джерела до поточного місцезнаходження (походження даних). Таким чином, користувачі можуть відновити ланцюг проходження даних за час перетворення, та перевірити точність інформації, що повертається;

- *покращення взаємодії в системі сховища.* Взаємодія відбувається як через виконання простих запитів та звітних застосунків, так й з використанням складних аналітичних інструментів. Метадані забезпечують відомості про значення даних, термінологію та бізнес-концепції підприємства, а також їх зв'язок з даними. Тому, метадані підвищують якість запитів, що виконуються, за рахунок більш точного та строгого формулювання, а також скорочують витрати на користувачів, яким потрібний доступ, оцінку та застосування відповідної інформації;

- *покращення аналізу даних.* Методи аналізу даних представлені широко – починаючи від простих застосунків звітності та OLAP, та, закінчуючи склад-

ними застосунками data mining. В цьому напрямку метадані необхідні для розуміння предметної області та її представлення у сховищі, для адекватного застосування та інтерпретації результатів;

– застосуванню загальної термінології та мови взаємодії у корпорації. Доступність метаданих як унікального джерела документації для користувачів має й інші переваги. Вона гарантує погоджені дії та інтерпретації інформації із сховища, а також запобігає двозначності та забезпечує погодженість відомостей у компанії, дозволяє розділити знання та досвід.

Метадані системи сховища містяться в репозиторії – структурованій системі зберігання та витягнення, що реалізована на основі СКБД. Для інтерпретації метаданих необхідно зберігати структуру репозиторія (тобто схему метаданих) та їх семантику.

Існують різні способи визначення та зберігання метаданих у сховищі, один з яких є використання технології XML, що завдяки своїм перевагам, таким як зрозумілість, відкритість технології, гнучкість, є дуже зручним засобом опису. Він дозволяє публікувати метадані, що використовуються будь-якою програмою або БД, та забезпечує зв'язок між структурованою базою та неструктурованим контентом. Якщо є ПЗ, яке здатне прочитати та розшифрувати XML-файли, то метадані у будь-якому сховищі можна представити у вигляді звичайного XML-файлу, що створений на основі загального DTD (опис типу документу), а переваги XML представлення є очевидними. Але якщо буде забагато XML файлів, написаних відповідно до різних стандартів, то це зробить їх обробку досить складною. Тому, ефективна обробка метаданих, представлених таким чином, вимагає вирішення наступних проблем XML-середовища:

- погоджений стиль тегів;
- несуперечливі погодження про іменування;
- сумісне визначення тегів;
- керування XML об'єктами для їх наступного повторного використання;

- інструменти для динамічної перевірки DTD та схем;
- документовані набори кодів;
- чітко визначені простори імен бізнес-моделі;

Метадані в екосистемі Hadoop

Hadoop використовує метадані для ефективного керування даними. Метадані вбудовуються у самі дані при проходженні ними різних систем підприємства (робиться це за допомогою визначення спеціальних тегів). Більше того, метадані розширюються та включають додаткову інформацію окрім звичайних атрибутів, таких як: розмір файлу, роздільна здатність, дати модифікації тощо. Наприклад, до метаданих можуть бути включені відомості про бізнес, що може допомогти визначити корисність даних у конкретній моделі. Нарешті, на відміну від самих корпоративних даних метадані можуть бути централізовані на єдиній платформі.

HDFS здатна присвоювати розширені атрибути, що також дозволяє збагатити метадані. Але це не завжди підходить для великих даних. Тому, виникають альтернативні підходи, як, зокрема, визначення тегів (Apache Atlas) та створення централізованого сховища метаданих, а користувачі дружніх до Hadoop систем витягування даних (наприклад, Hive та Spark SQL) можуть визначати теги самостійно.

Задачі роботи з метаданими і сервіси для аналізу та обробки метаданих

Досить швидкий розвиток мережевих технологій та зростання обсягів даних різних типів обумовлюють нагальні вимоги як до самих метаданих, так й до інструментів, що дозволяють якимось ними користуватися. Метадані не мають сенсу, якщо вони не є «корисними», тобто немає можливості їх ефективного використання для вирішення задач користувачів. Тобто, необхідно мати розвинені методи та засоби, що дозволять:

- створювати метадані;
- читати метадані;
- редагувати метадані;

- організовувати ефективне зберігання метаданих;
- керувати метаданими;
- аналізувати метадані;
- оптимізувати метадані;
- використовувати метадані при пошуку інформації.

Насьогодні розроблено велику кількість доданків чи окремих застосунків, що намагаються вирішити перелічені задачі. Деякі з них просто допомагають переглянути чи вивести інформацію метаданих, інші додають інструменти для створення та редагування, а деякі є потужними інструментами аналізу метаданих, що дозволяють робити висновки про достовірність контенту тим чи іншим чином та навіть будувати статистичні звіти та прогнози, на основі аналізу метаданих.

Керування метаданими

Найважливішою особливістю метаданих є впорядкована структура. Інформація категоризована та має конкретну форму/формат. Завдяки структурованому вигляду, метадані є доступними для читання не лише людиною, але й комп'ютерами. Таким чином, метадані можуть бути оброблені автоматизовано та використані для різних цілей: індексація, пошук, об'єднання або автоматична обробка. Зокрема, одним з прикладів використання метаданих є пошук зображення на основі Exif значення поля. Це, наприклад, можна зробити в програмі Google Picasa, де передбачені спеціальні команди, які вводять значення в поле пошуку у застосунку. В Інтернеті, у галереях зображень стандартом є визначення метаданих поряд з фотографією, що переглядається. Сайти активно використовують дані з EXIF, наприклад, для «прив'язки» фотографії до географічної мапи. Так, у галереї Google, Google+ або Picasa Web Albums поряд з фото з'явиться мапа з попереднім переглядом.

Створення метаданих, перегляд, читання та редагування

Базовий набір метаданих визначається автоматично застосунком при створенні файлу даних конкретного типу. Так,

для файлів фотографій за це відповідає ПЗ, яке встановлене на цифровій камері.

Щоб переглянути метадані, перш за все, необхідно витягнути їх з файлу. Отримати доступ до метаданих веб-об'єктів можна за допомогою доданків до веб-браузера або спеціальних сервісів. Деякі з інструментів, що беруть на себе такі функції, підтримують лише один формат файлу (наприклад, JPEG), інші – багато форматів. Окрім цього, різні програми можуть підтримувати різні типи метаданих.

Серед прикладів інструментів витягування метаданих можна перелічити [28].

1. *Exiv2* [29] – ПЗ з відкритим кодом, яке декодує EXIF, IPTC та XMP метадані.

2. *ExifTool* [30] – один з найбільш потужних засобів витягування метаданих (через командну строку). Підтримує сотні різних форматів файлів та метаданих (PDF, Djvu, JPEG, AVI, MOV, MP3 тощо), включаючи дуже специфічні. Написаний на Perl, добре документований, добре працює під Linux та Mac, але може бути проблемним для користувачів Windows, у яких не встановлений Perl.

3. *Adobe Photoshop* – комерційний застосунок, що включає XMP переглядач. Хоча він не такий потужний або повний як Exiv2 або ExifTool, забезпечує можливість декодування XMP, IPTC, Exif та інші типи метаданих у графічному інтерфейсі.

4. *PreviewInspector*. За замовчуванням Apple Mac OS X для перегляду зображень використовує ПЗ Preview, яке містить спеціальний 'Inspector' для перегляду метаданих. Інструмент відображає невеличку частину доступних метаданих та може забезпечувати не достовірні результати аналізу, тому не рекомендований для використання в офіційних розробках.

Зручним інструментом для читання метаданих графічних файлів також є браузер графічних файлів *IrfanView*, за умови, що встановлено плагін, який включає бібліотеку для декодування Exif. Хоча там не вистачає можливості редагування Exif, IrfanView дозволяє створювати опис у форматі IPTC. З метаданими також чудово ладнають всі ПЗ для обробки цифрових фотографій. Для не професійного викорис-

тання, можна рекомендувати програму *Google Picasa*, яка має панель, що дозволяє перевірити всі дані фотографії. Нажаль, можливості редагування метаданих обмежена. Тому, у випадку більших вимог, слід звернути увагу на *Adobe Lightroom* [31], що містить дуже складні інструменти для перегляду та редагування метаданих зображень. Його використовують професійні фотографи. Програма містить також спеціальні інструменти, що використовують метадані для автоматичної корекції кольорів зображень. *Lightroom* дозволяє визначати тип датчика, тип об'єктива, фокусну відстань та на цій основі покращити геометрію зображення.

Розвинені можливості щодо створення та редагування метаданих надає відома платформа для побудови клієнтських застосунків WPF (Windows Presentation Foundation). Вона дозволяє обробляти XMP, IPTC та EXIF метадані та використовує для цього спеціальні класи. При цьому, відповідні поля у метаданих різних стандартів знаходяться у наступному порядку: XMP, IPTC та EXIF. Запис тегів метаданих виконується у XMP форматі. Окрім цього, для читання та запису метаданих можуть використовуватися функції *GetQuery/SetQuery*, які працюють з ієрархічними іменами тегів метаданих. Існують спеціальні класи, що дозволяють працювати з конкретними форматами зображень. А клас *InPlaceMetadataWriter* дозволяє змінювати метадані на місці, без перекодування файла.

Досить потужним інструментом для редагування тегів аудіофайлів є *TagScanner*. Він вміє редагувати у пакетному режимі теги більшості сучасних аудіоформатів. Підтримуються теги ID3v1 та ID3v2, Vorbis Comments, APEv2, WMA та MP4 (iTunes). Застосунок дозволяє змінювати назву файла за інформацією з тегів, генерувати тег за назвою файла/директорії або виконувати будь-які перетворення та змінення тексту в тегах та іменах файлів. Програма має розвинені можливості для отримання інформації та роботи з музичним альбомом чи архівом.

Розвинені можливості керування метаданими в XMP форматі надає застосу-

нок *Premiere Pro* [32]. Він відкриває загальний доступ до метаданих, а також дозволяє їх переглядати, створювати, видаляти, редагувати та шукати.

Аналіз метаданих

Для перевірки наявності метаданих та їх аналізу у фотографіях (наприклад, визначення місця та часу, коли воно було зроблене, чи було воно відредаговане) можна використовувати спеціальні он-лайн ресурси [33]. Одним з таких ресурсів є *Jeffray's Exif Viewer*, розроблений та викладений у відкритий доступ американським програмистом. Він відображає всю доступну інформацію з метаданих. Аналогічно працює інший схожий ресурс для перевірки метаданих *FindEXIF.com*. Але в ньому відсутня можливість завантаження фотографії. Сервіс працює лише з посиланнями. Фотографії з конкретних географічних місць можна також шукати за допомогою *Panoramio*. Цей сервіс використовує EXIF-дані для публікації фотографій на мапі. Дозволяє при розміщенні фотографії визначити її координати. Розібратися в локаціях допомагають також сервіси *Google Maps* та *Wikimapia*.

Сервіс *FotoForensics* дозволяє визначити, чи було фото відредаговане. Сервіс працює як з завантаженими фотографіями, так й з посиланнями на них. Окрім того, що сервіс виводить доступні метадані (дату створення, дату редагування тощо), він пропонує ELA (Error Level Alysis) рівень стискання файлів. Це свого роду сканер, який показує маніпуляції з зображенням, навіть, якщо їх не видно на перший погляд. Знаючи специфіку цих даних, можна ефективно визначати масштаби та тип редагування знімка, наприклад, чи був використаний фотомонтаж при редагуванні зображення.

Інструменти *Google Search by Image* та *TinEye* забезпечують можливості зворотнього пошуку зображень, тобто користувач може завантажити до сервісу фото та знайти його оригінальне джерело й подивитись, де воно ще публікувалося.

Програмний застосунок *JPEGSnoop* дозволяє дивитися метадані не лише зображень, а й форматів AVI, DNG, PDF, THM,

але працює тільки для Windows. Дозволяє перевірити, чи було зображення редактоване, виявити помилки у пошкодженому файлі тощо.

Сервіс *Pipl.com* призначений для пошуку «Інтернет-сліду» користувача, допомагає його ідентифікувати та знайти його контент (фотографії, файли тощо). Програма проводить пошук у всіх соціальних американських мережах (Facebook, LinkedIn, MySpace) – для цього потрібно ввести ім'я та прізвище латиницею. Особливість програми в тому, що вона веде пошук по «глибокому Інтернету», який ігнорується звичайними пошуковими системами та недоступний для користувачів.

Ресурс *WebMii* шукає посилання з визначеним ім'ям людини, дає рейтинг «веб-видимості», за допомогою якого можна встановити фейкові акаунти. Завдяки інструменту кожен може знайти згадування власного імені на іноземних ресурсах.

Застосунок *Geofeedia* є «куратором соціальних мереж», який агрегує результати не за ключовими словами чи хеш-тегами, а за заданим місцем розташування. Сервіс обробляє повідомлення з Twitter, Flickr, Youtube, Instagram та Picasa, надіслані з використанням GPS, і потім представляє їх у вигляді колажу. І хоча значну кількість повідомлень він не охоплює, надає загальну картину. Сервіс платний, безкоштовною надається лише демоверсія.

Окремої уваги заслуговує сервіс *Wolfram Alpha* [34]. Це навіть не пошукова система, а база знань з науковим ухилом, інтелектуальний робот, який може відповідати на будь-які питання. Але він орієнтується лише в темах, які стосуються точної, більш енциклопедичної інформації, а не поточних подій. Він не надає посилань на інші сайти, а видає вже готовий варіант відповіді. Спочатку він був рекомендований для перевірки погоди, в тому числі використовувався для перевірки достовірності зображень через перевірку погоди за датою та місцем зображення. Але, насправді ця система може порівнювати відомі світові компанії за безліччю показників, а також міста, країни, відомих осіб, будівлі. Програма також містить багато еко-

номічної інформації, у тому числі може вираховувати прогнози, наприклад, ціна на газ та нафту в довгостроковій перспективі; може вирішувати складні математичні приклади, давати детальну інформацію про поживну цінність різних продуктів, показувати карту зоряного неба для різних точок земної кулі. Окрім того, вираховує індекс маси тіла (потрібно ввести свої дані) та ризику захворюваності; визначає час, потрібний для читання чи написання певної кількості слів та багато іншого. Нарешті, одна з найцікавіших можливостей інструменту – це аналіз статистики користувача у Facebook. Програма формує повний звіт із інфографікою, який показує активність користувача від моменту реєстрації, кількість завантажених лінків та фото, також показує активність упродовж доби й аналізує середню довжину постів. Крім того, можна подивитися статистику по друзях за різними показниками – вік, стать, сімейний стан, місце проживання на карті та мережеві зв'язки друзів між собою.

Питання оптимізації метаданих

Як було продемонстровано прикладом вище, обсяги метаданих часто можуть перевищувати обсяги самих (великих) даних. Тому, якщо документ містить велику кількість метаданих, доцільно зберігати метадані в різних файлах та використовувати дві URL-адреси – окремо для метаданих та для самих даних. Ці дві сторінки можна зв'язати за допомогою вказівників. Це дозволить суттєво підвищити продуктивність роботи з даними. Цей варіант ідеально підійде для адаптивних сайтів: матеріали однієї й тієї самої сторінки можна показувати як у браузері комп'ютера, так і на мобільному пристрої.

Facebook пропонує [35] також для застосунків, що використовують піддомени з метою розміщення додаткових версій, які оптимізовані для мобільних пристроїв, запобігати додавання додаткових даних до представлення сторінок для мобільних пристроїв. Для цього пропонується використовувати канонічні URL, що вказують на представлення цих сторінок для перегляду на комп'ютері.

Висновки

Проведені дослідження продемонстрували нагальність та масштаби проблем у галузі обробки різнорідних не структурованих даних великих обсягів – величезні, безперервно зростаючі обсяги різноманітної, слабкоструктурованої (чи взагалі не структурованої) інформації, велика кількість розрізнених сервісів для обробки їх метаданих, що, як правило, описують лише технічні характеристики даних, відсутність семантичних описів даних та єдиної ефективної системи керування метаданими.

Головною перевагою метаданих є їх структурованість, що забезпечує можливість їх автоматизованої обробки та використання для роботи з великими даними, а саме: для індексації, пошуку, об'єднання, автоматичної обробки та ін. Це набуває особливого значення при роботі з не структурованими даними та сприяє вирішенню складних бізнес-задач.

Але створення ефективної системи керування метаданими, вимагає їх узгодженої загальної класифікації, в першу чергу, з урахуванням походження самих даних, їх цілей, форматів представлення, задач, у вирішенні яких ці дані використовуються. Тому, мета даного дослідження – це аналіз можливих джерел великих даних та визначення притаманних ним характеристик.

В результаті проведеного аналізу було визначено 8 типів джерел, а саме:

- архіви відсканованих документів;
- документи-файли різних форматів;
- сховища даних;
- бізнес – застосунки;
- публічний веб;
- засоби масової інформації;
- дані журналів;
- автоматично генерований контент.

Аналіз перелічених джерел дозволив визначити наступні основні типи контенту відповідних даних: зображення, аудіо-, відео- файли, документи, повідомлення, дані з датчиків, веб-сторінки. Для представлення вказаного контенту можуть бути використані різні формати, що може впливати на вибір протоколу метаданих, що використовується. Насьогодні, розроб-

лено велику кількість протоколів метаданих, що підтримують визначення метаданих для одного чи кількох форматів представлення даних одного чи декількох типів, однак не існує єдиного підходу чи протоколу, який надавав би загальну таксономію характеристик. Окрім цього, більшість таких протоколів визначають досить детальні, але технічні характеристики даних. Опис семантики даних обмежується, зазвичай, двома елементами: Title та Description. Найбільш гнучкими є протоколи, що надають XML формат представлення метаданих, як XMP специфікація. Такий формат представлення дозволяє включити до контенту будь-яку необхідну метаінформацію та використовувати для визначення семантики контенту цільові онтології. Слід зазначити, що на відміну від технічних характеристик, що зазвичай генеруються автоматично, визначення семантичних характеристик неможливе без участі спеціалістів або експертів, а їх застосування при вирішенні задач вимагає спеціальних програмних засобів, які будуть вміти їх розпізнавати та використовувати.

Розуміючи величезну роль метаданих для забезпечення семантичного визначення контенту великих даних, потрібно пам'ятати, що часто розміри метаданих значно перевищують обсяг самих даних (навіть великих), та дотримуватися принципів розумної ефективності при визначенні елементів метаданих та організації їх зберігання та обробки.

Література

1. <https://habr.com/ru/post/93119/>
2. <https://www.exif.org/category/specifications>
3. <http://exif.org/dcf.PDF>
4. <https://helpx.adobe.com/after-effects/using/xmp-metadata.html>
5. <https://www.dublincore.org/specifications/dublin-core/dces/>
6. ISO 16684-1:2012, Graphic technology – Extensible metadata platform (XMP) specification – Part 1: Data model, serialization and core properties
7. <https://www.adobe.com/devnet/xmp.html>

References

8. <https://forum.allnokia.ru/viewtopic.php?t=51934>
9. <https://habr.com/ru/post/103635/>
10. <http://id3.org/id3v2.3.0>
11. <http://www.xspf.org/xspf-v0.html>
12. https://www.ibm.com/support/knowledgecenter/ru/SS88XH_1.6.0/iva/ov_metadata.html
13. https://mediaarea.net/AVIMetaEdit/tech_view_help
14. <https://stackoverflow.com/questions/2075175/is-there-a-standard-schema-for-video-metadata>
15. <https://schema.org/>
16. https://studref.com/379466/menedzhment/metadannye_dokumentov
17. <http://prgssr.ru/development/posty-dannye-i-metadannye-v-wintersmith.html#heading-section-1>
18. https://symfony.com.ua/doc/current/components/yaml/yaml_format.html
19. <https://uk.wikipedia.org/wiki/YAML>
20. <https://frontender.info/like-able-content-spread-your-message-with-third-party-metadata/>
21. <https://www.ixbt.com/soft/audio-tag-editors.shtml>
22. <http://ogp.me/>
23. <https://hostenko.com/wpcafe/plugins/kak-nastroit-open-graph-i-twitter-karty-dlja-wordpress/>
24. <https://blogs.loc.gov/loc/2010/04/how-tweet-it-is-library-acquires-entire-twitter-archive/>
25. <https://www.oncrawl.com/oncrawl-seo-thoughts/a-complete-guide-to-twitter-cards/>
26. <https://www.datasciencecentral.com/profiles/blogs/importance-of-metadata-in-a-big-data-world>
27. <http://iso.ru/ru/press-center/journal/2122.phtml>
28. <https://fotoforensics.com/tutorial-meta.php>
29. <https://www.exif.org/Exif2-2.PDF>
30. <http://www.belurus.info/soft/i.php?c=exiftool>
31. https://webznam.ru/blog/metadannye_fajlov_fotografij/2015-04-01-135
32. <https://helpx.adobe.com/ru/premiere-pro/using/metadata.html>
33. <https://www.stopfake.org/metadannye-nevidimaya-informatsiya-o-fotografii/>
34. https://ms.detector.media/mediaprovita/how_to/13_onlayninstrumentiv_dlya_perevirki_kontentu
35. https://developers.facebook.com/docs/sharing/webmasters/optimizing?locale=ru_RU
1. <https://habr.com/ru/post/93119/>
2. <https://www.exif.org/category/specifications>
3. <http://exif.org/dcf.PDF>
4. <https://helpx.adobe.com/after-effects/using/xmp-metadata.html>
5. <https://www.dublincore.org/specifications/dublin-core/dces/>
6. ISO 16684-1:2012, Graphic technology – Extensible metadata platform (XMP) specification – Part 1: Data model, serialization and core properties
7. <https://www.adobe.com/devnet/xmp.html>
8. <https://forum.allnokia.ru/viewtopic.php?t=51934>
9. <https://habr.com/ru/post/103635/>
10. <http://id3.org/id3v2.3.0>
11. <http://www.xspf.org/xspf-v0.html>
12. https://www.ibm.com/support/knowledgecenter/ru/SS88XH_1.6.0/iva/ov_metadata.html
13. https://mediaarea.net/AVIMetaEdit/tech_view_help
14. <https://stackoverflow.com/questions/2075175/is-there-a-standard-schema-for-video-metadata>
15. <https://schema.org/>
16. https://studref.com/379466/menedzhment/metadannye_dokumentov
17. <http://prgssr.ru/development/posty-dannye-i-metadannye-v-wintersmith.html#heading-section-1>
18. https://symfony.com.ua/doc/current/components/yaml/yaml_format.html
19. <https://uk.wikipedia.org/wiki/YAML>
20. <https://frontender.info/like-able-content-spread-your-message-with-third-party-metadata/>
21. <https://www.ixbt.com/soft/audio-tag-editors.shtml>
22. <http://ogp.me/>
23. <https://hostenko.com/wpcafe/plugins/kak-nastroit-open-graph-i-twitter-karty-dlja-wordpress/>
24. <https://blogs.loc.gov/loc/2010/04/how-tweet-it-is-library-acquires-entire-twitter-archive/>
25. <https://www.oncrawl.com/oncrawl-seo-thoughts/a-complete-guide-to-twitter-cards/>
26. <https://www.datasciencecentral.com/profiles/blogs/importance-of-metadata-in-a-big-data-world>
27. <http://iso.ru/ru/press-center/journal/2122.phtml>
28. <https://fotoforensics.com/tutorial-meta.php>
29. <https://www.exif.org/Exif2-2.PDF>

30. <http://www.belurus.info/soft/i.php?c=exiftool>
31. https://webznam.ru/blog/metadannye_fajlov_fotografij/2015-04-01-135
32. <https://helpx.adobe.com/ru/premiere-pro/using/metadata.html>
33. <https://www.stopfake.org/metadannye-nevidimaya-informatsiya-o-fotografii/>
34. https://ms.detector.media/mediaprosvita/how_to/13_onlayninstrumentiv_dlya_perevirki_kontentu
35. https://developers.facebook.com/docs/sharing/webmasters/optimizing?locale=ru_RU

Одержано 17.10.2019

Про автора:

Захарова Ольга Вікторівна,
кандидат технічних наук,
старший науковий співробітник.
Кількість наукових публікацій в
українських виданнях – 30.
<http://orcid.org/0000-0002-9579-2973>.

Місце роботи автора:

Інститут програмних систем
НАН України,
проспект Академіка Глушкова, 40.
Тел.: 526 5139.
E-mail: ozakharova68@gmail.com

О.П. Ільїна, О.О. Слабоспицька

ПІДХІД ДО ПОБУДОВИ ТА ВИКОРИСТАННЯ МОДЕЛІ ПРЕДМЕТНОЇ ОБЛАСТІ РИЗИКІВ СТРАТЕГІЧНИХ РІШЕНЬ ОРГАНІЗАЦІЇ

Запропоновано концептуальну модель предметної області ризиків стратегічних рішень у системах організаційного управління (СОУ), яка є результатом операцій вибору, доповнення та інтерпретації, зумовлених специфікою відповідного об'єкту ризику, здійснених щодо елементів нормативного концептуального базису оперування з ризиками в інших сферах діяльності. Модель містить концепти: *Рішення*, *Наслідки для Рішення і для СОУ*, *Джерело*, *Фактор*, *Аспект* і *Подія ризику*, *Вразливість*, *Небезпечний вплив*, *Властивість, що зазнає впливу* та їх взаємозв'язки. На основі запропонованої моделі життєвого циклу стратегічного рішення та узагальнення відомих практик стратегічного управління розглянута система заходів менеджменту ризиків стратегічних рішень, поданих як дії щодо небезпечних впливів, загроз, уразливостей, об'єктів ризику та наслідків. Продемонстровано використання запропонованої концептуальної моделі при розробці методу оцінки ризиків, спричинених неоднозначністю знань, для рішень експертів із представництвом різних ділових груп з використанням моделі «Дерево цінності». Метод ґрунтується на моделі, в якій формалізується множина факторів ризику, викликаного альтернативністю знань. Джерелами факторів є елементи постановки експертної проблеми. Види факторів є видами некогерентного співвідношення знань, вразливості пов'язані з характером дій експертів, небезпечні впливи спричиняються через використовувані системи експертних оцінок, що створюють події ризику в ході процесу життєвого циклу рішення. Кожен з аспектів ризику деталізується властивостями рішення, які є мішенями ризику внаслідок подій ризику. Метод надає оцінки ризику для висновку про задовільність експертизи та можливість вибору найменш ризикованої форми організації її процесу. Розвиток і використання запропонованої концептуальної моделі ризиків стратегічних рішень є перспективним для створення засобів підтримки процесів оборонного планування механізмами онтологічно базованого оперування досвідом щодо проблемних ситуацій та антикризових заходів.

Ключові слова: експертне прийняття рішень, стратегічне рішення, антикризова програма, менеджмент ризиків, концептуальний базис аналізу ризиків, ризик неоднозначності, дерево цінності, оборонне планування в умовах невизначеності.

Постановка та стан проблеми

Менеджмент сучасних систем організаційного управління (СОУ) має справу з множиною викликів, пов'язаних з новими умовами функціонування таких систем [1]. Серед них вирізняються:

- динамічність і значна непрогнозованість впливів зовнішнього середовища;
- ресурсна обмеженість можливостей, і, водночас, ситуативна поява непланованих шансів їх розширення (міжнародні проекти, міжгалузева кооперація тощо);
- множинність центрів відповідальності та компетентності об'єктного й функціонального профілю в структурі СОУ.

Окреме джерело викликів, критичне для створення засобів підтримки діяль-

ності СОУ, складають такі властивості системи знань стосовно стану справ у предметній області (ПрО) цієї діяльності як неповнота, динамічність, розподіленість, слабка формалізованість та неоднозначність [2].

Характерним прикладом діяльності в системі зазначених викликів є сфера державної та міжнародної обороноздатності. Для її організації сучасні умови створюють необхідність планування повсякденної спроможності до відповіді на максимально непередбачувані загрози, зумовивши відповідну побудову процесів управління на всіх рівнях (від стратегічного до тактичного). Для рішень стратегічного управління як факторів невизначеності виступають при цьому множинність потенційних та актуальних конфліктів і

слабка передбачуваність їх майбутньої поведінки. Водночас контексти прийняття рішень і критерії переваг для вибору варіантів дій формуються на основі знань з різних ПрО та інтенцій різних, а часто взаємопротирічних, систем інтересів. До цього додається необхідність інтегрованості окремих рішень до цілісної системи рішень СОУ, яка підлягає процедурам вирівнювання [3] та є формованою побічними впливами рішень у тій же мірі, як і цільовими.

До сучасних тенденцій належить зростання ролі рішень у моделях управління СОУ. Йдеться про концепцію організації, центрованої на рішеннях [1], і введення окремого ракурсу рішень до моделей бізнес-архітектур нарівні з ракурсом бізнес-процесів. Теорія та практика менеджменту декларують як максимальний внесок до втрати ефективності діяльності СОУ дефектів у системі прийняття рішень, насамперед повсякденних (через їх кількість і безпосередню участь у реалізації цілей) і стратегічних (через загрозу спрямування процесу діяльності в цілому).

За умов розглянутих рівня невизначеності та ціни помилки саме проблематика прийняття рішень у СОУ має бути центром розгортання менеджменту ризику [4]. Тому дещо парадоксальним видається факт відсутності розгляду рішення як самостійного об'єкта ризику в основних нормативних і методичних доробках цієї активно зростаючої галузі.

Наприклад, у сфері стратегічного оборонного планування НАТО використовується модель [5], де як об'єкт ризику розглядаються цілі, а процес планування служить потенційним джерелом ризику й розглядається інтегрально.

Водночас, з огляду на використання при цьому в діяльності НАТО моделі робастного прийняття рішень [6] і поєднання в портфелі рішень щодо створення спроможностей, антикризової протидії та превентивних впливів на небезпечні точки зовнішнього середовища [7], безпосереднє запровадження ризику рі-

шень до сфери менеджменту ризиків може надати вагомий внесок до якості процесів планування. При цьому ризики рішень слід розглядати не тільки в зв'язку зі сприянням цілям СОУ, а й з іншими їх важливими властивостями [3], що мають вирішальне значення для процесу та наслідків їх виконання.

Використовуючи визначення ризику як «впливу невизначеності на цілі» [8] та сформувавши концептуальне середовище його конструктивного застосування до стратегічного рішення, можна створити підґрунтя для подальшої побудови інструментарію підтримки різних етапів процесу менеджменту ризику [4] рішень СОУ та онтологічної моделі для ведення й інтелектуального оперування щодо ретроспективи антикризових практик, як це успішно здійснюють в галузі кіберзахисту [9].

Метою роботи є побудова рамкової концептуальної моделі ПрО ризиків стратегічних рішень та її застосування до аналізу найбільш специфічного для такого об'єкта фактора ризику, пов'язаного з неоднозначністю залученого знання [2].

Для аналізу концептуальної бази розгляду ризиків різних типів, які є предметом опрацювання в основних сферах продуктивної діяльності, були використані інформативні джерела, що фіксують відповідний науковий доробок і кращі практики: індустріально апробовані міжнародні, національні й галузеві стандарти де-юре й де-факто і регламенти та публікації авторитетних фахових спільнот [10–29]. Отримані результати подані в табл. 1.

Наведені результати дозволяють стверджувати, що концептуальний базис розгляду ризиків, пов'язаних з різними об'єктами, має спільні закономірності, але й демонструє суттєві особливості, відповідні специфіці оперування ними в розглянутих сферах діяльності.

Отже, наявний досвід концептуалізації надає обґрунтування та передумови для побудови рамкової концептуальної моделі ПрО ризику стратегічних рішень.

Таблиця 1. Показові підходи до менеджменту ризиків продуктивної діяльності

Об'єкт ризику	Трагування ризику	Опис підходу		
		Назва й джерело	Ціль управління	Спеціальні поняття
Сутність, що має цілі	Вплив невідзначеності на цілі [8]	Менеджмент ризику [4]	Створення та захист цінності для причетних сторін	Опис, джерело, подія, власник ризику; небезпечний чинник; уразливість; правдоподібність; наслідок; принцип, структура, процес менеджменту ризику
Організація	Можливість відхилення фактично досягнутих цілей від очікуваних захищеними сторонами	Стандарт Комітету спонсорських організацій комісії Тредвєя (COSO ERM – Integrated Framework) [10]	Забезпечення прийнятної ефективності організації й відстеження її стратегії за допомогою внутрішнього аудиту	Корпоративна система управління ризиками; 8-етапний процес управління; «куб COSO»: цілі (стратегічні, операційні, звітування, відповідності регламентам), етапи процесу, його організаційна структура
		Стандарт Федерації європейських асоціацій ризик-менеджменту (FERMA) [11]	Узгоджене управління ризиками на стратегічному, тактично-оперативному й оперативному рівні	Ті самі, що й у [8]; внутрішній і зовнішній звіт про ризики
		AS/NZS 4360:2004 [12]	Реалізація можливостей при управлінні шкідливими впливами	Сприятлива можливість; контекст (управління ризиками, внутрішній, зовнішній); подія ризику, її імовірність і наслідки; рівень ризику; прийняття ризику для додавання цінності
		Управління ризиками підприємства, кероване цінністю (VDERM), J. Celona [13]	Одночасне узгоджене створення та захист цінності для всіх причетних сторін	Можливість, драйвер цінності, ризиковий апетит, системний ризик, кореляція ризиків, теплова карта ризиків, техніка аналізу рішень
		Керівництво стратегічними ризиками, J.A. Torben [14]	Проактивне прийняття стратегічних ризиків при нейтралізації загроз для додаткової цінності	Стратегічний ризик; невизначеність; прийняття ризику; ризик стратегічного рішення; організація, що приймає стратегічні ризики
Проект	Можливість відхилення стану продуктів проекту від очікуваного	Настанови РМВоК (2017 р.) і практичного стандарту PMI [15]	Стале успішне завершення проектів	Можливість, імовірність події ризику, втрати й прибуток від неї, структура декомпозиції та реєстрризиків, рівень ризику
		Неперервне управління ризиками програмного проекту, С. Alberts [16]	Стале успішне завершення проектів і вдосконалення процесу розроблення	Таксономія ризиків, відповідний їй опитувальник, твердження про ризик, граф ризиків, матрична шкала для рівня ризику
		Методологія випереджального управління загрозами й можливостями (ATOM), D. Hillson [17]	Випереджальне перетворення загроз у можливості, доступне за наявних ресурсів	Можливість, загроза, структура декомпозиції ризиків за джерелом, об'єктом і наслідками, матрична шкала для рівня ризику
		Спрощений аналіз ризиків проекту, L. Virine [18]	Стале успішне завершення проектів	Ланцюг подій ризику, джерело ризику

Експертні та інтелектуальні інформаційні системи

Об'єкт ризику	Трагування ризику	Опис підходу		
		Назва й джерело	Ціль управління	Спеціальні поняття
Інформаційний актив	Можливість спотворення активу і/або невідповідності прав доступу до нього регламентованим	Стандартизоване управління ризиками [9, 19]	Забезпечення рівня інформаційної безпеки, прийнятного для причетних сторін	Загроза, небезпечний вплив, можливість, прояв загрози, вразливість, управлінський вплив, опрацювання ризику
		Аналіз факторів інформаційного ризику (FAIR), J. Freund, J. Jones [20]	Свочасна нейтралізація загроз, доступна на наявних ресурсів	Онтологія ризиків, актив під ризиком, подія загрози, її частота, подія ризику, частота й обсяг втрат від неї, імовірність активізації загрози, вторинний ризик
		Загальна схема кібербезпеки NIST (NCF) [21]	Усунення ризиків із запобіганням або послабленням кібератак	Кібер-атака, загроза, вразливість, актив під ризиком, управлінський вплив, аудит активів і загроз
Інвестовані кошти	Можливість неотримання очікуваного прибутку від інвестиції і/або шкоди для інвестора	Регламенти III Базельського комітету з банківського нагляду (Basel III, 2011) [22]	Забезпечення мінімального регуляторного капіталу й виконання вимог регулятора	Операційний і кредитний ризик; Вартість під ризиком (VaR), середній збиток, ліквідний VaR, показники окупності за невизначеності (RORAC, RAROC, RARORAC) [22]
		Регламенти Європейської агенції зі страхування й фінансових пенсій (Solvency II, 2016) [22]	Забезпечення мінімальної припустимої маржі й виконання вимог регулятора	Те ж саме; страхування й хеджування ризиків; стресове тестування варіантів фінансових операцій
Діловий процес організації	Можливість невідповідності процесу його призначенню в діяльності організації	Стандартизоване визначення оцінок ризику процесу за профілем його зрілості [23]	Обмежена оцінюванням ризику, необхідним для прийняття управлінських рішень	Коренева причина, свідчення й тип ризику (якості продукту, поточний і майбутній організаційний ризик), рівень спроможності процесу
		Стандартизоване управління ризиками в життєвому циклі систем і програмних засобів [24]	Забезпечення рівня ризику життєвого циклу, прийнятного для причетних сторін	Подія ризику, її імовірність і наслідки, профіль ризику проекту, категорія та критерії ризику, джерело, рівень, профіль і стан ризику
Інформаційна система	Можливість відмови системи	Вимірювання й управління ризиком відмови, S.A. Sherer [25]	Забезпечення рівня ризику відмови системи, прийнятного для зацікавлених сторін	Імовірність і збиток від відмови, витрати на забезпечення надійності системи та усунення дефектів після відмов
Інформаційна технологія	Можливість невідповідності технології потребам ділових процесів	Низхідне всебічне оцінювання ризиків, пов'язаних з елементами керування в інформаційній технології (GAIT), E. Hill [26],	Обмежена виявленням критичних функцій технології й управлінських впливів, які перешкоджають підтриманню технологією потреб ділових процесів	Ризик ділового процесу, ключовий управлінський вплив, автоматизований управлінський вплив, функція інформаційної технології, недолік функції або впливу, імовірність і наслідки недоліку
		Цілі керування інформаційними й пов'язаними технологіями (COBIT'2019) [27]	Досягнення цілей керованості (APO12 у [27]) та оптимізації (EDM03) ризиків, пов'язаних з інформа-	Профіль ризику, метрика досягнення цілей керованості й оптимізації ризиків, процес оцінювання управління ризиком

Об'єкт ризику	Трагування ризику	Опис підходу		
		Назва й джерело	Ціль управління	Спеціальні поняття
			ційними технологіями	
Стратегічна ціль в оборонному плануванні	Недосягнення стратегічної цілі	Стратегічне планування, ґрунтоване на ризиках (NATO RBFSP) [5]	Розроблення й стале виконання успішної програми управління ризиками стратегічних планів з передбаченням майбутніх невизначеностей та управлінням їх впливами	Внутрішній і зовнішній контекст планування, політика управління ризиком і комунікації щодо нього, ідентифікація, аналіз, оцінювання та опрацювання ризику, матриця RBFSP (зіставлення етапів управління ризиком за ISO 31000:2009 і діяльностей зі стратегічного планування НАТО
Цільова програма /ключовий процес у критичній галузі	Наявність шкідливих наслідків і сценаріїв, що призводять до їх появи	Прийняття рішень з усвідомленням ризиків (RIDM), D.M. Gerstein [28]	Консолідація різних чинників ризику, наявних у програмах NASA, для вироблення управлінських рішень	Дворівнева класифікація ризиків (чинник ризику, його компонент), граничні умови, поріг, індикатор і стратегія опрацювання для компонента ризику, імовірність і наслідки події ризику
		Управління ризиками, проблемами й можливостями в оборонних програмах, F. Kendall [29]	Забезпечення на всіх етапах оборонної програми рівня ризику, прийнятного для причетних сторін	Небажана подія або умова, її імовірність, наслідки, вплив, тяжкість, проблема, сприятлива можливість, категорія ризику, твердження про ризик

Концептуальна модель менеджменту ризиків стратегічних рішень

Визначення концептуального базису для подання й аналізу ризиків рішень має ґрунтуватися на вимогах повноти врахування:

- складу та співвідношень елементів моделі таких рішень;
- методів і задач, використовуваних кращими практиками їх прийняття;
- методів менеджменту ризику, перспективних для них як для елементів процесів планування в організаційних системах.

В діяльності сучасних організаційних систем важливе місце займає клас рішень, що можуть бути визначені як випереджальні антикризові. Саме вони відіграють роль стратегічних антикризових рішень та максимально відображують парадигму організаційного управління, пов'язану з факторами динамічності, невизначеності та інтегрованості, схарактеризованими в попередньому розділі. Ці рішення складають основу антикризових

програм організацій, що набули особливої уваги в галузі кіберзахисту, а також становлять центральний елемент сучасних підходів до забезпечення обороноздатності держави [30]. Таким чином, побудова концептуального базису, достатнього для розгляду ризиків випереджальних антикризових рішень, може забезпечити достатність отриманого результату для організаційних рішень у цілому.

В табл. 2 надана характеристика моделі процесу життєвого циклу (ЖЦ) випереджальних антикризових рішень, яка відображає його найбільш визначальні риси та розвиває модель, використану в попередніх дослідженнях систем рішень організації [31].

Розгляд поданих характеристик процесу дозволяє дійти таких висновків.

1. Процес характеризується високим ступенем партисипативності, вимагаючи здійснення комунікації представників різних професійних точок зору на об'єкт планованого впливу та інтеграції суджень представників таких точок зору. Це створює ризик так званої неоднозначності (ambiguity) [2], зумовлений різницею в си-

стемах характеристик об'єктів та їх зв'язків, що відповідають досвіду професійної діяльності учасників процесу.

2. Повнота знань і даних стосовно Про рішення включно з повнотою представництва всіх професійних поглядів та інтересів є специфічним джерелом ризику.

3. Багато-вимірність впливів, які справляє рішення на діяльність СОУ, обумовлює багато-вимірність ризику, який має розглядатися в зв'язку з ним.

4. Множинність, різнотиповість і складність структури системи взаємозв'язків між елементами моделі різних етапів процесу, разом із комплексними

впливами цих елементів на ризики, вимагають забезпечення засобів їх відображення в концептуальній моделі ризиків.

5. Високий рівень позатиповості та нестандартності випереджальних антикризових рішень разом із принциповою непередбачуваністю цілого спектру характеристик проблемних ситуацій призводить до потреби формування оцінок ризиків на ґрунті систем гіпотез щодо перебігу подій та з використанням відносного зіставлення ризиків варіантів замість стандартного використання ймовірностей настання й розмірів шкоди подій ризику для абсолютної оцінки ризику варіанта.

Таблиця 2. Принципові характеристики процесу ЖЦ випереджальних антикризових рішень

Етап	Виконувані дії	Учасники й джерела інформації	Результат
1. Аналіз проблемної ситуації	Виявлення й формалізація базового конфлікту Аналіз співіснуючих конфліктів Аналіз загрози впливу на цілі й цінності СОУ Побудова сценаріїв розвитку ситуації	ОПР (за сферами відповідальності) Експерти (за сферами компетентності) Базові сценарії	Модель конфлікту Сценарії граничних варіантів розвитку Оцінки впливу
2. Вибір впливу на проблемну ситуацію	Вибір та обґрунтування ступеню впливу на базовий конфлікт Вибір мішеней впливу Аналіз наявних і доступних засобів	Фахівці: з актуальних загроз, з процесів планування, з забезпечуючої сфери, з зовнішніх зв'язків Носії інтересів, поставлених під загрозу Моделі оцінки загроз	Об'єкти, напрямки та рівні обраних впливів Принципові вимоги до забезпечення Вимоги до очікуваних результатів
3. Вироблення та первинний аналіз варіантів дій	Добір та інтеграція базових заходів для досягнення цілі Виявлення потенційних перешкод в досягненні цілі Побудова сценарного простору з урахуванням варіації факторів перешкод Аналіз стійкості запропонованих варіантів у сценарному просторі [6,7]	Фахівці в цільовій сфері «Червоні команди» [6,7] експертів Інформація щодо стану справ, засоби сценарного аналізу та критерії відсіву варіантів	Множина альтернативних варіантів з їх обґрунтуванням
4. Вибір варіанта	Формування та узгодження складу й структури моделі вибору Формування моделі та складу експертних груп Експертне оцінювання Інтеграція та узгодження експертних оцінок	Фахівці з обраних заходів Експерти з базових аспектів оцінки варіанта Стейкхолдери Рамкові моделі цінності	Обґрунтований варіант вибору
5. Підготовка до ви-	Планування заходів Розподіл відповідальностей і повноважень	Відповідні служби СОУ Діючі регламенти Наявні канали	План заходів Інструкції з виконання плану

Етап	Виконувані дії	Учасники й джерела інформації	Результат
конання	Розробка й планування системи допоміжних заходів Висунення вимог до документації, базових комунікацій і каналів інформування		заходів Склад і правила документообігу Інструкції з комунікацій
6. Аналіз з результатів	Перевірка відповідності вимогам етапу 2 Аналіз виявленого рівня задовільності виконаного АВР для оперативної антикризової діяльності Анкетування учасників щодо вузьких місць процесів діяльності та пропозицій Корекція та актуалізація елементів моделі процесу	ОПР Аналітики з антикризового управління Учасники ділових процесів Стейкхолдери	Оцінка виконання Актуалізовані результати попередніх етапів Висновки стосовно змін в умовах і моделях випереджального антикризового управління.

Як концептуальний базис для розгляду організаційних рішень, що враховує зазначені особливості, може бути взята поняттєва основа стандарту [8]. Введення додаткових концептів і взаємозв'язків, а також інтерпретація щодо ПрО стратегічних рішень здійснювалися з використанням підходів [5, 9]. Пропонована система концептів має структуру, показану на рисунку.

Концепти позначено як *E* з номерами, в той час як дуги *R* з номерами ідентифікують зв'язки між ними з такою семантикою:

- R1* – «...має елемент, що виступає в якості...»;
- R2* – «...породжує...»;
- R3* – «...характеризує...»;
- R4* – «...належить до ... як елемент деталізації »;
- R5* – «...змінює стан...»;
- R6* – «...включає ... до сфери впливу...», де знак ... позначає концепт у початковій або кінцевій вершині дуги *R*.

Об'єкт ризику *E3* – це організаційне рішення, що розглядається у вигляді п'яти-етапної структури його ЖЦ, охарактеризованої в табл. 1, та пов'язаних з

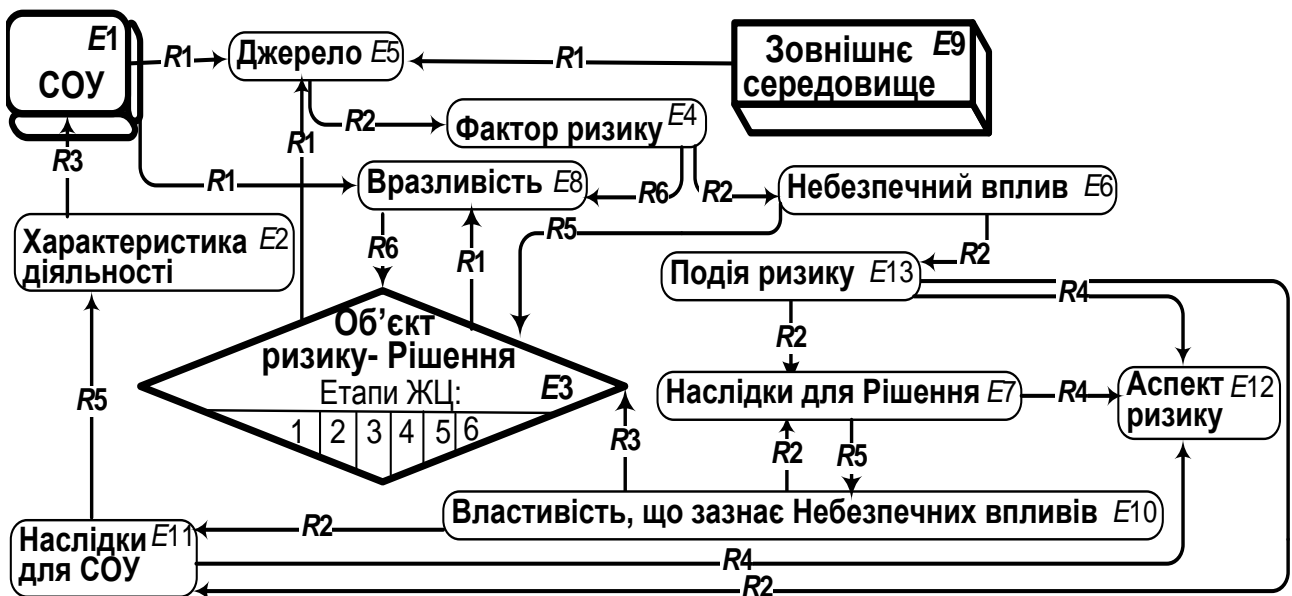


Рисунок. Структура концептуальної моделі ПрО ризиків стратегічних рішень

ним властивостей, що визначають його процесну якість і положення в системі рішень організації [32]. *Властивості E10* становлять підмножину Властивостей рішення, зміна значень яких з наближенням до виділених критичних значень або з виходом за межі виділеної області прийнятності надає первинну характеристизацію ризиків.

Безпосередніми маркерами ризиків слугують *Наслідки E7* та *E11* *Подій ризику E13*, рівень тяжкості (а також прийнятності та істотності) яких може бути оціненим на основі значень Властивостей *E10* і Характеристик діяльності СОУ *E2*. У виродженому випадку *E7* може співпадати з *E10*, а *E11* – з *E2*.

Аспекти ризику E12 становлять базові функціональні та ціннісні виміри, за якими розглядається організаційне рішення як центральний елемент СОУ. Кожний з таких аспектів характеризується актуальними для нього елементами *E7* та *E11* як своїми складовими, а також подіями ризику *E13*, що спричиняють зміни їх значень.

Фактор ризику E4 становить властивість, притаманну об'єкту, який служить його джерелом *E5* і належить до Зовнішнього середовища *E9* (зовнішній фактор) або до елементів СОУ *E1*, безпосередньо пов'язаних із процесом ЖЦ рішення *E3*, чи до його характеристик, виступаючи в ролі внутрішнього фактору. Фактор характеризується значенням рівня його прояву, що визначає активізацію пов'язаного з ним *Небезпечного впливу E6* при перевищенні критичного рівня. Характеризується також множиною *Вразливостей E8*, які уможливають та посилюють його вплив.

Небезпечний вплив *E6* характеризується, крім породжуючого його фактору, множиною елементів рішення *E3*, на стан яких може впливати (зокрема, при наявності відповідних Уразливостей *E8*) і *Подіями ризику E13*, які змінюють стан Об'єкту ризику. Якщо *Небезпечний вплив* може бути пов'язаний із суб'єктами Зовнішнього середовища або СОУ, інтереси яких виступають у ролі Джерела *E5* для відповідного Фактора ризику *E4*, він має

тип *Загрози* та може додатково характеризуватися стадією свого розвитку. Обов'язковою характеристикою є *Рівень впливу*, що є суттєвим при оцінці Наслідків *E7* та *E11*.

Подія ризику *E13* визначається як зміна стану Об'єкту ризику *E3*, що спричиняє Наслідки *E7* та *E11*. Рівень її значущості залежить від їх рівнів тяжкості. Подія ризику належить до сфери певного Аспекту ризику *E12*.

Крім розглянутих концептів, для побудови моделей менеджменту ризиків рішень зручно ввести комплексні концепти, що характеризують поточну ситуацію, яка має місце в ЖЦ окремого рішення.

Перший з них являє собою *Ситуацію небезпеки з моделлю*

$$SH = \{ \langle E4_i^*, E5_i \rangle, SE8_i, \{ E6_{ij}^{**}, j=1, \dots, N_i \}, i=1, \dots, M \}, \quad (1)$$

а другий – Ситуацію ризику з моделлю

$$SR = \{ E6_{ij}^{**}, SE10_{ij}, SE13_{ij}^{***}, \{ E12_{ijk}^{***}, SE7_{ijk}^{***}, SE11_{ijk}^{***}, k=1, \dots, K_{ij} \}, j=1, \dots, N_i, i=1, \dots, M \}, \quad (2)$$

де $E4_i^*$ – i -й актуалізований фактор ризику з джерелом $E5_i$;

$SE8_i$ – множина наявних уразливостей, що сприяють впливу фактора $E4_i^*$;

$E6_{ij}^{**}$ – j -й небезпечний вплив, створюваний фактором $E4_i^*$;

N_i і M – кількості створюваних впливів та актуалізованих факторів;

$SE10_{ij}$ – множина властивостей рішення з критичними значеннями;

$SE13_{ij}^{***}$ – множина ініційованих подій ризику;

$E12_{ijk}^{***}$ – k -й актуальний аспект ризику;

$SE7_{ijk}^{***}$, $SE11_{ijk}^{***}$ – множини актуальних наслідків, що його репрезентують.

Далі продемонстровано приклади елементів, що здійснюють репрезентацію концептів $E4$, $E7$, $E12$.

В табл. 3 наведено основні аспекти ризику організаційних рішень і наслідки небезпечних впливів, пов'язані з ними, –

представлені у вигляді властивостей рішення, що складають відповідні мішені для таких впливів.

Таблиця 3. Мішені дії небезпечних впливів у складі аспектів ризику

Аспект ризику	Мішень дії небезпечного впливу
Результативність процесу вироблення (AR1)	Завершеність процесу (TR1.1) Збереження області застосування постановки (TR1.2) Якість (TR1.3) Своєчасність ухвалення (TR1.4) Вписуваність в систему процесу управління (TR1.5) Повторна використовуваність (TR1.6)
Здійсненість (AR2)	Забезпеченість ресурсом (TR2.1) Ефективність роботи команди виконавців (TR2.2) Задоволення нормативно-правових вимог (TR2.3) Своєчасність виконання (TR2.4) Здатність до адаптації щодо умов виконання (TR2.5)
Ефективність (AR3)	Прийнятність співвідношення вигоди/витрати (TR3.1) Конкурентоспроможність забезпеченого ефекту (як послуги або продукції) (TR3.2) Скомпенсованість негативних впливів (TR3.3)
Прийнятність для стейкхолдерів (AR4)	Відповідність досягнутої вигоди очікуванням (TR4.1) Збереження балансу між інтересами різних стейкхолдерів (TR4.2) Забезпечення довіря стейкхолдерів до дотримання їх інтересів (TR4.3)

В табл. 4 наведена рамкова структура груп факторів ризику з джерелами зі складу зовнішнього середовища, СОУ та системи знань, використовуваних на етапах процесу ЖЦ рішення.

Схарактеризовані основні фактори ризику, які створюють небезпеки для властивостей організаційного рішення, що є їх мішенями в аспектах AR1–AR4. Запропонована рамкова ієрархічна структура цих чинників орієнтована на можливість оцінки й моделювання подій і ризиків у зв'язку з індикаторами стану середовища та на вироблення й аналіз заходів менеджменту ризику.

Таблиця 4. Рамкова структура груп факторів ризику

Група факторів	Структура та склад групи
Фактори з джерелом у зовнішньому середовищі	Невизначеність проявів F1.i.1 Мінливість значень F1.i.1.1 Непоінформованість F1.i.1.2 Нестабільність впливів F1.i.2 Залежність від інших зовнішніх чинників F1.i.2.1 Залежність від внутрішніх параметрів організації F1.i.2.2 Залежність від стану об'єкту ризику F1.i.2.3 Керованість F1.i.3 Регістрованість F1.i.3.1 Компенсованість F1.i.3.2
Фактори з джерелом у внутрішній структурі й діяльності СОУ	Стан системи цілей F2.1 Невизначеність F2.1.1 Несинхронізованість F2.1.2 Ресурсна несумісність F2.1.3 Неінтегрованість F2.1.4 Ресурсний потенціал F2.2 Недостатність F2.2.1 Недоступність F2.2.2 Невигідність F2.2.3 Структура F2.3 Функціональна неадекватність F2.3.1 Неадаптивність F2.3.2 Надмірність і дублюваність F2.3.3 Технології F2.4 Механізми управління F2.5 Нерозподіленість і нефіксованість відповідальності F2.5.2 Некоординованість F2.5.3 Неінтегрованість цілей процесів F2.5.1 Неконтрольованість якості F2.5.4 Непідтриманість взаємодії F2.5.5 Неефективність зворотних зв'язків F2.5.6
Фактори з джерелом у системі використуваного знання	Неформалізованість F3.1 Неалгоритмізованість F3.1.1 Невизначеність експертних ресурсів F3.1.2 Несистематизованість досвіду F3.1.3 Неповнота F3.2 Концептуальна F3.2.1 Інформаційна F3.2.2 В частині методологій застосування F3.2.3

В табл. 5 розглянуто можливості здійснення функцій менеджменту ризиків за рахунок заходів із впливу на об'єкти запропонованої концептуальної моделі.

Дії з менеджменту ризиків рішень СОУ, розглянуті в табл. 5, та способи їх реалізації гармонійно сполучені з моделлю Стратегічного планування, ґрунтованого на ризиках (SPRBF) [5]. Детальний розгляд наведених в табл. 5 способів реалізації необхідних впливів та виміру «Техніки» моделі SPRBF дозволяє виокремити множину технік, актуальних для таких впливів. Вона включає класи експертних методів, системно-аналітич-

них технік, схем планування та організаційного аналізу. Всі вони потребують, для свого використання в конкретній предметній області, аналізу тих зв'язків та властивостей, які використані в розглянутій концептуальній моделі та можуть поповнюватися надалі, при збереженні її когерентності.

В наступному розділі наведено приклад використання запропонованої концептуальної моделі для задач менеджменту ризиків, спричинених фактором неоднозначності використовуваних знань [2].

Таблиця 5. Цілі та заходи здійснення менеджменту ризиків рішень СОУ в середовищі концептуальної моделі

Тип об'єкту впливу	Здійснюваний цільовий вплив на об'єкт	Спосіб реалізації
Небезпечний вплив (загроза)	Безпосередня протидія	Перспективне створення засобів Використання штатних засобів Ситуативний пошук підходів у ретроспективі процесів прийняття рішень
	Зниження стадії розвитку загрози	Моніторинг з метою ранньої діагностики Випереджуючі дії Досягнення компромісу конфліктних інтересів Зміна балансу сил (залученням ресурсів, створенням кооперацій та ін.
	Послаблення чи нейтралізація джерела фактору ризику	Випереджувальний вплив на зовнішнє джерело за рахунок участі в процесах зовнішнього середовища Вплив на внутрішні процеси СОУ, актуальні для об'єкту – джерела фактору
Вразливість	Захист об'єкту – носія вразливості від актуальних негативних впливів	Використання штатних засобів Ситуативний пошук можливостей
	Заміщення об'єкту – носія в складі ділових процесів, які визначають критичні властивості рішення	Створення альтернативних об'єктів та маршрутів Використання штатних альтернатив
	Зміна сценаріїв та маршрутів функціонування об'єкту носія в ділових процесах виконання рішення	Створення сприятливих умов (ресурси, партнерська участь, додаткове інформування) Використання рамкових альтернатив Ситуативне використання прецедентів
Рішення (як об'єкт ризику)	Забезпечення повноти залучення знань у процесі прийняття рішення	Побудова ефективного розподіленого процесу (розподіл відповідальностей, регламентація взаємного інформування) Використання ретроспективної інформації та пошук аналогів Побудова та актуалізація сценарного простору, що визначає умови та

Тип об'єкту впливу	Здійснюваний цільовий вплив на об'єкт	Спосіб реалізації
		небезпечні впливи при виконанні рішення Створення профільних експертних команд (поточного аналізу, багатоаспектної інтеграції, пошуку вузьких місць і перешкод та ін.)
	Забезпечення готовності до змін	Створення універсального та взаємосумісного набору можливостей Розробка рамкових антикризових програм Створення моделей багатокритеріального та робастного до потенційних перешкод вибору варіантів дій Реалізація ітеративних та рефлексивних схем процесів прийняття рішень Ведення, актуалізація та наскрізне використання системи пріоритетів Аргументоване виведення висновків із отриманих результатів та їх конструктивне збереження
	Підтримка інтегрованості рішення	Ведення моделі та аналіз стану системи рішень СОУ Створення інтегральних контекстів для експертиз Реалізація багатокритеріальних та портфельних підходів
Наслідки ризику	Відвернення	Передбачення та моніторинг побічних наслідків рішень Вибір варіантів дії за принципом робастності до змін середовища Ресурсне забезпечення рішень, адекватне пріоритетності об'єктів їх впливу
	Протидія	Розробка і реалізація рамкових заходів Ситуативне використання можливостей СОУ Використання досвіду прецедентів Залучення зовнішніх можливостей
	Післякризова стабілізація	Планування та використання заходів та ресурсів для відновлення діяльності об'єктів та системи в цілому Модифікація моделей та сценаріїв діяльності на ґрунті винесених висновків

Оцінка ризиків, спричинених неоднозначністю, для рішень експертиз із представництвом різних ділових груп

Для рішень, розглянутих у перших розділах, характерна риса – це потреба в використанні розподілених та частково альтернативних знань відносно проблемної ситуації, а також властивостей та наслідків заходів з її розв'язання.

Запропонована раніше [32] методологія комплексної експертно-аналітичної підтримки формування та ведення систем рішень відповідних СОУ виводить на перший план у задачах менеджменту ризику підклас внутрішніх факторів альтернативності знань, джерелом яких є елементи розв'язуваної експертної проблеми. При цьому види факторів є видами некогерентного співвідношення знань, вразливості пов'язані з характером дій експертів, небезпечні впливи спричиняються через вико-

ристовувані системи експертних оцінок, що створюють події ризику в ході процесу життєвого циклу рішення. Кожен з аспектів ризику деталізується властивостями рішення, які є мішенями ризику внаслідок подій ризику.

Запропонований метод оцінки відповідних ризиків, орієнтованої на такі проблеми їх менеджменту як моніторинг динаміки ризиків у системі рішень СОУ та вибір сприятливих методів реалізації процесу їх прийняття, демонструє можливості використання концептуальної моделі, описаної у попередньому розділі.

Будемо розглядати модель задачі прийняття рішення TS

$$TS = \langle PS, G, \{A\}, P, ST, NI, MP \rangle, \quad (3)$$

де PS – проблемна ситуація, G – ціль дії щодо неї, A – альтернативний захід для реалізації G , P – доступний потенціал виконання, ST – поточний стан системи рішень організації [32], NI – можливі негативні

наслідки, MP – модель переваг [33] для оцінки перспективності альтернатив.

Кожний елемент E з (3) є структурою концептів C , що характеризуються своїми онтологічними визначеннями

$$Def(C) = \{X, R\}, \quad (4)$$

де X – концепти, які визначають склад і властивості C ; R – відношення, що пов'язують C і X .

Множина актуальних для TS точок зору $\{VP\}$ відповідає діловим групам і стейкхолдерам організації і характеризується онтологічно множиною $O(VP)$ своїх версій Def . Багатокритеріальна ієрархічна модель [33] MP із (3) включає, на першому рівні, систему ракурсів, що розкривають показник переваги варіанта рішення

$$AMP = \langle R, AC, F, AP \rangle, \quad (5)$$

де R – результативність для вирішення PS , AC – досяжність цілі G , F – здійсненність заходу, AP – прийнятність і можливість усунення наслідків.

Конструктивним для розв'язання задачі (3) є застосування, як формального подання MP з (3), додатково розвинутого апарату моделі Дерево цінності [33]. Він включає: механізми онтологічної аргументованості взаємозв'язків критеріїв, продукційні правила виведення рекомендацій з управління та спеціальні методи інтеграції і узагальнення оцінок [33].

Залучена до експертизи VP реалізує відображення свого фахового досвіду в оцінках, які надає для варіанта рішення, за допомогою оцінювання системи часткових критеріїв, що деталізують аспекти з (5):

$$\begin{aligned} CR(VP, R) &= \psi_{VP}(G, PS), CR(VP, AC) = \\ &= \psi_{VP}(G, A, ST), \\ CR(VP, F) &= \psi_{VP}(A, P), CR(VP, AP) = \\ &= \psi_{VP}(A, ST, NI), \end{aligned} \quad (6)$$

де CR – множина критеріїв, що представляють аспект; ψ – функція формування складу множини критеріїв та їх оцінок на підставі фахових знань про елементи (3).

Це визначає альтернативність експертного знання про елементи (5) при залученні множини VP до прийняття рішення (3).

Використання в постановці задачі для експертів єдиної моделі MP є поширеною практикою. Застосування вище охарактеризованого подання MP служить при цьому: забезпеченню підстав для інтеграції оцінок різних експертів; використанню проміжних рівнів ієрархії критеріїв для формування рекомендацій і класифікації альтернатив [33]; аналітичному використанню оцінок за межами даної експертизи і повторному використанню постановки задачі при оцінюванні рішень даного класу.

Проте відмінності між складом MP в (3) і результатами (6) призводять до можливості однієї з ризикованих форм W поведінки представників VP в експертизі:

- відмови від оцінювання критерію (W_1);
- оцінювання наданих критеріїв з викривленням професійних уявлень (W_2);
- оцінювання часткових критеріїв з «підгонкою» під оцінку інтегрального, яка відображає професійні пріоритети (в реальності зумовлені іншими аргументами) (W_3).

Такі форми поведінки експертів складають вразливості системи експертних оцінок, які роблять її чутливою до відмінностей інтерпретації елементів постановки задачі різними VP .

Розглянемо модель ризиків рішення, що приймається за таких умов. Вона має вигляд

$$MR = (\{AR_i, \{TR_{ij}, \{TRR_{ijk}(T, FC, SR)\}_{k=1, K_j}\}_{j=1, J_i}\}_{i=1, \dots, 4}, TT), \quad (7)$$

де AR – аспект ризику рішення (див. табл. 3); TR – мішень ризику в межах аспекту (властивість рішення, порушення якої є одним із підаспектів ризику); TRR – небезпечний вплив, що викликає подію ризику, має тип T і продукується фактором ризику FC з джерелом SR (елемент (5)); J_i – число мішеней ризику в аспекті; K_j – число загроз, актуальних для мішені; TT – множина типів небезпечних впливів. Множина TT включає такі дефекти системи індивідуа-

льних експертних оцінок: T_1 – неповноту поданості в оцінці аспектів з (5); T_2 – невідповідність оцінки інтегрального критерію переваг фаховому досвіду; T_3 – невідповідність оцінки часткових критеріїв фаховому досвіду; T_4 – незіставність оцінок для процедур інтеграції й логічного виведення рекомендацій; T_5 – непереконливість для інших точок зору.

Ситуація небезпеки має модель

$$MSR = \langle TSR, CB, \{CST(V_i, V_j), FC, W(V_j)\} \rangle, \quad (8)$$

де TSR – тип джерела фактора ризику (R, AC, F, AP з (3)); CB – концептуальний базис небезпечного впливу (підмножина елементів з правої частини виразу (6) для TSR); $CST(V_i, V_j)$ – конфліктний елемент: концепт C , для якого

$$C \in E \mid (E \in CB) \wedge Def(C, V_i) \neq Def(C, V_j);$$

фактор ризику FC – одна з п'яти форм альтернативного співвідношення визначень C ; $W(V_j)$ – форма поведінки представника точки зору V_j в ситуації при оцінюванні.

$$FC_1. \quad \exists X \mid X \in (O(VP_i), O(VP_j)) \wedge \\ \wedge X \in Def(C, VP_i) \wedge X \notin Def(C, VP_j);$$

$$FC_2. \quad \exists X \mid X \in O(VP_i) \wedge X \notin O(VP_j) \\ \wedge X \in Def(C, VP_i) \wedge \neg \underline{UND}(X, VP_i, VP_j),$$

де \underline{UND} – відношення розуміння [32];

$$FC_3. \quad \exists X \mid (X \in O(VP_i) \wedge X \notin (O(VP_j)) \wedge \\ \wedge X \in Def(C, VP_i) \wedge \underline{UND}(X, VP_i, VP_j) \wedge \neg \exists Z \mid \\ (Z \in VP_j) \wedge \underline{CONTR}(TR(X, VP_i, VP_j), Z), \quad (9)$$

де \underline{CONTR} – відношення суперечності [32], $TR(X, VP_i, VP_j)$ – образ X в $O(VP_j)$;

$$FC_4. \quad \exists X \mid (X \in O(VP_i) \wedge X \notin (O(VP_j)) \wedge \\ \wedge X \in Def(C, VP_i) \wedge \underline{UND}(X, VP_i, VP_j) \wedge \exists Z \mid \\ (Z \in VP_i) \wedge \underline{CONTR}((TR(TR(X, VP_i, VP_j), \\ VP_j, VP_i)), Z);$$

$$FC_5. \quad \exists X_1 \mid (X_1 \in O(VP_i), O(VP_j)) \wedge X_1 \in \\ \in Def(C, VP_i) \wedge \exists X_2 \mid (X_2 \in O(VP_i), (O(VP_j)) \wedge$$

$$\wedge X_2 \in Def(C, VP_j)) \wedge \underline{CONTR}(X_1, X_2).$$

Кожен тип небезпечних впливів T визначено диз'юнкцією варіантів (FC, W).

$$T_1. \quad (FC_1, W_1) \vee (FC_2, W_3) \vee (FC_5, W_1);$$

$$T_2. \quad (FC_1, W_2) \vee (FC_2, W_2) \vee (FC_4, W_2) \vee (FC_4, W_3) \vee \\ \vee (FC_5, W_2) \vee (FC_5, W_3); \quad (10)$$

$$T_3. \quad (FC_1, W_3) \vee (FC_2, W_3) \vee (FC_4, W_2) \vee (FC_4, W_3) \vee \\ \vee (FC_5, W_2) \vee (FC_5, W_3) \vee (FC_3, W_2) \vee (FC_3, W_3);$$

$$T_4. \quad (FC_4, W_2) \vee (FC_4, W_3) \vee (FC_5, W_2) \vee (FC_5, W_3);$$

$$T_5. \quad (FC_2, W_2) \vee (FC_2, W_3) \vee (FC_4, W_2) \vee (FC_4, W_3) \vee \\ \vee (FC_5, W_2) \vee (FC_5, W_3).$$

Грунтуючись на моделі ризику (7), можна запропонувати оцінку його рівня для рішення, що приймається в постановці задачі (3), із залученням представників заданої множини $\{VP\}$, для яких є виявленою – апріорно або в рамках експертизи, онтологічна специфікація концептів постановки задачі.

Для кожного з 20 елементів множини $GTRR$ можливих небезпечних впливів $TRR_{ij}(T_i, SR_j)$ оцінюється значущість EMT_{ij} з урахуванням всіх актуальних для нього конфліктних елементів CST_{ij} (див. (8)).

Нехай NVP – загальне число залучених точок зору, NT – число їх різних пар.

Значущість конфліктного елемента $CST_{ijr}(V_{L1}, V_{L2})$, актуального для небезпечних впливів $TRR_{ij}(T_i, SR_j)$ у зв'язку з r -ю парою точок зору (V_{L1}, V_{L2}) , $r \leq NT$, має вигляд

$$EMC_{ijr} = H \sum_{w \in W_a} P(w) / |W_a|, \quad (11)$$

де $P(w)$ – імовірність реакції у формі $w \in W_a$ на ситуацію, утворювану конфліктним елементом, з боку точки зору, що включає більш інформативне визначення CST ; $H \in (0, 1)$ – усереднений для V_{L1}, V_{L2} рівень актуальності знань про концепт CST ; W_a – множина форм реакції, актуальних для типу небезпечного впливу T_i (див. (10)).

Тоді значущість небезпечного впливу TRR_{ij} можна визначити як

$$E(TRR_{ij}) = \sum_{r=1}^{NT} \sum_{j=1}^{Nir} EMC_{ijr} / K_j \cdot NT,$$

де K_j – кількість концептів у визначенні SR_j з (6); Nir – число CST_{ijr} , актуальних для TRR_{ij} ; NT – число пар точок зору.

Позначимо $ACT_S \in GTRR$ підмножину небезпечних впливів TRR_{ij} , актуальних для мішеней аспекту ризику AR_S (див. (7)). Тоді оцінку рівня ризику за цим аспектом, $RL_S \in (0,1)$, може визначити як

$$RL_S = \sum_{TRR_{ij} \in ACT_S} E(TRR_{ij}) / |ACT_S|.$$

На основі запропонованих оцінок ризику може визначатися рейтинг елементів системи рішень СОУ та задовільність окремого рішення в порівнянні з результатами кращих і гірших практик.

Число форм співвідношення (9) може бути розширено такою формою як вилученість концепту з визначень в одній або обох точках зору, з урахуванням рівня актуальності знань про нього (див. (11)). Пов'язавши цю форму з типами небезпечних впливів в рамках умов (10), можна використати запропонований підхід для вибору процедур управління ризиками рішень, пов'язаних з альтернативністю знань (вилучення суперечностей з постановки задачі, включення компромісної версії моделі цінності, звуження кола точок зору, що залучаються тощо).

Висновки

На ґрунті аналізу світового досвіду побудови концептуального базису менеджмента ризику в різних сферах діяльності була створена рамкова концептуальна модель предметної області ризиків для стратегічних рішень, які характеризуються запропонованою моделлю життєвого циклу, в сучасних системах організаційного управління.

Розглянуто основні аспекти ризику та властивості процесу життєвого циклу рішення – мішені дії небезпечних впливів

у складі кожного з аспектів. Схарактеризовано основні групи факторів ризику.

Запропоновано подання цілей і заходів менеджменту ризиків як цільових впливів, що здійснюються для об'єкту концептуальної моделі, за допомогою засобів, характерних для сучасного антикризового управління та стратегічного планування за умов високого ступеню невизначеності (з залученням моделей робастного оборонного планування).

Показано використання запропонованої концептуальної моделі для аналізу й оцінки ризиків рішення, спричинених дією фактора неоднозначності розподілених знань при використанні прийнятої для стратегічних рішень експертної процедури оцінювання за Деревом цінності.

Планується подальше використання та розвиток результатів для створення моделей і засобів онтологічної підтримки ведення й використання знань стосовно проблемних ситуацій і добору антикризових заходів, базованому на ризиках рішень оборонного планування.

Література

1. Blenko M.W., Mankins M., Rogers P. The Decision-Driven Organization. Harvard Business Review. 2010. Is. 6. [Electronic resource]. Mode of access: <https://hbr.org/2010/06/the-decision-driven-organization>.
2. Renn O. Coping with complexity, uncertainty and ambiguity. The risk governance approach. NSF-DFG Joint Risk Meeting, Washington, D.C., Oct. 3-5, 2012. 33 p.
3. Pisano G.P. Creating an R&D Strategy. 2012. [Electronic resource]. Mode of access: http://www.hbs.edu/facultyPublication%20File/s/12-095_fb1bdf97-e0ec-4a82-b7c0-42279dd4d00e.pdf.
4. ISO 31000:2018 Risk management – Guidelines. 16 p.
5. Analysis Support Guide for Risk-Based Strategic Planning. Technical Report STO-TR-SAS-093-Part-I – 2018, STO/NATO. 156 p.
6. Lempert R.J., Warren D., Henry R. et al. Defense Resource Planning Under Uncertainty. An Application of Robust Decision Making to

- Munitions Mix Planning. RAND Corporation, 2016. 109 p.
7. Johnson S., Libicki M., Treverton G.F. New challenges, new tools for defense decisionmaking. MR-1576, RAND Corporation, 2003. 408 p.
 8. ISO Guide 73:2009 Risk management Vocabulary. 15p.
 9. ISO 27000:2018 Information technology – Security techniques. Information security management systems. Overview and vocabulary. 34 p.
 10. Enterprise Risk Management – Integrating with Strategy and Performance. Executive Summary. COSO, 2017. 16 p. [Electronic resource]. Mode of access: <https://www.coso.org/Documents/2017-COSO-ERM-Integrating-with-Strategy-and-Performance-Executive-Summary.pdf>.
 11. FERMA Risk Management standard – FERMA, 2002. 16 p. [Electronic resource]. Mode of access: // <http://www.ferma.eu/>.
 12. Australia/New Zealand AS/NZS 4360:2004 Risk management /Standards Australia. 65 p.
 13. Celona J., Hall E., Driver J. Value-Driven ERM: Making ERM an Engine for Simultaneous Value Creation and Value Protection. *J. of healthcare risk management: the journal of the American Society for Healthcare Risk Management*. 2011. N 30(4). P. 15–33.
 14. Torben J.A., Garvey M., Roggi O. Managing Risk and Opportunity. The Governance of Strategic Risk-Taking. Oxford University Press, 2014. 204 p.
 15. Practice Standard for Project Risk Management. *Project Mngement Institute, Inc.*, 2009. 128 p.
 16. Alberts C., Dorofee A., Marino L. Executive Overview of SEI MOSAIC: Managing for Success Using a Risk-Based Approach. Technical Note CMU/SEI-2007-TN-008, 2007. 33 p.
 17. Hillson D.A., Simon P.W. Practical project risk management: The ATOM Methodology (2nd ed.). Vienna, US: Management Concepts, 2012. 410 p.
 18. Virine L., Trumper M. Project Risk Analysis Made Ridiculously Simple. World Scientific Publishing Co. Pte. Ltd., 2017. 283 p.
 19. ISO 27005:2018 Information technology – Security techniques – Information security risk management. 56 p. [Electronic resource]. Mode of access: <https://www.iso.org/standard/75281.html>.
 20. Freund J., Jones J. Measuring and Managing Information Risk. A FAIR Approach. Elsevier, 2015. 391 p.
 21. Офіційний сайт NIST CyberSecurity Framework. [Electronic resource]. Mode of access: <https://www.nist.gov/cyberframework>.
 22. Grouhy M., Galai D., Mark R. The Essentials of Risk Management. McGraw-Hill Education, 2014. 669 p.
 23. ISO/IEC PDTR 33015.3:2019 Information technology – Process assessment – Guide to process risk determination. 41 p.
 24. ISO/IEC 16085:2006 Systems and software engineering – Life cycle processes – Risk management. 34 p.
 25. Sherer S.A., Alter S. Information Systems Risks and Risks Factors, are they Mostly about Information Systems? *Communications of AIS 2004*. Vol. 14. N 1. P. 29–64.
 26. GAIT for IT General Control Deficiency Assessment. The Institute of Internal Auditors. [Electronic resource]. Mode of access: https://www.iiacolombia.com/resource/guias/GAIT_GeneralControl.pdf. 25 p.
 27. COBIT'2019 Framework. Governance and Management Objectives – ISACA, 2018. 302 p.
 28. Gerstein D.M. et al. Developing a Risk Assessment Methodology for the National Aeronautic and Space Administration. RAND Corporation, 2016. 113 p.
 29. Kendall F. Department of Defense Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs. Washington, DC 20301-3030, 2017. 96 p. [Electronic resource]. Mode of access: <https://www.acq.osd.mil/se/docs/2017-rio.pdf>.
 30. NATO Standard AJP-5. Allied Joint Doctrine for the Planning of Operations. Ed. A V.2. NATO Standardization Office, 2019. 134 p.
 31. Ильина Е.П., Сеницын И.П. Модели и методы поддержки аналитического сопровождения поля решений организации. *Проблеми програмування*. 2017. № 3. С. 93–107.
 32. Ильина Е.П. Методы и модели использования экспертно-аналитического знания для поддержки принятия решений в организации. Часть 1. *Модели знания о решениях. Проблеми програмування*. 2016. № 1. С. 89–101.
 33. Ильина Е.П., Слабоспицкая О.А., Сеницын И.П., Яблокова Т.Л. Автоматизированная поддержка принятия решений по управлению программами фундаментальных научных исследований с использованием экспертной методологии. Киев / Препринт. 2011. 94 с.

References

1. Blenko M.W. The Decision-Driven Organization – M.W.Blenko, M.Mankins, P.Rogers / Harvard Business Review. – 2010. – Is. 6. [Electronic resource]. Mode of access: <https://hbr.org/2010/06/the-decision-driven-organization>.
2. Renn O. Coping with complexity, uncertainty and ambiguity. The risk governance approach / O.Renn – NSF-DFG Joint Risk Meeting, Washington, D.C., Oct. 3-5, 2012. – 33 p.
3. Pisano G.P. Creating an R&D Strategy / G.P. Pisano – 2012. [Electronic resource]. Mode of access: http://www.hbs.edu/facultyPublication%20Files//12-095_fb1bdf97-e0ec-4a82-b7c0-42279dd4d00e.pdf.
4. ISO 31000:2018 Risk management – Guidelines. – 16p.
5. Analysis Support Guide for Risk-Based Strategic Planning. / Technical Report STO-TR-SAS-093-Part-I – 2018, STO/NATO. – 156 p.
6. Lempert R.J. Defense Resource Planning Under Uncertainty. An Application of Robust Decision Making to Munitions Mix Planning / R.J. Lempert, D.Warren, R.Henry et al. – RAND Corporation, 2016. – 109 p.
7. Johnson S. New challenges, new tools for defense decisionmaking / S.Johnson, M.Libicki, G.F. Treverton – MR-1576, RAND Corporation, 2003. – 408 p.
8. ISO Guide 73:2009 Risk management Vocabulary. – 15p.
9. ISO 27000:2018 Information technology – Security techniques – Information security management systems — Overview and vocabulary – 34 p.
10. Enterprise Risk Management – Integrating with Strategy and Performance. Executive Summary. – COSO, 2017. – 16 p. [Electronic resource]. Mode of access: <https://www.coso.org/Documents/2017-COSO-ERM-Integrating-with-Strategy-and-Performance-Executive-Summary.pdf>.
11. FERMA Risk Management standard – FERMA, 2002. – 16 p. [Electronic resource]. Mode of access: // <http://www.ferma.eu/>.
12. Australia/New Zealand AS/NZS 4360:2004 Risk management /Standards Australia.– 65 p.
13. Celona J. Value-Driven ERM: Making ERM an Engine for Simultaneous Value Creation and Value Protection / J.Celona, E.Hall, J.Driver // J. of healthcare risk management: the journal of the American Society for Healthcare Risk Management – 2011 – N 30(4) – P.15-33.
14. Torben J.A. Managing Risk and Opportunity. The Governance of Strategic Risk-Taking / J.A. Torben M.Garvey, O.Roggi – Oxford University Press, 2014. – 204 p.
15. Practice Standard for Project Risk Management / Project Mngement Institute, Inc., 2009. – 128 p.
16. Alberts C. Executive Overview of SEI MOSAIC: Managing for Success Using a Risk-Based Approach / C.Alberts, A. Dorofee, L.Marino – Technical Note CMU/SEI-2007-TN-008, 2007. – 33 p.
17. Hillson D.A. Practical project risk management: The ATOM Methodology (2nd ed.) / D.A.Hillson, P.W.Simon – Vienna, US: Management Concepts, 2012 – 410 p.
18. Virine L. Project Risk Analysis Made Ridiculously Simple / L.Virine, M.Trumper – World Scientific Publishing Co. Pte. Ltd., 2017. – 283 p.
19. ISO 27005:2018 Information technology – Security techniques – Information security risk management.. – 56 p. [Electronic resource]. Mode of access: <https://www.iso.org/standard/75281.html>.
20. Freund J. Measuring and Managing Information Risk. A FAIR Approach / J.Freund, J. Jones – Elsevier, 2015. – 391 p.
21. Офіційний сайт NIST CyberSecurity Framework. [Electronic resource]. Mode of access:<https://www.nist.gov/cyberframework>.
22. Grouhy M. The Essentials od Risk Management / M.Grouhy, D.Galai, R.Mark – McGraw-Hill Education, 2014. – 669 p.
23. ISO/IEC PDTR 33015.3:2019 Information technology – Process assessment – Guide to process risk determination. – 41 p.
24. ISO/IEC 16085:2006 Systems and software engineering – Life cycle processes – Risk management. – 34 p..
25. Sherer S.A. Information Systems Risks and Risks Factors, are they Mostly about Information Systems? / S.A.Sherer, S.Alter // Communications of AIS 2004. – V.14. – N 1. – P. 29-64.
26. GAIT for IT General Control Deficiency Assessment / The Institute of Internal Auditors. [Electronic resource]. Mode of access: https://www.iiacolombia.com/resource/guias/GAIT_GeneralControl.pdf. – 25 p.
27. COBIT'2019 Framework. Governance and Management Objectives – ISACA, 2018. – 302 p.
28. Gerstein D.M. et al. Developing a Risk Assessment Methodology for the National Aeronautic and Space Administration /

- D.M.Gerstein et al. – RAND Corporation, 2016. – 113 p.
29. Kendall F. Department of Defense Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs /F.Kendall – Washington, DC 20301-3030, 2017. – 96 p. [Electronic resource]. Mode of access: <https://www.acq.osd.mil/se/docs/2017-rio.pdf>.
 30. NATO Standard AJP-5. Allied Joint Doctrine for the Planning of Operations. Ed. A V.2 / NATO Standardization Office, 2019. – 134 p.
 31. Ilina E.P. Models and methods for automated analytic support of the organization decisions field/ E.P.Ilina, I.P.Sinitsyn / Problems in Programming – 2017. – N 3 – P. 93-107.
 32. Ilina E.P. Methods and models of the expert analytic knowledge using for the decision support in organization. Part 1. Decisions models / E.P.Ilina // Problems in Programming. – 2016. – N 1. – P. 89–101.
 33. Ilyina E. Program Management of Fundamental Scientific Research Decision Making Au-tomated Support with Expert Methodology. / E.Ilyina, O.Slabospitskaya, I.Sinitsyn, T.Yablokova. – Draft of Software Systems Institute of NAS of Ukraine, 2011. – Kiev, 2011. – 94 p.

Одержано 31.10.2019

Про авторів:

Льїна Олена Павлівна,
кандидат фізико-математичних наук,
старший науковий співробітник,
провідний науковий співробітник.
Кількість наукових публікацій в
українських виданнях – більше 60,
<http://orcid.org/0000-0002-1528-366X>,

Слабоспицька Ольга Олександрівна,
кандидат фізико-математичних наук,
старший науковий співробітник,
старший науковий співробітник.
Кількість наукових публікацій в
українських виданнях – більше 50.
Кількість наукових публікацій в
зарубіжних виданнях – 7,
<http://orcid.org/0000-0001-6556-0947>.

Місце роботи авторів:

Інститут програмних систем
НАН України,
03187, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: +38(044) 526 4286.
E-mail: olsips2017@gmail.com

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ В ПРОГРАММНОЙ ИНЖЕНЕРИИ: ОБЗОР

В статье дается краткое описание некоторых методов машинного обучения, таких как искусственные нейронные сети, машины опорных векторов, деревья решений, индуктивное логическое программирование, CBR и других, приводятся примеры применения этих методов для решения проблем прогнозирования и оценки качества в программной инженерии, даются общие рекомендации по применению алгоритмов машинного обучения к задачам программной инженерии. Обзор будет полезен исследователям и практикам в качестве отправной точки, поскольку он обеспечивает важные будущие направления исследований. Это в конечном итоге приведет к более эффективному принятию решений в программной инженерии, тем самым обеспечивая лучшие, более надежные и экономически эффективные программные продукты.

Ключевые слова: программная инженерия, программное обеспечение, машинное обучение, нейронные сети, машины опорных векторов, деревья решений.

Введение

Сегодня достижение успеха в промышленной, военной, коммерческой, социальной и многих других сферах деятельности человека немыслимо без применения различных компьютерных систем, функционирование которых невозможно без соответствующего программного обеспечения (ПО), ставшего основной технологией для развития современного общества и являющегося неотъемлемой, но небезопасной частью повседневной жизни. Человечество знает множество неприятных историй, произошедших из-за проблем с ПО, к примеру, неудачная посадка спускаемого модуля на Марс в 1999 г. обошлась США потерей \$165 млн. или нецелевой запуск ракет «Патриот» в системе обороны США в 1991 г., что привело к гибели мирного населения. Поэтому в настоящее время проблема качества – это главная проблема для мировой индустрии ПО. Аспектами повышения качества и надежности программных продуктов на всех этапах их жизненного цикла, начиная с подготовительных работ и заканчивая снятием их из обращения, занимается отдельная дисциплина – программная инженерия (software engineering, **SE**).

Основные проблемы разработки и поддержки больших программных систем в изменяющейся среде были изложены в классической статье Ф. Брукса [1] еще в

далеком 1987 году. В ней же отмечалось, что многие из классических проблем разработки программных продуктов обусловлены их существенной сложностью, которая нелинейно увеличивается с ростом размера ПО. Спустя более 30 лет разнообразие, размеры и сложность ПО многократно увеличились, что привело к появлению множества новых чрезвычайно сложных проблем, возникающих на всех этапах его жизненного цикла [2]. Уже в конце XX столетия возникла острая необходимость в поиске альтернативы существующим технологиям разработки и анализа ПО. Обнадеживающие результаты в преодолении некоторых из возникающих трудностей дало применение в программной инженерии методов искусственного интеллекта [3–8]. В своей основной речи на ежегодной конференции Американской ассоциации искусственного интеллекта в 1992 году О. Селфридж призывал к активному использованию для решения проблем программной инженерии методов машинного обучения (Machine Learning, **ML**) – одной из важных подобластей искусственного интеллекта [9].

Последние два десятилетия показали, что многие задачи по разработке и сопровождению ПО могут быть сформулированы как проблемы обучения и алгоритмы ML являются жизнеспособной альтернативой существующим подходам к реше-

нию многим проблем SE. Чтобы лучше использовать методы ML в качестве инструментов для решения реальных проблем SE, необходимо знать, как эти проблемы, так и условия, при которых имеющиеся в нашем распоряжении методы ML могут быть наиболее эффективно применены.

Остальная часть статьи организована следующим образом: первый раздел посвящен краткому обзору наиболее важных, с точки зрения SE, методов ML; во втором разделе рассматриваются вопросы применения ML для решения проблем прогнозирования и оценки качества ПО; в третьем разделе даются некоторые рекомендации по применению алгоритмов ML к задачам SE, далее приведены основные выводы.

1. Подходы к машинному обучению

В качестве подобласти искусственного интеллекта ML имеет дело с тем, как создавать компьютерные программы, которые улучшают свою производительность при выполнении определенной задачи с помощью опыта [8]. Поскольку многие задачи разработки или сопровождения SE зависят от некоторой функции (или функций, отображений или моделей) для прогнозирования, оценки, классификации, диагностики, обнаружения, приобретения, понимания, генерации или преобразования определенного качественного или количественного аспекта программного артефакта или программного процесса, применение ML в SE сводится к тому, как в процессе обучения найти такую целевую функцию (отображение или модель), которую можно использовать для выполнения задач SE.

В зависимости от используемых методов обучения целевая функция может быть представлена в различных формах языка гипотез (например, деревья решений, битовые строки или правила). Когда целевая функция не определена явно, но учащийся может сгенерировать ее значения для заданных входных запросов (например, в случае обучения на основе экземпляров), то говорят, что функция определена неявно.

При изучении целевой функции на заданном наборе обучающих данных ее обобщение может быть либо *усердным* (на этапе обучения), либо *ленивым* (на этапе классификации). Усердное обучение может создать одну целевую функцию из всех данных обучения, в то время как ленивое обучение может принимать разные (неявные) функции для каждого запроса. Оценка целевой функции зависит от многих факторов: точности прогнозирования, интерпретируемости, статистической значимости, содержания информации и компромисса между ее сложностью и степенью соответствия данным [10].

Алгоритмы обучения в зависимости от чувствительности к изменениям данных обучения делятся на стабильные и нестабильные [11]. Для нестабильных алгоритмов небольшие изменения в обучающих данных приведут к тому, что алгоритмы будут генерировать существенно различную выходную функцию. С другой стороны, стабильные алгоритмы невосприимчивы к небольшим изменениям в данных [11].

Существует четыре основных типа машинного обучения [8, 12, 13]:

- **Контролируемое обучение** (supervised learning): все входные и выходные данные помечены, и алгоритмы учатся предсказывать выходные данные из входных данных, т. е. целевая функция изучается на основе обучающих примеров ее входов и выходов.

- **Неконтролируемое обучение** (unsupervised learning): все данные не помечены, и алгоритмы учатся присваивать структуру из входных данных. Пытаются изучить шаблоны на входе, для которых нет доступных выходных значений. Примером является кластеризация.

- **Частично контролируемое обучение** (semi-supervised learning): некоторые данные помечены, но большинство из них немаркированы, (данные обучения включают в себя несколько желаемых результатов.) Может использоваться смесь контролируемых и неконтролируемых методов. Хорошим примером является фотоархив, в котором помечены только некоторые

изображения (например, собака, кошка, человек), а большинство изображений не имеет маркировки.

• **Обучение с подкреплением** (reinforcement learning): это наиболее общая форма обучения. В нем рассматривается вопрос о том, как выучить последовательность действий, называемых стратегией управления, из косвенной и отложенной информации о вознаграждении.

Контролируемое обучение является наиболее зрелым и изученным типом обучения. Подавляющее большинство используемых в SE методов ML принадлежат этому типу обучения. Далее приводится краткое описание основных методов обучения, имеющих отношение к данной работе, в следующих группах:

- деревья решений (decision trees, **DT**);
- нейронные сети (neural networks, **NN**);
- обучение на основе экземпляров (instance-based learning, **IBL**);
- индуктивное логическое программирование (inductive logic programming, **ILP**);
- машины опорных векторов (support vector machines, **SVM**);
- генетические алгоритмы (Genetic Algorithms, **GA**);
- генетическое программирование (genetic programming, **GP**);
- байесовское обучение (Bayesian learning, **BL**);

Организация различных методов обучения в группы в значительной степени находится под влиянием [8, 11].

Деревья решений

Деревья решений, предложенные в [14, 15], состоят из листьев и ветвей, где листья представляют классификации, а ветви – соединения признаков, которые приводят к классификациям. Таким образом, DT являются инструментом моделирования принятия решений, который графически отображает процесс классификации заданных входных данных для заданных меток выходных классов. Внутренние

узлы DT обозначают различные атрибуты, а ветви между узлами представляют возможные значения, которые эти атрибуты могут иметь в наблюдаемых выборках.

В алгоритмах DT целевая функция определяется как дерево решений. Поиск в DT часто управляется мерой прироста информации, основанной на энтропии, которая показывает, сколько информации дает тест по атрибуту. Алгоритмы обучения DT часто имеют тенденцию к небольшим деревьям. Это усердный, контролируемый и нестабильный метод обучения, подверженный шумным данным, что является причиной переобучения. Он не может использовать априорные знания в процессе обучения. Тем не менее, он хорошо масштабируется с большими данными несколькими различными способами [11].

Деревья решений являются одним из самых популярных алгоритмов классификации, применяемых в интеллектуальном анализе данных и машинном обучении. Наиболее востребованными алгоритмами DT являются такие как ID3 (*iterative dichotomiser 3*), C4.5, C5, J48 и CART (*classification and regression trees*) [16–19].

Нейронные сети

Каждая из миллиардов клеток нейронов человеческого мозга подобна крошечному компьютеру с чрезвычайно ограниченными возможностями. Искусственные NNs формируются из сотен или тысяч симулированных нейронов, соединенных вместе подобно нейронам в мозге [20]. Модель нейрона показана на рисунке.

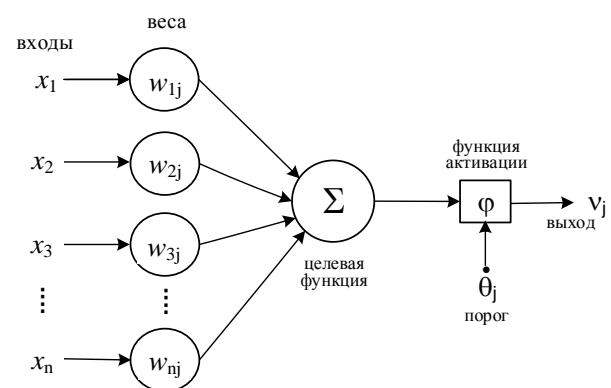


Рисунок. Нелинейная модель нейрона

Нейронные сети, являясь мощным инструментом анализа данных, способна фиксировать и представлять сложные отношения ввода/вывода. Мотивацией для разработки технологии NN является создание искусственной системы, которая могла бы выполнять «интеллектуальные» задачи, подобные тем, которые выполняются человеческим мозгом. NNs подобно человеческому мозгу приобретают знания через обучение и хранят их в пределах межнейронных связей, известных как синоптические веса [21].

Истинная сила и преимущество NN заключается в их способности представлять, как линейные, так и нелинейные отношения и изучать эти отношения непосредственно из моделируемых данных [22]. Традиционные линейные модели просто неадекватны, когда речь идет о данных моделирования, содержащих нелинейные характеристики.

Учитывая фиксированную структуру NN, обучение ее целевой функции равнозначно нахождению вектора весов соединений, минимизирующего сумму квадратов ошибок, применяемых ко всем наборам данных. Поскольку вектор весов по существу определяет целевую функцию, то это делает ее очень трудной для чтения и интерпретации человеком.

Нейронные сети – это усердный, контролируемый и нестабильный подход к обучению, который не может использовать предварительные знания. Популярным алгоритмом для сетей прямой связи является Backpropagation, который принимает поиск с градиентным спуском и санкционирует индуктивное смещение гладкой интерполяции между точками данных [8]. Среди других архитектур NN, применяемых в SE, выделим следующие: многослойный перцептрон (multilayer perceptron, MLP) [23], сеть радиально базисных функций (radial basis function network, RBFN) [24], вероятностная нейронная сеть (probabilistic neural network, PNN) [25], сеть Элмана (Elman network, EN) [26], «самоорганизующаяся» каскадно-корреляционная нейронная сеть (Cascade correlation neural network, CCNN) [27], карта самоорганизующихся объектов (self-organising feature map, SOM) [28].

Обучение на основе экземпляров

В машинном обучении IBL представляет собой семейство алгоритмов обучения, которые вместо выполнения явного обобщения сравнивают новые проблемные экземпляры с экземплярами, наблюдаемыми в процессе обучения, хранящиеся в памяти. По этой причине IBL иногда называют обучением на основе памяти [29]. IBL – это типичный метод ленивого обучения в том смысле, что обобщение за пределы обучающих данных откладывается до классификации невидимого случая. Кроме того, целевая функция явно неопределена; вместо этого учащийся возвращает значение целевой функции при классификации данного невидимого случая. Значение целевой функции генерируется на основе подмножества обучающих данных, считающихся локальными для невидимого примера, а не на всех обучающих данных. Это равносильно аппроксимации другой целевой функции для отдельного невидимого примера. Это значительный отход от методов усердного обучения, когда одна целевая функция получается в результате того, что учащийся обобщает все данные обучения. Процесс поиска основан на статистических рассуждениях и состоит в определении данных обучения, которые близки к данному невидимому случаю, и создании значения целевой функции на основе его соседей. Одним из преимуществ IBL по сравнению с другими методами ML является его способность адаптировать свою модель к ранее невидимым данным. Учащиеся на основе экземпляров могут просто сохранить новый экземпляр или выбросить старый.

Наиболее известными алгоритмами IBL являются алгоритмы k-ближайших соседей (K-nearest neighbors) [30] и рассуждения по прецедентах (Case-Based Reasoning, CBR) [31, 32].

Индуктивное логическое программирование

Индуктивное логическое программирование впервые упоминается в статье [33], где «индуктивное» употребляется в философском (предложение теории для объяснения наблюдаемых фактов), а не в

математическом (доказательство свойства членов множества) смысле. Сегодня ИЛР – раздел машинного обучения, который использует логическое программирование как форму представления примеров, фоновых знаний и гипотез. Получив описания уже известных фоновых знаний и набор примеров, представленных как логическая база фактов, система ИЛР может породить логическую программу в форме гипотез, объясняющую все положительные примеры и ни одного отрицательного. Поскольку целевая функция в ИЛР определяется набором правил (пропозициональных или первого порядка), она легко поддается восприятию и интерпретации человеком. ИЛР позволяет использовать фоновые знания в процессе обучения и является усердным и контролируемым обучением. Наиболее известными алгоритмами ИЛР являются FOIL [34, 35] и PROGOL [36].

Машины опорных векторов

Предложенный В.Н. Вапником метод SVM [37] представляет собой набор схожих алгоритмов обучения, использующихся для задач классификации и регрессионного анализа. Принадлежит семейству линейных классификаторов и может также рассматриваться как специальный случай регуляризации по А.Н. Тихонову. Являясь не вероятностным двоичным линейным классификатором, SVM, вместо того, чтобы изучать нелинейную целевую функцию непосредственно из данных во входном пространстве, использует функцию ядра (определенную в форме внутреннего произведения обучающих данных) для преобразования обучающих данных из входного пространства сначала в пространство объектов большого размера F , а затем изучает оптимальный линейный разделитель (гиперплоскость) в F . Функцию выбора решения, определенную на основе линейного разделителя, можно использовать для классификации невидимых случаев. Функция ядра играет ключевую роль в SVM. Она основывается только на подмножестве обучающих данных, называемых опорными векторами.

Модели SVM тесно связаны с искусственными NN [38, 39]. Поэтому SVM

можно использовать в качестве альтернативного метода обучения для полиномиальной, радиальной базисной функции и многослойных сетей персептрона с использованием функции ядра [40].

Генетические алгоритмы и генетическое программирование

GA был разработан в [41] как альтернативный метод решения общих задач оптимизации с большими поисковыми пространствами. Он имеет то преимущество, что ему не нужны предварительные знания, опыт или логика, связанные с конкретным решением проблемы. Основные идеи GA основаны на дарвиновской теории эволюции, которая в сущности говорит о том, что генетические операции между хромосомами особей в конечном итоге приводят к выживанию сильнейших. Таким образом, в течение длительного периода времени улучшается популяция особей в целом. Целевая функция представляется в виде битовых строк. Процесс поиска начинается с совокупности исходных гипотез. Посредством операций кроссинговера и мутации особи текущей популяции образуют новую популяцию с лучшими средними характеристиками. На каждом шаге итерации гипотезы в текущей популяции оцениваются с учетом заданной степени пригодности, причем наиболее подходящие гипотезы выбираются в качестве членов следующего поколения. Процесс поиска заканчивается, когда некоторая гипотеза имеет значение пригодности выше некоторого порога. Таким образом, процесс обучения по существу воплощен в поиске по пучку траектории с порождением и проверкой вариантов [8].

Предложенное в [42] GP в машинном обучении используется для автоматического создания или изменения программ с помощью GA. Выбор способа кодирования программы в GA – один из основных вопросов в GP. Программа должна быть закодирована в таком виде, чтобы легко было автоматически вносить случайные изменения (оператор мутации) и объединять два алгоритма в один (оператор скрещивания). Различают два способа кодирования: прямое и косвенное. В первом

случае GA работает с программой в явном виде, а во втором – GA работает не с самим кодом программы, а с правилами его построения, то есть GA работает с программой, генерирующая нужную нам программу.

Байесовское обучение

ВL предлагает вероятностный подход к выводу, который основан на предположении, что интересующие величины определяются распределением вероятностей, и что оптимальные решения или классификации могут быть достигнуты путем рассуждения об этих вероятностях наряду с наблюдаемыми данными [8]. Методы ВL на основе результатов учащегося могут быть разделены на две группы: те, которые дают наиболее вероятную гипотезу с учетом данных обучения, и те, которые производят наиболее вероятную классификацию нового экземпляра с учетом данных обучения. Таким образом, целевая функция явно представлена в первой группе, но неявно определена во второй группе. Одним из основных преимуществ является то, что ВL учитывает предшествующее знание (в форме байесовских сетей доверия, априорных вероятностей для гипотез-кандидатов или распределения вероятностей по наблюдаемым данным для возможной гипотезы). Классификация невидимого случая получается посредством комбинированных предсказаний нескольких гипотез. Это также хорошо масштабируется с большими данными. ВL – усердный и контролируемый метод обучения, который не требующий поиска в процессе обучения. Хотя ВL не имеет проблем с зашумленными данными, у него возникают проблемы с небольшими наборами данных. ВL принимает смещение, основанное на принципе минимальной длины описания, который предпочитает гипотезу h , которая минимизирует длину описания h плюс длину описания данных, заданных h [8].

Среди популярных алгоритмов ВL можно выделить следующие: максимальный апостериорный (maximum a posteriori (MAP)), оптимальный байесовский классификатор (Bayes optimal classijer), наив-

ный байесовский классификатор (naive Bayes classijer), GIBBS [8, 11].

2. Применение машинного обучения в программной инженерии

В разработке ПО есть три категории объектов: *процессы* (наборы действий, связанных с ПО, такие как создание спецификации, детальное проектирование или тестирование), *продукты* (артефакты, результаты, документы, являющиеся результатом действия процесса, такие как документ спецификации, проектный документ или сегмент кода) и *ресурсы* (объекты, необходимые для деятельности процесса, такие как персонал, программные средства или оборудование) [2, 43]. Здесь есть внутренние и внешние атрибуты для сущностей вышеназванных категорий. Внутренние атрибуты описывают сам объект, тогда как внешние атрибуты характеризуют поведение объекта (как объект относится к его среде).

В работе [44] выделяются следующие области SE, к решению задач которых успешно применяются методы ML:

- 1) прогнозирование или оценка измерений для внутренних или внешних атрибутов процессов, продуктов или ресурсов;
- 2) обнаружение внутренних или внешних свойств процессов, продуктов или ресурсов;
- 3) преобразование продуктов для достижения некоторых желаемых или улучшенных внешних атрибутов;
- 4) синтезирование различных программных продуктов;
- 5) повторное использование продуктов или процессов;
- 6) улучшение процессов (например, восстановление спецификации из ПО);
- 7) управление специальными продуктами (такими как знания в области дизайна и разработки).

Поскольку на сегодня этот список стал длиннее, а размер статьи ограничен, то в этом разделе мы кратко рассмотрим результаты, попадающие только в первую из эти областей, где методы ML использу-

ються для прогнозування или оценок внутренних, так и внешних атрибутов процессов, продуктов или ресурсов. К ним, в частности, относятся: качество ПО, размер ПО, стоимость разработки ПО, работа над проектом или ПО, работа по техническому обслуживанию, ресурс ПО, стоимость исправления, надежность ПО, дефекты ПО, возможность повторного использования, время выпуска ПО, производительность, время выполнения и тестируемость программных модулей и прочее.

Список приложений, включенных в раздел, хотя и далеко неполный, все же должен убедить читателя о необходимости и важности применения методов машинного обучения в программной инженерии.

Прогнозирование качества программного обеспечения

В работе [45] GP используется для генерации моделей качества ПО, принимающих в качестве входных данных метрики ПО, собранные ранее в процессе разработки, и прогнозируют для каждого модуля количество ошибок, которые будут обнаружены позднее в процессе разработки или во время эксплуатации. Эти прогнозы станут основой для ранжирования модулей, что позволит менеджеру выбрать столько модулей в верхней части списка, сколько ресурсов позволяют повысить надежность.

В работе [46] проведено эмпирическое сравнение нескольких методов моделирования (логистическую регрессию, многоуровневые нейронные сети, графические сети и др.) для прогнозирования качества программных компонентов в начале жизненного цикла ПО. Другая работа по прогнозированию качества ПО на основе NN, как сообщается в [47], связана с конкретным языком, где сначала определяются метрики проектирования для языка SDL (язык спецификации и описания), а затем используются при построении моделей прогнозирования для идентификации подверженных сбоям компонентов. В работе [48] модели, основанные на NN, используются для прогнозирования ошибок и показателей качества ПО. В работе [49]

использован алгоритм CART для оценки моделей качества ПО в течение нескольких выпусков. Авторы использовали очень большую унаследованную телекоммуникационную систему для разработки двух деревьев классификации качества ПО. Оба модуля дали лучшую точность, которая была бы полезной для разработчиков в различных методах разработки ПО.

В статье [50] представлено использование эволюционных DT в качестве подхода к прогнозированию ошибок. Было показано, что α -метрика как атрибут вместе с другими показателями сложности ПО может быть успешно использована для создания деревьев решений для прогнозирования опасных модулей (модулей, имеющих много необнаруженных ошибок). Модернизация таких модулей или выделение для них дополнительных усилий по тестированию или обслуживанию может в значительной степени повысить качество и надежность, сделав программное обеспечение гораздо более безопасным в использовании. В работах [51–55] для прогнозирования качества ПО используется метод обучения CBR. Основное внимание в [51] уделяется сравнению производительности различных классификаторов CBR. В работе [52] CBR применяется для моделирования качества ПО семейства полномасштабных промышленных программных систем, и точность прогнозируется лучше, чем соответствующая модель множественной линейной регрессии при прогнозировании числа ошибок проектирования. Два практических правила классификации (большинство голосов и кластеризация данных) предложены в [53] для оценки качества ПО высоконадежных систем. В работе [54] обсуждается процедура выбора атрибута, который может помочь идентифицировать соответствующие метрики качества ПО, которые будут использоваться в прогнозировании качества на основе CBR. В работе [55] подход CBR используется для калибровки моделей классификации качества ПО.

В работе [56] подход, основанный на DT, используется для генерации моделей компонентов высокого риска. Предложенный метод опирается на исторические

данные (показатели из предыдущих выпусков или проектов) для определения компонентов, склонных к сбоям. Другой подход, основанный на DT, используется при построении моделей для прогнозирования компонентов Ada высокого риска [57]. Деревья регрессии используются в [58] для классификации программных модулей, подверженных ошибкам. Подход позволяет иметь предпочтительный баланс между типами ошибочной классификации типов I и II. Алгоритм SPRINT DT используется в [59] для построения деревьев классификации как моделей оценки качества, предсказывающих класс программных модулей (подверженных ошибкам или не подверженных ошибкам). Набор методов вычислительного интеллекта, одним из которых является DT, предложен в [60] для анализа качества ПО. Гибридный подход, основанный на GP и DT, предложен в [61] для прогнозирования качества ПО и позволяет оптимизировать как показатели производительности, так и точности. Другой сравнительный результат исследования представлен в [62] об использовании методов ILP при прогнозировании сбоев ПО для программ на C++. Как естественные, так и искусственные данные используются при оценке производительности двух методов ILP (FOIL и FLIPPER), для одного из них предлагаются некоторые расширения.

Оценка размера программного обеспечения

NN и GP используются в [63] для валидации основанного на компонентах метода для оценки размера ПО. В дополнение к полученным результатам, подтверждающие компонентный подход к определению размера ПО, в исследовании отмечается, что NN хорошо работает с данными, распознавая некоторые нелинейные отношения, которые не удается обнаружить методом множественной линейной регрессии. Уравнения, разработанные GP, дают аналогичные или лучшие значения, чем полученные уравнениями регрессии, и являются понятными, обеспечивая уверенность в результатах.

Прогноз стоимости программного обеспечения

Общий подход, называемый *оптимизированное сокращение набора* (optimized set reduction) и основанный на DT, описан в [64] для анализа данных разработки ПО и продемонстрирован как эффективный метод оценки стоимости ПО. Сравнительное исследование нескольких методов моделирования стоимости ПО, основанное на реальных данных, проведено в [65] и включает метод CBR для прогнозирования стоимости ПО. Результат работы [66] указывает на то, что улучшенные прогностические характеристики моделей стоимости ПО могут быть получены с использованием байесовского анализа, предлагающего структуру, в которой могут быть использованы как предварительные экспертные знания, так и данные выборки для получения прогнозов. В работе [67] предложен подход на основе GP для поиска возможных функций стоимости ПО.

Прогнозирование усилий по разработке ПО (проекта)

Методы IBL используются в [68] для прогнозирования усилий проекта ПО для новых проектов. Полученные эмпирические результаты (из девяти различных промышленных наборов данных, насчитывающих в общей сложности 275 проектов) показывают, что CBR предлагает жизнеспособное дополнение к существующим методам прогнозирования и оценки. Другое применение CBR в оценке усилий ПО описано в [69]. Работа [70] фокусируется на эвристике поиска, чтобы помочь определить оптимальный набор функций в системе CBR для прогнозирования усилий по проекту ПО. Сравнение нескольких методов оценки CBR сделано в [71], и результаты показывают, что оценки, полученные с помощью аналогов, выбранных человеком, являются более точными, чем оценки, полученные с помощью аналогов, выбранных инструментами, и более точными, чем оценки с помощью простой регрессионной модели. В работе [72] DT и NN используются для прогнозирования усилий по раз-

работке ПО. Результаты были конкурентоспособны с традиционными методами, такими как СОСОМО и функциональные точки. Основное преимущество систем оценки на основе DT и NN заключается в том, что они являются адаптируемыми и непараметрическими. В работах [73, 74] для прогнозирования усилий по разработке ПО также использовались NN и результаты обнадеживают с точки зрения точности. Дополнительные исследования по прогнозированию усилий на основе ML включают в себя: генетически обученный предиктор NN (GA + NN) [75] и подход, основанный на GP [76]. В работах [77–79] сообщалось о нескольких сравнительных исследованиях оценки программных усилий с использованием NN&CBR в [78], CBR&NN&DT в [79] и NN&CBR&GP в [77].

Прогнозирование усилий по обслуживанию

В работе [80] модели генерируются в терминах методов NN и DT и методов регрессии для прогнозирования усилий по обслуживанию ПО. Исследование измеряет и сравнивает точность прогноза для каждой модели и делает вывод, что модели на основе DT и множественной регрессии имеют лучшие результаты точности. Рекомендуется использовать прогнозные модели в качестве инструментов для поддержки экспертных оценок и анализа влияния переменных технического обслуживания на процесс и продукт технического обслуживания.

Анализ программных ресурсов

В работе [81] DT используется при анализе данных о программных ресурсах для определения классов программных модулей, испытывающих большие усилия при разработке или отказы (понятие «высокий» определено в отношении самого верхнего квартиля относительно прошлых данных). Шестнадцать программных систем используются в исследовании. Деревья решений правильно идентифицируют 79,3 процента программных модулей, у

которых были большие усилия по разработке или сбои.

Оценка стоимости исправления ошибок

Эмпирическое исследование сделано в [82], где DT и ILP используются для генерации моделей для оценки стоимости коррекции при обслуживании ПО. Сгенерированные модели оказываются полезными для оптимизации распределения ресурсов при корректирующих действиях по техническому обслуживанию и принятию решений относительно того, когда следует реструктурировать или реинжиниринговать компонент, чтобы сделать его более обслуживаемым. Сравнение приводит к наблюдению, что результаты на основе ILP работают лучше, чем результаты на основе DT.

Прогноз повторного использования

Прогнозирующие модели построены с помощью DT в [83], чтобы проверить влияние некоторых внутренних свойств объектно-ориентированных приложений на возможность повторного использования. Усилия направлены на установление корреляции между возможностью повторного использования компонентов и тремя атрибутами ПО (наследование, связывание и сложность). Результаты эксперимента показывают, что некоторые программные метрики могут использоваться для прогнозирования с высокой степенью точности потенциальных классов многократного использования.

Время выпуска программного обеспечения

Как определить график выпуска ПО – это проблема, влияющая как на разработчика программного продукта, так и на пользователя и рынок. Метод, основанный на NN, предложен в [84] для оценки оптимального времени выпуска ПО. Метод принимает критерий минимизации затрат и переводит его в задачу прогнозирования временных рядов. Затем NN используется

для оценки времени обнаружения неисправности в будущем.

Прогнозирование тестируемости

В работе [85] описывается пример, в котором NN используется для прогнозирования тестируемости программных модулей на основе статических измерений исходного кода. Цель исследования – предсказать величину от нуля до единицы, распределение которой сильно смещено к нулю, что затруднительно для стандартных статистических методов. Результаты отражают характерную особенность прогнозирующих моделей на основе NN, обсуждаемые по настоящее время: их способность моделировать нелинейные отношения.

Производительность

Подход, основанный на BL, описан в [86] для оценки производительности программных проектов. Определена демонстрационная байесовская сеть доверия для захвата причинно-следственных связей между компонентами в модели СОСОМО81 вместе с таблицами вероятностей для узлов.

Время выполнения

Временное поведение ПО в реальном времени имеет решающее значение для общей правильности системы. Таким образом, проверка того, нарушает ли система реального времени определенные ограничения по времени для определенных входных данных, становится критической проблемой. Подход, основанный на GA, описан в [87] для получения входных данных с самым длинным или самым коротким временем выполнения, которые можно использовать для проверки того, приведут ли они к временной ошибке или нарушению временных ограничений системы.

Прогнозирование надежности программного обеспечения

В работе [88] сообщается об использовании NN для прогнозирования роста надежности ПО. Эмпирическое сравнение проводится между NN-моделями и пятью известными моделями роста надеж-

ности ПО с использованием фактических наборов данных из ряда различных программных проектов. Результаты показывают, что основанные на NN модели хорошо адаптируются к различным наборам данных и имеют лучшую точность прогнозирования. Однако эффективность моделей прогнозирования на основе нейронной сети зависит от поведения набора данных, который в основном носит флуктуирующий характер. Поэтому часто NN страдает от переобучения результатов при работе с реальными неизвестными наборами данных. В работе [89] предложена модель прогнозирования надежности ПО на основе SVM и показано, что данные об ошибках, собранные на ранних этапах жизненного цикла разработки ПО, являются более подходящими для использования и влияют на точность прогнозирования. В работе [90] исследовали состояние методов раннего прогнозирования надежности ПО, используя SVM. Определено, что модель раннего прогнозирования, основанная на SVM, более точна в своем прогнозе с лучшей способностью обобщения. В работе [91] проводилось эмпирическое исследование нескольких методов ML, таких как ANN (BPN, RBFN и сеть Элмана), SVM, CCNN, DT и др., для прогнозирования надежности ПО. В частности, изучался вопрос: являются ли рабочие характеристики SVM и DT лучше, чем CCNN и NN. Отмечалось, что хотя DT действительно демонстрируют свои сильные стороны в различных реальных приложениях, однако на практике они неэффективны для прогнозирования надежности ПО.

Предсказание дефектов

Для прогнозирования дефектов ПО в работе [92] используется BL. Хотя система, о которой сообщается, является всего лишь прототипом, она показывает потенциал байесовских сетей доверия (BBN) в объединении нескольких точек зрения на прогнозирование дефектов в единую унифицированную модель. Переменные в прототипе системы BBN выбираются для представления процессов спецификации, проектирования и реализации, а также тестирования жизненного цикла (сложность

проблеми, усилия на проект, размер проекта, введенные дефекты, тестирующее усилие, обнаруженные дефекты, плотность дефектов при тестировании, остаточное количество дефектов и остаточная плотность дефектов). Надлежащие причинно-следственные связи между этими процессами жизненного цикла ПО затем фиксируются и отражаются в виде дуг, соединяющих переменные. Затем используется инструмент в отношении модели BBN следующим образом. Для заданных фактов об усилиях на проект и размере проекта в качестве входных данных инструмент будет использовать байесовский вывод для получения распределений вероятностей для введенных дефектов, обнаруженных дефектов и плотности дефектов.

3. Общие рекомендации по применению алгоритмов ML к задачам SE

Применяя машинное обучение к решению любой реальной проблемы, обычно нужно следовать определенному курсу действий. Проблемы программной инженерии не являются исключением. Предлагаемые рекомендации по применению ML к задачам SE имеют такие шаги.

Постановка проблемы. Первый шаг – сформулировать данную проблему так, чтобы она соответствовала структуре конкретного метода обучения, выбранного для этой задачи. Разные методы обучения имеют различную индуктивную предвзятость, используют разные стратегии поиска, основанные на различных руководящих факторах, предъявляют различные требования в отношении теории предметной области (наличие или отсутствие) и обучающих данных (оценка и свойства) и основаны на разных рассуждениях обоснования. Все эти вопросы должны быть приняты во внимание на этапе формулирования проблемы. Этот шаг имеет решающее значение для применимости метода обучения. Такие стратегии, как «разделяй и властвуй», могут потребоваться для разложения исходной проблемы на набор подзадач, более поддающихся выбранному методу обучения. Иногда лучшая формулировка проблемы не всегда может быть наиболее

интуитивной для исследователя машинного обучения [94].

Проблемное представление. Следующим шагом является выбор подходящего представления как для данных обучения, так и для знаний, которые необходимо изучить. Разные методы обучения имеют разные формализмы представления. Таким образом, представление атрибутов и функций в учебной задаче часто зависит от конкретной проблемы и от формализма.

Сбор информации. Третий шаг – сбор данных, необходимых для учебного процесса. Качество и количество необходимых данных зависят от выбранного метода обучения. Данные, возможно, должны быть предварительно обработаны, прежде чем они могут быть использованы в процессе обучения.

Подготовка теории предметной области. Некоторые методы обучения (например, EBL) полагаются на доступность теории предметной области для данной проблемы. Приобретение и подготовка теории предметной области (или базовых знаний) и обеспечение качества этой теории (правильность, полноту) становится важной проблемой, которая повлияет на результат процесса обучения.

Выполнение учебного процесса. Как только данные и теория предметной области (при необходимости) готовы, процесс обучения может быть выполнен. Данные будут разделены на тренировочный набор и тестовый набор. Если используется какой-либо инструмент обучения или среда, данные обучения и тестовые данные, возможно, должны быть организованы в соответствии с требованиями инструмента. Знания, полученные из обучающего набора, проверяются на тестовом наборе. Из-за различий между обучающим набором и тестовым набором сам процесс обучения является итеративным.

Анализ и оценка полученных знаний является неотъемлемой частью учебного процесса. Интерес и эффективность полученных знаний будут тщательно изучены на этом этапе, часто с помощью экспертов-людей, что, как мы надеемся, приведет к улучшению знаний. Если полученные знания считаются несущественными,

неінтересними, неуместними или отклоняющимися, это может указывать на необходимость пересмотра на ранних стадиях, таких как постановка проблемы и представление. Во многих методах обучения существуют известные практические проблемы, такие как переоснащение, локальные минимумы или проклятие размерности, которые обусловлены либо неадекватностью данных, шумом или не относящимися к ним атрибутами в данных, характером стратегии поиска или неверной теорией предметной области.

Хранение базы знаний. Этот шаг влечет за собой использование полученных знаний [93]. Знания могут быть встроены в систему разработки программного обеспечения или в программный продукт или использованы без встраивания их в компьютерную систему. Как отмечалось в [93], сила методов машинного обучения заключается не в конкретном индукционном методе, а в правильной постановке задач и создании представления, позволяющего сделать процесс обучения более понятным.

Выводы

Область разработки ПО оказалась благодатной почвой, где многие задачи по разработке и сопровождению ПО могут быть сформулированы как проблемы обучения и подходят с точки зрения алгоритмов обучения. Последние два десятилетия стали свидетелями многих приложений МЛ в разработке и обслуживании ПО. Большинство публикаций посвящены вопросу о том, как создавать модели для прогнозирования или оценки определенных свойств процесса разработки ПО или артефактов. При этом формируется некоторое представление о том, какие типы техник МЛ чаще других используются при решении задач прогнозирования и оценки качества в программной инженерии. Тремя основными из них являются CBR, NN и DT.

Литература

1. Brooks F. No silver bullet: essence and accidents of software engineering. *IEEE Computer*. 1987. Vol. 20, N 4. P. 10–19.

2. Андон Ф.И. и др. Основы инженерии качества программных систем. 2-е изд., перераб. и доп. К.: Академперіодика. 2007. 672 с.
3. Lowry M. Software engineering in the twenty first century. *AI Magazine*. 1992. Vol. 14, N 3. P. 71–87.
4. Mostow J. Special issue on artificial intelligence and software engineering. *IEEE Trans. SE*. 1985. Vol. 11, N 11. P. 1253–1408.
5. Partridge D. Artificial Intelligence and Software Engineering. AMACOM. 1998. 277 p.
6. Rich C., Waters R. Readings in Artificial Intelligence and Software Engineering. Morgan Kaufmann. 1986. 589 p.
7. Tsai J.J.P., Weigert T., Knowledge-Based Software Development for Real-Time Distributed Systems. *World Scientific Inc*. 1993. 236 p.
8. Mitchell T. Machine Learning. *McGraw-Hill*. 1997. 414 p.
9. Selfridge O. The gardens of learning: a vision for AI. *AI Magazine*. 1993. Vol.14, N 2. P. 36–48.
10. Quinlan J.R. Some elements of machine learning. *9th International Workshop on Inductive Logic Programming. Lecture Notes in Artificial Intelligence*. Springer-Verlag. 1999. Vol. 1634. P.15–18.
11. Dietterich T.G. Machine learning research: four current directions. *AI Magazine*. 1997. Vol. 18, N 4. P. 97–136.
12. Seeger M. Learning with labeled and unlabeled data. Technical Report. University of Edinburgh. 2001.
13. Zhu X., Ghahramani Z., Lafferty J. D. Semi-supervised learning using gaussian fields and harmonic functions. *International Conference on Machine Learning (ICML)*. 2003. P. 912–919.
14. Quinlan J.R. Decision trees as probabilistic classifiers. *4th International Workshop on Machine Learning*. Irvine, CA. 1987. P. 31–37.
15. Gehrke J., Ramakrishnan R., Loh W.R. BOAT-optimistic decision tree construction. *ACM SIGMOD International Conference Management of Data*. 1999. P. 169–180.
16. Quinlan J.R. C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo. CA. 1993. 312 p.
17. Breiman L., Friedman J., Olshen R., Stone C. Classification and Regression Trees. Technical report, Wadsworth International, Monterey, CA. 1984. 358 p.

18. Kohavi R. The power of decision tables. *The eighth european conference on machine learning (ECML)*, 1995. P. 174–189.
19. Han J., Kamber M. Data mining: concepts and techniques. Morgan Kaufmann. 2006. 740 p.
20. Lyu M.R. Handbook of Software Reliability Engineering. New York: McGraw-Hill. 1996. 873 p.
21. Park S., Hoyoung N., Sugumaran V. A Semi automated filtering technique for software process tailoring using neural networks. *Expert System and Applications*. 2006. Vol. 30. P. 179–189.
22. Perlovsky L.I. Neural Networks and Intellect: Using Model Based Concepts. New York: Oxford University Press. 2000. 496 p.
23. Rumelhart D.E., Hinto G.E., Williams R.J. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: The MIT Press. 1986. Vol. 1. P. 318–362.
24. Moody J., Darken C.J. Fast learning in networks of locally tuned processing units. *Neural Computing*. 1989. Vol. 1. P. 81–294.
25. Specht D.F. Probabilistic neural networks. *Journal of Neural Networks*. 1990. Vol. 3. P. 110–118.
26. Elman J.L. Finding Structure in Time. *Cognitive science*. 1990. Vol.14, N 2. P. 179–211.
27. Fahlman S.E., Lebiere C. The cascade-correlation learning architecture. *Advances in Neural Information Processing Systems*. San Mated, CA: Morgan Kaufmann.1990. P. 524–532.
28. Kohonen T. Self-Organizing Maps. Berlin: Springer-Verlag. 1997. 513 p.
29. Daelemans W., Van den Bosch A. Memory-Based Language Processing. Cambridge University Press. 2005. 198 p.
30. Russell S.P., Norvig P. Artificial Intelligence. A Modern Approach. New Jersey, USA: Prentice-Hall. 2003. 932 p.
31. Hammond K.J. Case-Based Planning. Academic Press: New York. 1989. 297 p.
32. Kolodner J.L. An introduction to Case Based Reasoning. *Artificial Intelligence Review*. 1992. Vol. 6, N 1. P. 3–34.
33. Muggleton S. Inductive Logic Programming. *New Generation Computing*. 1991. Vol. 8. P. 295–318.
34. Quinlan J.R. Learning logical definitions from relations. *Machine Learning*. 1990. Vol. 5. P. 239–266.
35. Muggleton S., Feng C. Efficient induction of logic programs. *The First Conference on Algorithmic Learning Theory*. Japanese Society for Artificial Intelligence, Tokyo. 1990. P. 368–381.
36. Muggleton S. Inverse Entailment and Progol. *New Generation Computing*. 1995. Vol. 13. P. 245–286.
37. Vapnik V. Statistical learning theory. *Adaptive and Learning Systems*. 1998. Vol. 736.
38. Hanley J, McNeil B. J. The meaning and use of the area under a receiver operating characteristic ROC curve. *Radiology*. 1982. Vol. 143. P. 29–36.
39. Yang B, Xiang L. A study on software reliability prediction based on support vector machines. *International conference on industrial engineering and engineering management (IEEM)*. 2007. P. 1176–1180.
40. Phillip S. DTReg predictive modeling software available. 2003. 395 p. <http://www.dtreg.com>.
41. Goldberg G.E. Genetic Algorithmic Search, Optimization and Machine Learning. Reading, MA: Addison-Wisely. 1989. 624 p.
42. Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press. 1992. 609 p.
43. Fenton N.E., Pfleeger S.L. Software Metrics. PWS Publishing Company, 2nd ed. 1997. 411 p.
44. Zhang D., Tsai J.J.P. Machine learning and software engineering. *Software Quality Journal*. 2003. Vol. 11, Issue 2. P. 87–119.
45. Evett M., Khoshgoftar T., Chien P. and Allen E. GP-based software quality prediction. *Third Annual Genetic Programming Conference*. 1998. P. 60–65.
46. Lanubile F. and Visaggio G. Evaluating predictive quality models derived from software measures: lessons learned. *Journal of Systems and Software*. 1997. Vol. 38. P. 225–234.
47. Hong E., Wu C. Criticality models using SDL metrics set. *4th Asia-Pacific Software Engineering and International Computer Science Conference*. 1997. P. 23–30.
48. Khoshgoftaar T., Pandya A., Lanning D. Application of neural networks for predicting faults. *Annals of Software Engineering*. 1995. Vol. 1. P. 141–154.
49. Khoshgoftaar T.M., Allen E.B., Jones W.D., Hudepohl J.P. Classification – tree models of software quality over multiple releases. *IEEE Transactions on Reliability*. 2000. Vol. 49, N 1. P. 4–11.
50. Kokol P., Podgorelec V., Pighim M. Using software metrics and evolutionary

- decision trees for software quality control. 2001.
<http://www.escom.co.uk/conference2001/papers/kokol.pdf>.
51. El Emam K., Benlarbi S., Goel N., Rai S. Comparing case-based reasoning classifiers for predicting high risk software components. *Journal of Systems and Software*. 2001. Vol. 55, N 3. P. 301–320.
 52. Ganesan K., Khoshgoftaar T., Allen E. Case-based software quality prediction. *International Journal of Software Engineering and Knowledge Engineering*. 2000. Vol. 10, N 2. P. 139–152.
 53. Khoshgoftaar T., Seliya N. Analogy-Based Practical Classification Rules for Software Quality Estimation. *Empirical Software Engineering*. 2003. Vol. 8, N 4. P. 325–350.
 54. Khoshgoftaar T., Nguyen L., Gao K., Rajeevalochanam J. Application of an attribute selection method to CBR-based software quality classification. 15th IEEE *International Conference on Tools with AI*. 2003.
 55. Khoshgoftaar T., Cukic B., Seliya N. Predicting fault-prone modules in embedded systems using analogy-based classification models. *International Journal of Software Engineering and Knowledge Engineering*. 2002. Vol. 12, N 2. P. 201–221.
 56. Porter A., Selby R. Empirically-guided software development using metric-based classification trees. *IEEE Software*. 1990. Vol. 7. P. 46–54.
 57. Briand L., Basili V., Hetmanski C. Developing interpretable models with optimized set reduction for identifying high-risk software components. *IEEE Trans. SE*. 1993. Vol. 19, N 11. P. 1028–1043.
 58. Khoshgoftaar T., Allen E.B., Deng J. Using regression trees to classify fault-prone software modules. *IEEE Transactions on Reliability*. 2002. Vol. 51, N 4. P. 455–462.
 59. Khoshgoftaar T., Seliya N. Software quality classification modeling using the SPRINT decision tree algorithm. 14th IEEE *International Conference on Tools with AI*. 2002. P. 365–374.
 60. Reformat M., Pedrycz W., Pizzi N.J. Software quality analysis with the use of computational intelligence. *Information and Software Technology*. 2003. Vol. 45, N 7. P. 405–417.
 61. Khoshgoftaar T., Liu Y., Seliya N. Genetic programming-based decision trees for software quality classification. 15th IEEE *International Conference on Tools with AI*. 2003.
 62. Cohen W., Devanbu P. A comparative study of inductive logic programming for software fault prediction. *The fourteenth International Conference on Machine Learning*. 1997.
 63. Dolado J. A validation of the component-based method for software size estimation. *IEEE Trans. SE*. 2000. Vol. 26, N 10. P. 1006–1021.
 64. Briand L., Basili V., Thomas W. A pattern recognition approach for software engineering data analysis. *IEEE Trans. SE*. 1992. Vol. 18, No 11. P. 931–942.
 65. Briand L. et al. An assessment and Comparison of common software cost estimation modeling techniques. *International Conference on Software Engineering*. 1999. P. 313–322.
 66. Chulani S., Boehm B., Steece B. Bayesian analysis of empirical software engineering cost models. *IEEE Trans. SE*. 1999. Vol. 25, N 4. P. 573–583.
 67. Dolado J.J. On the problem of the software cost function. *Information and Software Technology*. 2001. Vol. 43, N 1. P. 61–72.
 68. Shepperd M., Schofield C. Estimating software project effort using analogies. *IEEE Trans. SE*. 1997. Vol. 23, N 12. P. 736–743.
 69. Vicinanza S., Prietulla M.J., Mukhopadhyay T. Case-based reasoning in software effort estimation. 11th Intl. Conf. On *Information Systems*. 1990. P. 149–158.
 70. Kirsopp C., Shepperd M. J., Hart J. Search Heuristics, Case-based Reasoning And Software Project Effort Prediction. *Genetic and Evolutionary Computation Conference (GECCO)*. 2002. P. 1367–1374.
 71. Walkerden F., Jeffrey R. An empirical study of analogy-based software effort estimation. *Empirical Software Engineering*. 1999. Vol. 4. P. 135–158.
 72. Srinivasan K., Fisher D. Machine learning approaches to estimating software development effort. *IEEE Trans. SE*. 1995. Vol. 21, N 2. P. 126–137.
 73. Heiat A. Comparison of artificial neural network and regression models for estimating software development effort. *Information and Software Technology*. 2002. Vol. 44, N 15. P. 911–922.
 74. Wittig G., Finnie G. Estimating software development effort with connectionist models. *Information and Software Technology*. 1997. Vol. 39. P. 469–476.
 75. Shukla K. Neuro-genetic prediction of software development effort. *Information and Software Technology*. 2000. Vol. 42, N 10. P. 701–713.

76. Lefley M., Shepperd M. J. Using genetic programming to improve software effort estimation based on general data sets. *Genetic and Evolutionary Computation Conference (GECCO)*. 2003. P. 2477–2487.
77. Burgess C.J., Lefley M. Can genetic programming improve software effort estimation a comparative evaluation. *Information and Software Technology*. 2001. Vol. 43, N 14. P. 863–873.
78. Finnie G., Wittig G., Desharnais J-M. A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models. *Journal of Systems and Software*. 1997. Vol. 39, N 3. P. 281–289.
79. Mair C., Kadoda G., Lefley M., Phalp K., Schofield C., Shepperd M., Webster S. An investigation of machine learning based prediction systems. *Journal of Systems and Software*. 2000. Vol. 53, N 1. P. 23–29.
80. Jorgensen M. Experience with the accuracy of software maintenance task effort prediction models. *IEEE Trans. SE*. 1995. Vol. 21, N 8. P. 674–681.
81. Selby R., Porter A. Learning from examples: generation and evaluation of decision trees for software resource analysis. *IEEE Trans. SE*. 1988. Vol. 14. P. 1743–1757.
82. De Almeida M., Lounis H., Melo W. An investigation on the use of machine learned models for estimating correction costs. *International Conference on Software Engineering*. 1998. P. 473–476.
83. Mao Y., Sahraoui H., Lounis H. Reusability hypothesis verification using machine learning techniques: a case study. 13th *IEEE International Conference on Automated Software Engineering*. 1998. P. 84–93.
84. Dohi T., Nishio Y., Osaki S. Optimal software release scheduling based on artificial neural networks. *Annals of Software Engineering*. 1999. Vol. 8, N 1. P. 167–185.
85. Khoshgoftaar T., Allen E., Xu Z. Predicting testability of program modules using a neural network. *IEEE Symposium on Application-Specific Systems and Software Engineering Technology*. 2000. P. 57–62.
86. Stamelos I., Angelis L., Dimou P., Sakellaris E. On the use of Bayesian belief networks for the prediction of software productivity. *Information and Software Technology*. 2003. Vol. 45, N 1. P. 51–60.
87. Wegener J., Sthamer H., Jones B.F., Eyres D.E. Testing real-time systems using genetic algorithms. *Software Quality Journal*. 1997. Vol. 6. P. 127–135.
88. Karunanithi N., Whitely D., Malaiya Y. Prediction of software reliability using connectionist models. *IEEE Trans. SE*. 1992. Vol. 18, N 7. P. 563–574.
89. Yang B., Xiang L. A study on software reliability prediction based on support vector machines. *International conference on industrial engineering and engineering management (IEEM)*. 2007. P. 1176–1180.
90. Xingguo L., Yanhua S. An early prediction method of software reliability based on support vector machine. *International conference on wireless communications, networking and mobile computing (WiCom)*. 2007. P. 6075–6078.
91. Kumar P., Singh Y. An empirical study of software reliability prediction using machine learning techniques. *International Journal of System Assurance Engineering and Management (Int J Syst Assur Eng Manag)*. 2012. Vol. 3, N 3. P. 194–208.
92. Fenton N., Neil M. A critique of software defect prediction models. *IEEE Trans. SE*. 1999. Vol. 25, N 5. P. 675–689.
93. Langley P., Simon H. Applications of machine learning and rule induction. *Communications of ACM*. 1995. Vol. 38, N 11. P. 55–64.

References

1. Brooks F. (1987), "No silver bullet: essence and accidents of software engineering", *IEEE Computer*, Vol. 20 No.4, pp.10-19.
2. Andon P.I., Koval G.I., Korotune T.M., Lavrischeva E.M. and Suslov V.Yu. (2007), *The Fundamentals for Software Quality Engineering*, 2-nd ed., K.: Akadempriodika, 672 p. (in Russian).
3. Lowry M. (1992), "Software engineering in the twenty first century", *AI Magazine*, Vol.14 No.3, pp.71–87.
4. Mostow J. (1985), "Special issue on artificial intelligence and software engineering", *IEEE Trans. SE*, Vol.11 No.11, pp. 1253–1408.
5. Partridge D. (1998), *Artificial Intelligence and Software Engineering*, AMACOM, 277 p.
6. Rich C. and Waters R.(1986), "Readings in Artificial Intelligence and Software Engineering", Morgan Kaufmann, 589 p.
7. Tsai J.J.P. and Weigert T. (1993), *Knowledge-Based Software Development for*

- Real-Time Distributed Systems, World Scientific Inc., Singapore, 236 p.
8. Mitchell T. (1997), *Machine Learning*, McGraw-Hill, 414p.
 9. Selfridge O. (1993), "The gardens of learning: a vision for AI", *AI Magazine*, Vol.14, N 2. P.36–48.
 10. Quinlan J.R. (1999), "Some elements of machine learning", *Proceedings of the 9th International Workshop on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 1634. P. 15–18.
 11. Dietterich T. G. (1997), "Machine learning research: four current directions", *AI Magazine*. Vol. 18, N 4. P. 97–136.
 12. Seeger M. (2001). *Learning with labeled and unlabeled data* (Technical Report). University of Edinburgh.
 13. Zhu X., Ghahramani Z., and Lafferty J.D. (2003), "Semi-supervised learning using gaussian fields and harmonic functions", *In International Conference on Machine Learning (ICML)*. P. 912–919.
 14. Quinlan J.R. (1987), "Decision trees as probabilistic classifiers", *Proceedings of 4th International Workshop on Machine Learning*, Irvine, CA. P. 31–37.
 15. Gehrke J., Ramakrishnan R. and Loh W.R. (1999) "BOAT-optimistic decision tree construction", *In Proceedings ACM SIGMOD International Conference Management of Data*, Philadelphia, PA. P. 169–180.
 16. Quinlan J.R. (1993), *C4.5: Programs for machine learning*, Morgan Kaufmann, San Mateo, CA, 312 p.
 17. Breiman L., Friedman J., Olshen R. and Stone C. (1984), *Classification and Regression Trees*. Technical report, Wadsworth International, Monterey, CA, 358 p.
 18. Kohavi R. (1995), "The power of decision tables", *In: The eighth european conference on machine learning (ECML-95)*, Heraklion, Greece, P. 174–189.
 19. Han J. and Kamber M. (2006), *Data mining: concepts and techniques*, Morgan Kaufmann, India.
 20. Lyu M.R. (1996) *Handbook of Software Reliability Engineering*. New York: McGraw-Hill.
 21. Park S., Hoyoung N. and Sugumaran V. (2006) "A Semi automated filtering technique for software process tailoring using neural networks", *Expert System and Applications*, Vol. 30. P. 179–189.
 22. Perlovsky L.I. (2000), *Neural Networks and Intellect: Using Model Based Concepts*. New York: Oxford University Press.
 23. Rumelhart D.E, Hinto G.E. and Williams R.J. (1986), "Learning internal representations by error propagation", *In D.E. Rumelhart and J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA: The MIT Press, Vol. 1. P. 318–362.
 24. Moody J. and Darken C.J. (1989), "Fast learning in networks of locally tuned processing units", *Neural Computing*, Vol. 1, P. 81–294.
 25. Specht D.F. (1990), "Probabilistic neural networks", *Journal of Neural Networks*, Vol. 3. P. 110–118.
 26. Elman J. L. (1990), "Finding Structure in Time", *Cognitive science*, Vol. 14, N 2. P. 179–211.
 27. Fahlman S.E. and Lebiere C. (1990), "The cascade-correlation learning architecture", *In Advances in Neural Information Processing Systems*. San Mated, CA: Morgan Kaufmann. P. 524–532.
 28. Kohonen T. (1997), *Self-Organizing Maps*. Berlin: Springer-Verlag, 513p.
 29. Daelemans W. and Van den Bosch A. (2005), *Memory-Based Language Processing*. Cambridge University Press.
 30. Russell S.P. and Norvig, P. (2003), *Artificial Intelligence. A Modern Approach* (2nd ed.). New Jersey, USA: Prentice-Hall, 932 p.
 31. Hammond K. J. (1989), *Case-Based Planning*. Academic Press: New York, 297 p.
 32. Kolodner J.L. (1992), "An introduction to Case Based Reasoning", *Artificial Intelligence Review*, Vol. 6, N 1, P. 3–34.
 33. Muggleton S. (1991), "Inductive Logic Programming", *New Generation Computing*, Vol. 8. P. 295–318
 34. Quinlan J.R. (1990), "Learning logical definitions from relations", *Machine Learning*, Vol. 5. P. 239–266.
 35. Muggleton S. and Feng C. (1990), "Efficient induction of logic programs", *In Proceedings of the First Conference on Algorithmic Learning Theory*, Japanese Society for Artificial Intelligence, Tokyo, pp. 368–381.
 36. Muggleton S. (1995), "Inverse Entailment and Progol", *New Generation Computing*, Vol. 13, pp. 245–286.
 37. Vapnik V. (1998), "Statistical learning theory", *Adaptive and Learning Systems*, Vol. 736.
 38. Hanley J., McNeil B.J. (1982), "The meaning and use of the area under a receiver operating

- characteristic ROC curve", *Radiology*, Vol. 143. P. 29–36.
39. Yang B. and Xiang L. (2007), "A study on software reliability prediction based on support vector machines", *In: Proceedings of international conference on industrial engineering and engineering management (IEEM'07)*, pp. 1176–1180.
 40. Phillip S. (2003), "DTReg predictive modeling software", 395p.
<http://www.dtregr.com>.
 41. Goldberg G.E. (1989), "Genetic Algorithmic Search, Optimization and Machine Learning", Reading, MA: Addison-Wisely, 412p.
 42. Koza J.R. (1992), "Genetic Programming: On the Programming of Computers by Means of Natural Selection", MIT Press, 609 p.
 43. Fenton N.E. and Pfleeger S.L. (1997), *Software Metrics*, PWS Publishing Company, 2nd ed.
 44. Zhang D. and Tsai J.J.P. (2003), "Machine learning and software engineering", *Software Quality Journal*, Vol.11, Issue 2, pp.87–119.
 45. Evett M., Khoshgoftar T., Chien P. and E. Allen, (1998) "GP-based software quality prediction", *Proc. Third Annual Genetic Programming Conference*, P. 60–65.
 46. Lanubile F. and Visaggio G., (1997) "Evaluating predictive quality models derived from software measures: lessons learned", *Journal of Systems and Software*, Vol. 38, P. 225–234.
 47. Hong E. and Wu C., (1997), "Criticality models using SDL metrics set", *Proc. the 4th Asia-Pacific Software Engineering and International Computer Science Conference*, P. 23–30.
 48. Khoshgoftaar T., Pandya A. and Lanning D. (1995), "Application of neural networks for predicting faults", *Annals of Software Engineering*, Vol. 1. P. 141–154.
 49. Khoshgoftaar T.M., Allen E.B., Jones W.D. and Hudepohl J.P. (2000), "Classification – tree models of software quality over multiple releases", *IEEE Transactions on Reliability*, Vol. 49. N 1. P. 4–11.
 50. Kokol P., Podgorelec V. and Pighim M. (2001), "Using software metrics and evolutionary decision trees for software quality control", Available at:
<http://www.escom.co.uk/conference2001/papers/kokol.pdf>.
 51. El Emam K., Benlarbi S., Goel N. and Rai S. (2001), "Comparing case-based reasoning classifiers for predicting high risk software components", *Journal of Systems and Software*, Vol. 55, N 3. P. 301–320.
 52. Ganesan K., Khoshgoftaar T. and Allen E. (2000), "Cased-based software quality prediction", *International Journal of Software Engineering and Knowledge Engineering*, Vol.10 No.2, pp. 139–152.
 53. Khoshgoftaar T. and Seliya N. (2003), "Analogy-Based Practical Classification Rules for Software Quality Estimation", *Empirical Software Engineering*. Vol. 8. N 4. P. 325–350.
 54. Khoshgoftaar T., Nguyen L., Gao K. and Rajeevalochanam J. (2003), "Application of an attribute selection method to CBR-based software quality classification", *Proceedings of 15th IEEE International Conference on Tools with AI*.
 55. Khoshgoftaar T., Cukic B. and Seliya N. (2002), "Predicting fault-prone modules in embedded systems using analogy-based classification models", *International Journal of Software Engineering and Knowledge Engineering*, Vol.12 N 2. P. 201–221.
 56. Porter A. and Selby R.(1990), "Empirically-guided software development using metric-based classification trees", *IEEE Software*, Vol. 7. P. 46–54.
 57. Briand L., Basili V. and Hetmanski C. (1993), "Developing interpretable models with optimized set reduction for identifying high-risk software components", *IEEE Trans. SE*, Vol. 19. N 11. P. 1028–1043.
 58. Khoshgoftaar T., Allen E.B. and Deng J. (2002), "Using regression trees to classify fault-prone software modules", *IEEE Transactions on Reliability*, Vol. 51. N 4. P. 455–462.
 59. Khoshgoftaar T. and Seliya N. (2002), "Software quality classification modeling using the SPRINT decision tree algorithm", *Proceedings of 14th IEEE International Conference on Tools with AI*. P. 365–374.
 60. Reformat M., Pedrycz W. and Pizzi N.J. (2003), "Software quality analysis with the use of computational intelligence", *Information and Software Technology*, Vol.45 No.7, pp.405–417.
 61. Khoshgoftaar T., Liu Y. and Seliya N. (2003), "Genetic programming-based decision trees for software quality classification", *Proceedings of 15th IEEE International Conference on Tools with AI*.
 62. Cohen W. and Devanbu P. (1997), "A comparative study of inductive logic programming for software fault prediction", *Proc. the fourteenth International Conference on Machine Learning*.

63. Dolado J. (2000), "A validation of the component-based method for software size estimation", *IEEE Trans. SE*, Vol.26 No. 10, pp. 1006–1021.
64. Briand L., Basili V. and Thomas W. (1992), "A pattern recognition approach for software engineering data analysis", *IEEE Trans. SE*, Vol. 18 No. 11, pp. 931–942.
65. Briand L. et al. (1999), "An assessment and Comparison of common software cost estimation modeling techniques", *Proc. International Conference on Software Engineering*, pp.313–322.
66. Chulani S., Boehm B. and Steece B. (1999), "Bayesian analysis of empirical software engineering cost models", *IEEE Trans. SE*, Vol. 25 No. 4, pp. 573–583.
67. Dolado J.J. (2001), "On the problem of the software cost function", *Information and Software Technology*, Vol.43 No.1, pp.61–72.
68. Shepperd M. and Schofield C. (1997), "Estimating software project effort using analogies", *IEEE Trans. SE*, Vol. 23 No. 12, pp. 736–743.
69. Vicinanza S., Prietulla M.J. and Mukhopadhyay T. (1990), "Case-based reasoning in software effort estimation", *Proc. 11th Intl. Conf. On Information Systems*, pp.149–158.
70. Kirsopp C., Shepperd M. J. and Hart J. (2002), "Search Heuristics, Case-based Reasoning And Software Project Effort Prediction", *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1367–1374.
71. Walkerden F. and Jeffrey R. (1999), "An empirical study of analogy-based software effort estimation", *Empirical Software Engineering*, Vol.4, pp.135–158.
72. Srinivasan K. and Fisher D. (1995), "Machine learning approaches to estimating software development effort", *IEEE Trans. SE*, Vol. 21 No. 2, pp. 126–137.
73. Heiat A., (2002), "Comparison of artificial neural network and regression models for estimating software development effort", *Information and Software Technology*, Vol.44 No. 15, pp.911–922.
74. Wittig G. and Finnie G. (1997), "Estimating software development effort with connectionist models", *Information and Software Technology*, Vol.39, pp.469–476.
75. Shukla K. (2000), "Neuro-genetic prediction of software development effort", *Information and Software Technology*, Vol.42 No.10, pp.701–713.
76. Lefley M. and Shepperd M. J. (2003), "Using genetic programming to improve software effort estimation based on general data sets", *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, pp.2477–2487.
77. Burgess C.J. and Lefley M. (2001), "Can genetic programming improve software effort estimation? a comparative evaluation", *Information and Software Technology*, Vol.43 No.14, pp.863–873.
78. Finnie G., Wittig G. and Desharnais J-M. (1997), "A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models", *Journal of Systems and Software*, Vol.39 No.3, pp.281–289.
79. Mair C., Kadoda G., Lefley M., Phalp K., Schofield C., Shepperd M. and Webster S. (2000), "An investigation of machine learning based prediction systems", *Journal of Systems and Software*, Vol.53 No.1, pp.23–29.
80. Jorgensen M. (1995), "Experience with the accuracy of software maintenance task effort prediction models", *IEEE Trans. SE*, Vol.21 No.8, pp.674–681.
81. Selby R. and Porter A. (1988), "Learning from examples: generation and evaluation of decision trees for software resource analysis," *IEEE Trans. SE*, Vol. 14, pp.1743–1757.
82. De Almeida M., Lounis H. and Melo W. (1998), "*Proc. International Conference on Software Engineering*", 1998, pp.473–476.
83. Mao Y., Sahraoui H. and Lounis H. (1998), "Reusability hypothesis verification using machine learning techniques: a case study", *Proc. 13th IEEE International Conference on Automated Software Engineering*, 1998, pp.84–93.
84. Dohi T., Nishio Y. and Osaki S. (1999), "Optimal software release scheduling based on artificial neural networks", *Annals of Software Engineering*, Vol.8 No.1, pp.167–185.
85. Khoshgoftaar T., Allen E. and Xu Z. (2000), "Predicting testability of program modules using a neural network", *Proc. IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, pp.57–62.
86. Stamelos I., Angelis L., Dimou P., Sakellaris E. (2003), "On the use of Bayesian belief networks for the prediction of software productivity", *Information and Software Technology*, Vol.45 No.1, pp.51–60.
87. Wegener J., Sthamer H., Jones B.F. and Eyres D.E. (1997), "Testing real-time systems

- using genetic algorithms", *Software Quality Journal*, Vol. 6. P.127–135.
88. Karunanithi N., Whitely D. and Malaiya Y. (1992), Prediction of software reliability using connectionist models. *IEEE Trans. SE*. Vol. 18, N 7. P. 563–574.
89. Yang B. and Xiang L. (2007) A study on software reliability prediction based on support vector machines. *International conference on industrial engineering and engineering management (IEEM)*. P. 1176–1180.
90. Xingguo L. and Yanhua S. (2007), An early prediction method of software reliability based on support vector machine. *International conference on wireless communications, networking and mobile computing (WiCom)*. P. 6075–6078.
91. Kumar P., Singh Y. (2012) An empirical study of software reliability prediction using machine learning techniques. *International Journal of System Assurance Engineering and Management (Int J Syst Assur Eng Manag)*. Vol. 3, N 3. P. 194–208.
92. Fenton N., Neil M. (1999) A critique of software defect prediction models. *IEEE Trans. SE*. Vol. 25, N 5. P. 675–689.
93. Langley P., Simon H. (1995), "Applications of machine learning and rule induction", *Communications of ACM*, Vol. 38. N 11. P. 55–64.

Получено 14.10.2019

Об авторах:

Мороз Ольга Григорьевна,
кандидат технических наук,
научный сотрудник.
Количество научных публикаций в
украинских изданиях – 27.
Количество научных публикаций в
зарубежных изданиях – 2.
Индекс Хирша – 2.
<https://orcid.org/0000-0002-0356-8780>,

Мороз Григорий Борисович,
кандидат технических наук,
старший научный сотрудник.
Количество научных публикаций в
украинских изданиях – 65.
Количество научных публикаций в
зарубежных изданиях – 4.
Индекс Хирша – 3.
<https://orcid.org/0000-0001-8666-9503>.

Место работы авторов:

Международный научно-учебный центр
информационных технологий и систем
НАН Украины и МОН Украины,
03680, г. Киев,
проспект Академика Глушкова, 40.
Тел.: (044) 502 6355.

Институт программных систем
НАН Украины,
03187, г. Киев,
проспект Академика Глушкова, 40,
корпус 5.
Тел.: (044) 526 33 09.

E-mail: olhahryhmoroz@gmail.com,
mgb@isofts.kiev.ua

Р.Д. Григорян, А.Г. Дегода, Е.А. Джуринський

ПРОГРАММНЫЙ СИМУЛЯТОР МЕХАНИЗМОВ ДОЛГОВРЕМЕННОЙ РЕГУЛЯЦИИ ГЕМОДИНАМИКИ ЧЕЛОВЕКА

Разработан компьютерный симулятор физиологических механизмов долговременной регуляции состояния сердечно-сосудистой системы человека. Основа компьютерного симулятора – это количественные математические модели: а) влияния центральной ренин-ангиотензиновой системы на характеристики сердечно-сосудистой системы; б) динамики общего объема крови. Тестовые исследования математической модели в автономном режиме показали адекватность моделей. Планируется объединить эту модель с созданной ранее моделью срочной регуляции сердечно-сосудистой системы. На основе комплексной модели будет разработана специализированная программная технология для поддержания компьютерных симуляций по разным аспектам физиологии кровообращения здорового человека. Специальное имитационное исследование должно выявлять причины разных сценариев развития гипертонической болезни. Программа написана на языке C⁺⁺.

Ключевые слова: математическая модель, физиология, кровообращение, артериальное давление, регуляция, информационная технология.

Введение

Сердечно-сосудистая система (ССС) человека – это открытая система, сообщающаяся с другими жидкими средами тела. Наряду с этим, состояние ССС находится под непрерывным влиянием ряда эндогенных нервно-рефлекторных и гуморальных механизмов. В результате этих эндогенных изменений гемодинамика человека проявляет быстрые или медленные изменения даже на фоне стабильности внешней среды. При изменениях физико-химических и информационных характеристик внешней среды, прямо или опосредованно изменяется и гемодинамика. До настоящего времени не понятно, как разные эндогенные механизмы взаимодействуют и создают характерную для здорового человека гемодинамику. Эта неопределенность порождает проблемы при диагностике множества патологий, прямо или косвенно отражающихся на состоянии ССС и на количественных проявлениях ее функций. Сведения о большей части биофизических и физиологических механизмов получено в экспериментах на животных.

Основной экспериментальный метод – упрощение комплексного механизма путем пресечений нервных путей или химического отключения отдельных гуморальных контуров у модельных животных. Понятно, что устоявшиеся пред-

ставления о том, как собственные и сопряженные механизмы ССС контролируют гемодинамику животных и человека, не охватывают весь спектр нормальных и патологических проявлений кровообращения [1–5]. Не последнюю роль в этом сыграло то, что система регуляторов ССС слишком сложна, а суперпозиция результатов, полученных в исследованиях на упрощенных экспериментальных моделях, неадекватно отражает реальное взаимодействие всех механизмов в интактном организме.

Эффективным дополнением к эмпирической базе знаний могут быть симуляции на количественных математических моделях [6–11]. Большинство моделей симулирует эффекты механизмов кратковременной [8–12] центральной гемодинамики человека. Механизмы долговременной регуляции включены в описание лишь в модели [13]. На протяжении многих лет в Институте программных систем НАН Украины создавались программные симуляторы физиологических систем человека на основе количественных математических моделях [14–15]. В частности, нами уже созданы мультикомпаратментальная модель *объекта регулирования* (гемодинамики в ССС, лишённой регуляторов) [16], и модель нервно-рефлекторных механизмов регуля-

ции состояния ССС [17]. Последняя модель позволила симулировать краткосрочную регуляцию гемодинамики человека. Для компьютерной симуляции комплексной регуляции гемодинамики с учетом механизмов и срочной, и долговременной регуляции ССС, необходимо эти модели дополнить моделью механизмов долговременного контроля состояния ССС.

Цель статьи – описать математическую модель механизмов долговременного контроля ССС, а также компьютерную программу, созданную для проведения тестовых симуляций функций моделируемых механизмов.

Механизмы долговременного контроля кровообращения

Формально, любой нейроэндокринный механизм, способный повлиять на общий объем крови $V_S(t)$, или на тонус сосудов, или на производительность сердца $Q(t)$, является потенциальным регулятором гемодинамики. Тонус сосудов в моделях представляется ненапряженным объемом $U(t)$, жесткостью $D(t)$ и гидравлическим сопротивлением $R(t)$.

Производительность сердца – функция частоты $F(t)$ и силы $K(t)$ сердечных сокращений, а также объемов правого и левого желудочков сердца $V_{VR}(t)$, $V_{VL}(t)$.

Кратковременные обратимые изменения $U(t)$, $D(t)$, $R(t)$, $F(t)$ и $K(t)$, не касающиеся анатомии этих звеньев ССС, подпадают под категорию *срочная регуляция*. Медленные и долговременные сдвиги в просвете и мышечном слое сосудов, в размерах $V_{VR}(t)$, $V_{VL}(t)$, в $V_S(t)$, фактически приводящие к параметрической (морфологической) перестройке ССС [18, 19], подпадают под категорию *долговременная регуляция*.

Важно помнить, что при обоих типах изменений характеристик ССС, значения $P_a(t)$ и $Q(t)$ модифицируются. Для анализа процессов долговременной регуляции гемодинамики, вместо кардиоциклических изменений вышеприведенных переменных ССС, используются усред-

ненные за каждый сердечный цикл значения (например, среднее артериальное давление

$$\bar{P}_a(t) = \int_{t-T_h}^t P_a(t) dt / T_h,$$

где T_h – длительность текущего сердечного цикла). Наряду с $\bar{P}_a(t)$, часто используется также средний за минуту (60 сек) поток крови на выходе сердца

$$\bar{Q}(t) = \int_{t-60}^t Q(t) dt / 60.$$

Моделирование механизмов, изменяющих характеристики ССС

Одним из основных эндогенных химических агентов, способных повлиять на долговременный уровень $\bar{P}_a(t)$, является вазоактивное вещество *ангиотензин II* ($A(t)$) [1, 5]. Он является продуктом переработки *ангиотензина I*, вырабатываемого под влиянием *ренина*. Первоначально считалось, что *ренин* продуцируется лишь в почках при снижении потока крови в почечных артериолах. Сейчас известно, что почти во всех органах снижение органного кровотока вызывает выделение *ренина* [20]. Физиологи различают центральную (почечную) и локальные (органные) ренин-ангиотензиновые системы (РАС) [5, 20]. В данной модели эти системы не будут дифференцированы.

Ангиотензин II сужает артериолы (увеличивая $R(t)$) и вызывает повышение $\bar{P}_a(t)$. Для моделирования этого физиологического механизма необходимо формализовать процесс продукции $A(t)$, связав динамику $A(t)$ с величинами объемной скорости крови в органах-продуцентах *ренина*.

Обозначим $v_1(t)$ и $v_2(t)$ скорости продукции и утилизации $A(t)$, соответственно. Тогда динамика $A(t)$ связана с $v_1(t)$ и $v_2(t)$ следующим дифференциальным уравнением:

$$\frac{dA}{dt} = v_1(t) - v_2(t). \quad (1)$$

Пока допустим, что изменения $\bar{P}_a(t)$ одинаковы во всех участках артерий. Тогда изменения количества *ренина* в крови пропорциональны его изменениям в почках. В статике между объемами $A(t)$ и *ренина* существует пропорциональная связь. Обозначим статическую величину $v_1(t)$ как $\bar{v}_1(t)$. Тогда динамику $v_1(t)$ свяжем с $\bar{v}_1(t)$ как:

$$T_r \frac{dv_1}{dt} = v_1(t) - \bar{v}_1(t), \quad (2)$$

где T_r – постоянная времени для скорости продукции *ренина*.

Формальную зависимость $\bar{v}_1(t)$ от величины почечного кровотока $q_r(t)$ можно аппроксимировать как:

$$\bar{v}_1(t) = \begin{cases} 0, & q_0 \leq q_r(t), \\ k_r \cdot (q_0 - q_r(t)), & q_{r \min} < q_r(t) \leq q_{r \max}, \\ k_r \cdot (q_0 - q_{r \min}), & q_r(t) \leq q_{r \min}. \end{cases} \quad (3)$$

Скорость утилизации $A_2(t)$ постоянна, т. е.

$$v_2(t) = c. \quad (3)$$

Объектом управления для предлагаемой модели долговременной регуляции гемодинамики является опубликованная ранее мультикомпарментальная модель неуправляемой ССС [14, 15]. В ней текущее значение гидравлического сопротивления сосудистого участка $R_i(t)$ зависит от исходного сопротивления R_i0 и от изменений соотношения текущего сосудистого объема $V_i(t)$ к его исходному значению V_i0 как:

$$R_i(t) = R_i0 \cdot \left(\frac{V_i0}{V_i(t)}\right)^2.$$

В этой формуле принято, что R_i0 соответствует значению $R_i(t)$ при

$V_i(t) = V_i0$, а уровень $A(t) = 0$. Для учета влияния *ангиотензина II* на $R_i(t)$ эту формулу модифицируем к виду:

$$R_i(t) = \begin{cases} R_{i \min}, & R_i(t) < R_{i \min}, \\ R_i0 \cdot \left(\frac{V_i0}{V_i(t)}\right)^2 \cdot (1 + \alpha_i \cdot A(t)), & R_{i \min} \leq R_i(t) \leq R_{i \max}, \\ R_{i \max}, & R_i(t) > R_{i \max}, \end{cases} \quad (4)$$

где константа α_i характеризует локальную чувствительность данного сосудистого участка к изменениям концентрации *ангиотензина II* в крови.

Известно [1, 5], что кроме *ангиотензина II*, вазоактивными свойствами обладают и другие эндогенные химические агенты (обозначим все эти агенты, включая и *ангиотензин II*, как $A_j(t)$, где $j = 1, 2, \dots, m$). Тогда для общего случая формулу (4) перепишем как:

$$R_i(t) = \begin{cases} R_{i \min}, & R_i(t) < R_{i \min}, \\ R_i0 \cdot \left(\frac{V_i0}{V_i(t)}\right)^2 \cdot (1 + \sum_j \alpha_j \cdot A_j(t)), & R_{i \min} \leq R_i(t) \leq R_{i \max}, \\ R_{i \max}, & R_i(t) > R_{i \max}. \end{cases} \quad (5)$$

Помимо гидравлического сопротивления, вазоактивные вещества изменяют жесткость $D_i(t)$ и ненапряженный объем $U_i(t)$ сосудов. Эти механизмы моделируем как:

$$D_i(t) = \begin{cases} D_{i \min}, & D_i(t) < D_{i \min}, \\ D_i0 \cdot (1 + \sum_j \beta_j \cdot A_j(t)), & D_{i \min} \leq D_i(t) \leq D_{i \max}, \\ D_{i \max}, & D_i(t) > D_{i \max}. \end{cases} \quad (6)$$

$$U_i(t) = \begin{cases} U_{i \min}, & U_i(t) < U_{i \min}, \\ U_i0 \cdot (1 - \sum_j \gamma_j \cdot A_j(t)), & U_{i \min} \leq U_i(t) \leq U_{i \max}, \\ U_{i \max}, & U_i(t) > U_{i \max}. \end{cases} \quad (7)$$

В последних формулах константы β_j и γ_j имеют тот же смысл, что и α_j в (5).

Ряд эндогенных химических агентов влияет на длительность сердечного цикла (T_{hc}) [1, 5]. Одни из них удлиняют T_{hc} , другие – укорачивают. Учитывая, что и симпатические, и парасимпатические влияния на T_{hc} также осуществляются посредством специальных медиаторов (норадреналина и ацетилхолина соответственно), в общем случае динамику $T_{hc}(t)$ можно связать с количеством ускоряющих ($M^-_{hc}(t)$) или тормозных ($M^+_{hc}(t)$) медиаторов как:

$$T_{hc}(t) = \begin{cases} T_{hc\min}, & T_{hc}(t) < T_{hc\min} \\ T^a_{hc} \cdot (1 - h^-_j \cdot M^-_{hc}(t)) \cdot (1 + h^+_j \cdot M^+_{hc}(t)) +, & \\ h^o \cdot (T^o - T^o_0), & T_{hc\min} \leq T_{hc}(t) \leq T_{hc\max} \\ T_{hc\max}, & T_{hc}(t) > T_{hc\max} \end{cases} \quad (8)$$

где T^a_{hc} – длительность сердечного цикла в режиме автоматизма сокращений сердца, T^o – температура крови, $T^o_0 = 36.6$, h^o – константа чувствительности $T_{hc}(t)$ к изменениям T^o , h^-_j – константы чувствительности $T_{hc}(t)$ к изменениям $M^-_{hc}(t)$, h^+_j – константы чувствительности $T_{hc}(t)$ к изменениям $M^+_{hc}(t)$.

Моделирование механизмов изменения общего объема крови

Известно [1, 2, 5], что на общий объем крови $V_S(t)$ влияют скорости пяти процессов: притока жидкости из пищеварительного тракта $v_w(t)$; транскапиллярной фильтрации жидкости $v_{cf}(t)$; диуреза $v_d(t)$; испарения с потом ($v_{es}(t)$) и выдыхаемым воздухом ($v_{ee}(t)$). На величину $v_{cf}(t)$ влияют площадь капиллярных пор и среднее капиллярное давление $\bar{P}_c(t)$

(зависит от $\bar{P}_a(t)$ и сопротивления артериол). $v_d(t)$ зависит от степени растяжения волюмрецепторов (расположены в левом предсердии и в головном мозге [1, 5]), от концентрации солей ($C_{be}(t)$) и липидов ($C_{bl}(t)$) в крови [5]. Связь между $V_S(t)$ и указанными переменными представим как:

$$\frac{dV_S}{dt} = \lambda_0 \cdot v_w(t) - \lambda_1 \cdot v_d(t) - \lambda_2 \cdot v_{es}(t) - \lambda_3 \cdot v_{ee}(t) - \lambda_4 \cdot v_{cf}(t) + \lambda_5 \cdot C_{be}(t) + \lambda_6 \cdot C_{bl}(t) \quad (9)$$

где $\lambda_1 - \lambda_5$ – константы чувствительности (их значения можно задавать через интерфейс пользователя).

$v_{es}(t)$ зависит от температуры крови $T^o(t)$, а $v_{ee}(t)$ – от легочной вентиляции ($v_L(t)$). Полагая эти зависимости линейными с коэффициентами λ_2^* , λ_3^* и λ_4^* соответственно, уравнение (9) перепишем как:

$$\frac{dV_S}{dt} = \lambda_0 \cdot v_w(t) - \lambda_1 \cdot v_d(t) - \lambda_2 \cdot (1 + \lambda_2^*) \cdot v_{es}(t) - \lambda_3 \cdot (1 + \lambda_3^*) \cdot v_{ee}(t) - \lambda_4 \cdot (1 + \lambda_4^*) \cdot v_{cf}(t) + \lambda_5 \cdot C_{be}(t) + \lambda_6 \cdot C_{bl}(t). \quad (10)$$

Для моделирования регуляции $V_S(t)$ посредством механизма, основанного на работе волюмрецепторов, необходимо устанавливать некое соответствие между интенсивностью импульсации $I_{Vr}(t)$ и $v_d(t)$. Но между этими переменными есть промежуточная – среднее трансмуральное давление в зоне расположения сердечных $\bar{P}_{Th}(t)$ (и/или внутричерепных $\bar{P}_{Tc}(t)$) рецепторов. По аналогии с моделью функционирования сосудистых механорецепторов [15], предложена модель, связывающая $I_{Vr}(t)$ с трансмуральным давлением как:

$$I_{Vrj}(t) = \begin{cases} 0, & \bar{P}_{Tj}(t) \leq P_{Tj}^{TH}, \\ \frac{1 - e^{-\delta_j(\bar{P}_{Tj}(t) - P_{Tj}^{TH})}}{1 + \chi_j \cdot e^{-\delta_j(\bar{P}_{Tj}(t) - P_{Tj}^{TH})}}, & P_{Tj}^{TH} < \bar{P}_{Tj}(t) < P_{Tj}^S, \\ 1, & \bar{P}_{Tj}(t) \geq P_{Tj}^S, \end{cases} \quad (11)$$

$j = \overline{1,2}$

где P_{Tj}^{TH} – пороги чувствительности рецепторов, P_{Tj}^S – уровни их давления насыщения, χ_j, δ_j – аппроксимирующие константы.

$$v_d(t) = \theta_1 \cdot I_{Vr1}(t) + \theta_2 \cdot I_{Vr2}(t), \quad (12)$$

где I_{Vr1}, I_{Vr2} – активности волюмрецепторов сердца и головы соответственно, а θ_1 и θ_2 – константы аппроксимации.

Вторая версия модели диуреза, вызванного активацией волюмрецепторов, связывает исходное сопротивление почечных артериол $R_r,0$ со значениями $I_{Vr1}(t)$ и $I_{Vr2}(t)$ как:

$$R_r,0 = \theta_1^* \cdot I_{Vr1}(t) + \theta_2^* \cdot I_{Vr2}(t). \quad (13)$$

В обоих вариантах модели детерминантом диуреза является почечный кровоток.

Модель влияния онкотических и осмотических характеристик фильтруемой почками крови на скорость реабсорбции в почечных канальцах будет разработана позже.

Уравнение (10) позволяет осуществлять компьютерные симуляции с разными наборами коэффициентов чувствительности. Иначе говоря, можно симитировать любые сценарии изменения $V_S(t)$ как входное возмущение и анализировать реакции системной гемодинамики на эти возмущения.

Таким образом, система уравнений (1) – (13) представляет собой автономную модель механизмов долговременной регу-

ляции гемодинамики человека. Но чтобы реально моделировать эти механизмы, необходимо эту модель сопрягать с моделями гемодинамики [14, 15].

Краткие сведения о программе

Программная реализация использует язык программирования C++17 для построения работы симулятора и его взаимодействия с пользователем. Эта технология была выбрана, поскольку C++ позволяет реализовывать системы с наименее возможным количеством накладных расходов от дополнительных абстракций, которыми обладают другие языки (например, Java и её виртуальная машина, сборщик мусора и т. д.). Однако, данная особенность языка усложняет и замедляет разработку системы, которая смогла бы реализовывать универсальную обработку различных видов моделей, а именно взаимодействие пользователя с этими моделями (модификация алгоритма обчёта, смена значений отдельных параметров и т. д.). Для решения данной проблемы разработана собственная объектная модель, позволяющая хранить дополнительную метаинформацию.

Объектная модель необходима для: корректного отображения данных на интерфейсе, взаимодействия пользователя с моделью и её параметрами, описания процесса записи и чтения модели с диска, процесса копирования, инициализации и т. д. Выбранная версия C++17 помогает в этом, поскольку в данной спецификации языка добавлены и улучшены возможности написания кода, который выполняется в compile-time, что значительно ускоряет запуск приложения, а в некоторых ситуациях ускоряет работу взаимодействия с объектной моделью в runtime. Однако объектная модель требует дополнительного описания классов и их методов, что значительно раздувает кодовую базу проекта. Для решения данной проблемы разрабатывается инструмент на языке Python, который по файлу описания класса в определённом формате (IDL – interface definition language), будет производить генерацию кода, необходимого для работы объектной модели, что значительно упростит и ускорит время написания подобного служебного кода, кроме

того, генерация кода поможет избежать случайных ошибок («опечаток»).

В качестве инструмента для построения пользовательского интерфейса был выбран фреймворк Qt5, позволяющий создавать программы для большинства существующих платформ и операционных систем. Однако он использовался исключительно как средство для создания пользовательского интерфейса и в будущем, при необходимости, его можно будет заменить другим инструментом.

Для отображения графиков зависимости параметров от времени используется `qcustomplot`. Данная библиотека – это довольно простое и легко интегрируемое решение. Но данное решение будет заменено другим, так как `qcustomplot` имеет жесткую привязку к Qt и не поддерживает отображение трёхмерных графиков.

Результаты тестовых исследований

На этом этапе построения специализированного программно-моделирующего комплекса (ПМК) цель тестовых исследований проверить, насколько формулы модели отражают реальные физиологические взаимосвязи, а также, при отсутствии требуемых эмпирических характеристик, оценить наиболее вероятные величины принятых аппроксимирующих констант.

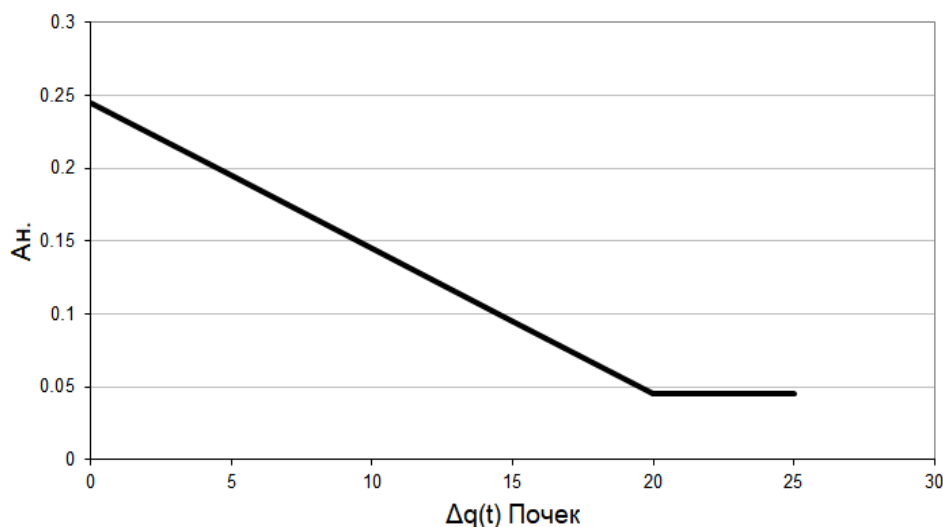


Рис. 1. Моделируемая зависимость концентрации ангиотензина II в крови (Ан в условных единицах) от уменьшения почечного кровотока от его исходного (принятого за норму в модели неуправляемой ССС) значения до некоторого минимума (здесь 5 мл./с)

В данной статье публикуется лишь часть результатов, полученных при тестировании модели долговременной регуляции состояния ССС. Тестирование проводилось на модели, в которой контуры барорефлекторной и хеморефлекторной регуляции гемодинамики были отключены. Также была отключена регуляция общего объема крови. Функционировала только модель ангиотензиновой регуляции сопротивления системных сосудов.

На рис. 1 показана зависимость концентрации ангиотензина II в крови от уменьшения почечного кровотока от его исходного (принятого за норму в модели неуправляемой ССС) значения до некоторого минимума (здесь 5 мл./с).

Обсуждение

Изображенные на рис. 1–3 результаты предварительного тестирования модели показывают, что уравнения правильно отражают направления изменений гемодинамических характеристик в ответ на дискретные изменения коэффициента чувствительности сосудов к концентрациям ангиотензина II. Пока об адекватности динамики речь не идет. Для исследования динамики необходима активация всех рефлекторных и неререфлекторных модуляторов гемодинамики. Эта работа предстоит, результаты будут опубликованы отдельно.

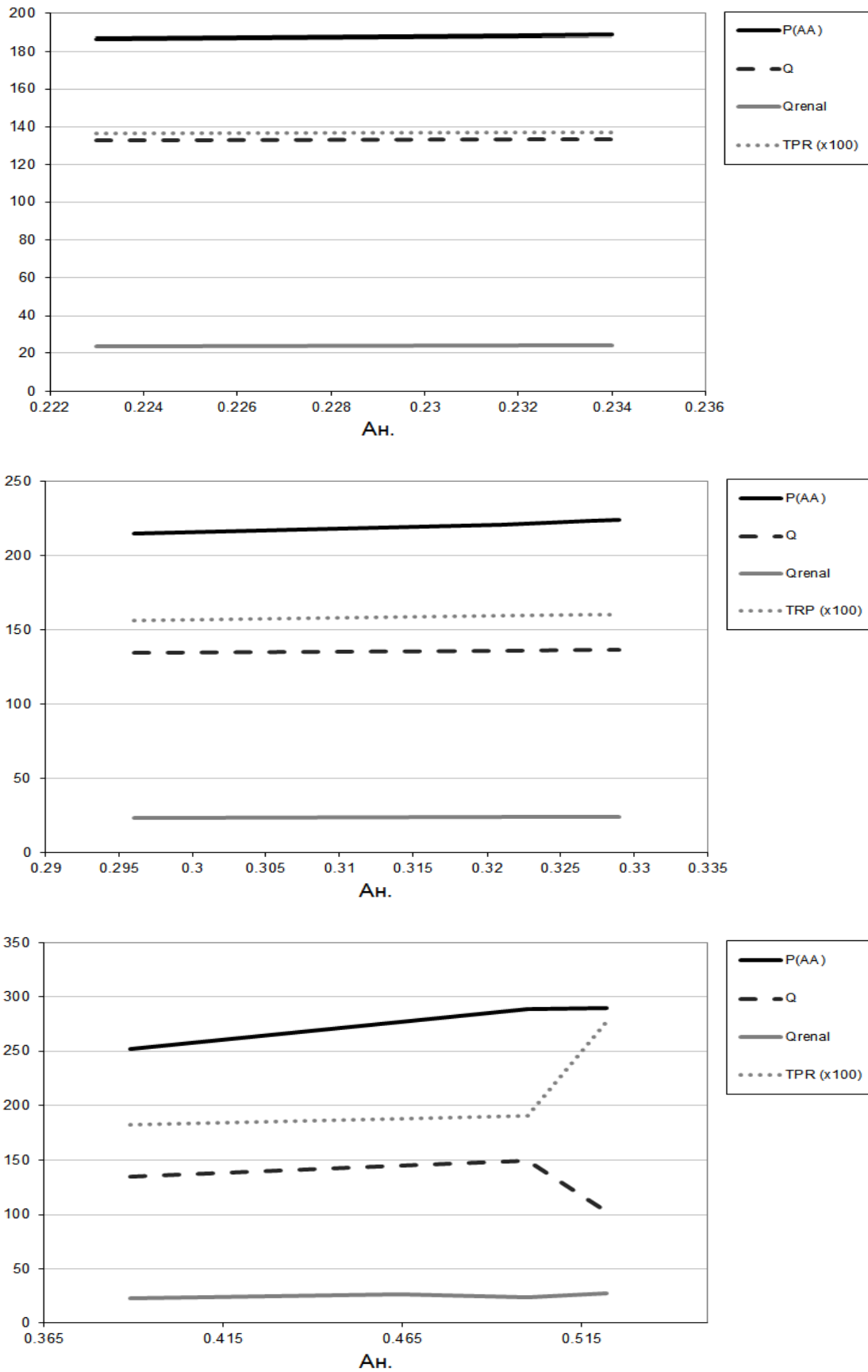


Рис. 2. Статические зависимости основных показателей центральной гемодинамики при трех значениях (возрастающих вверх-вниз) коэффициента α_i (отражающего чувствительность изменения сопротивления сосуда к изменению концентрации ангиотензина II в уравнении (4) (симуляция)

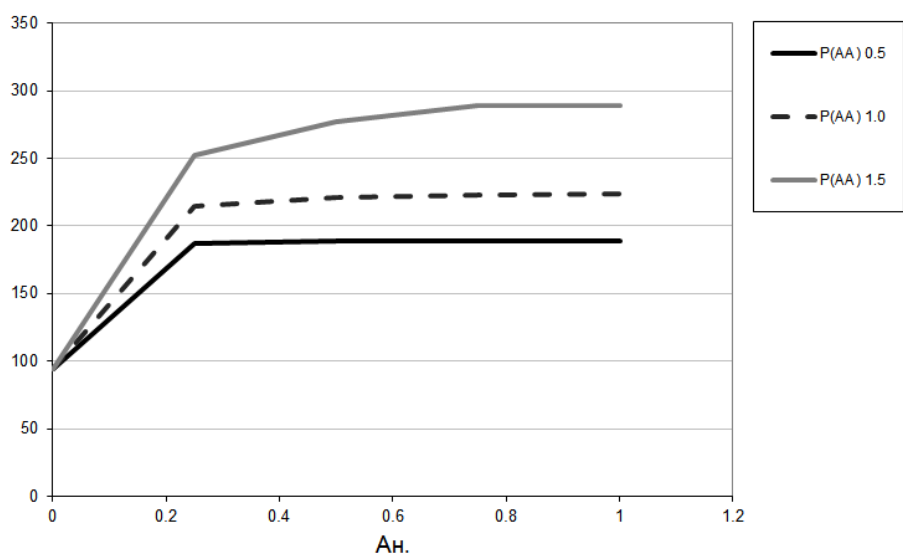


Рис. 3. Статические зависимости среднего давления в дуге аорты ($P(AA)$) от величины концентрации ангиотензина II при тех же трех значениях α_i (симуляция)

Выводы

Разработанная математическая модель долговременной регуляции гемодинамики человека в целом адекватно отражает реальные процессы в теле здорового человека. Для дальнейшего тестирования модели необходимо:

- разработать специальную программу, позволяющую интегрировать данную модель с моделями [14, 15] в единый ПМК;
- провести тестирование ПМК для нахождения наиболее оптимальных значений аппроксимирующих констант уравнений модели;
- исследовать адекватность ПМК в широких пределах изменений в характеристиках каждого моделируемого механизма;
- обозначить область потенциального применения ПМК.

Литература

1. Brands M.W. Chronic Blood Pressure Control. *Compr. Physiol.* 2012. Vol. 2. P. 2481–2494.
2. Grygoryan R.D. The optimal circulation: cells' contribution to arterial pressure. 2017. Nova Science. N.Y. 298 p.
3. Grygoryan R.D., Sagach V.F. The concept of physiological super-systems: New stage of integrative physiology. *Int. J. Physiol. and Pathophysiology.* 2018. 9.2. P. 169–180.
4. Grygoryan R.D. The unknown aspects of arterial pressure. *Znanstvena misel journal.* 2019. 33. P. 19–23.
5. Sparks M.A., Crowley S.D., Gurley S.B., Mirotsoy M., Coffman T.M. Classical Renin-Angiotensin system in kidney physiology. *Compr Physiol.* 2014. 4(3). P. 1201–1228. doi:10.1002/cphy.c130040.
6. Григорян Р.Д., Лиссов П.Н. Программный имитатор сердечно-сосудистой системы человека на основе ее математической модели. *Проблеми програмування.* 2004. № 4. С. 100–111.
7. Григорян Р.Д., Лиссов П.Н., Аксенова Т.В., Мороз А.Г. Специализированный программно-моделирующий комплекс «PhysiolResp». *Проблеми програмування.* 2009. № 2. С. 140–150.
8. Larrabide I., Blanco P.J., Urquiza S.A., Dari E.A., Ve'nere M.J., de Souza e Silva N.A., Feijo' R.A. HeMoLab – Hemodynamics Modelling Laboratory: An application for modelling the human cardiovascular system. *Computers in Biology and Medicine.* 2012. Vol. 42. P. 993–1004.
9. Ježe F., Kulhánek T., Kalecký K., Kofránek J. Lumped models of the cardiovascular system of various complexity. *Biocybernetics and*

- Biomedical Engineering*. 2017. Vol. 37, Issue 4. P. 666–678.
10. Kulhánek T., Kofránek J., Mateják M. Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object-oriented and acausal approach. *Computers in Biology and Medicine*. 2014. Vol. 54. P. 137–144.
 11. Mahdi A., Sturdy J., Ottesen J.T., Olufsen M.S. Modeling the Afferent Dynamics of the Baroreflex Control System. *PLoS Comput Biol*. 2013 Dec. 9(12). e1003384.
 12. Raphan T., Cohen B., Xiang Y., Yakushin S.B. A Model of Blood Pressure, Heart Rate, and Vaso-Vagal Responses Produced by Vestibulo-Sympathetic Activation. *Front Neurosci*. 2016. Vol. 10. P. 96–104.
 13. Fernandez de Canete J., Luque J., Barbancho J., Munoz V. Modelling of long-term and short-term mechanisms of arterial pressure control in the cardiovascular system: an object-oriented approach. *Comput Biol Med*. 2014. 47. P. 104–12. doi: 10.1016/j.combiomed.2014.01.006.
 14. Григорян Р.Д., Дегода А.Г., Джурицкий Е.А., Харсун В.С. Симулятор пульсирующего сердца. *Проблеми програмування*. 2017. № 4. С. 98–108.
 15. Григорян Р.Д., Дегода А.Г., Харсун В.С., Джурицкий Е.А. Симулятор механизмов срочной регуляции гемодинамики человека. *Проблеми програмування*. 2019, № 1. С. 90–98. doi: <https://doi.org/10.15407/pp2019.01.090>.
 16. Stewart J.M. Update on the theory and management of orthostatic intolerance and related syndromes in adolescents and children. *Expert Rev. Cardiovasc. Ther.* 2012. Vol. 10(11). P. 1387–1399.
 17. Kawada T., Yamamoto K., Kamiya A., Ariumi H., Michikami D., Shishido T., Sunagawa K., Sugimachi M. A derivative-sigmoidal model reproduces operating point-dependent baroreflex neural arc transfer characteristics. *Jpn. J. Physiol.* 2005. Vol. 55(3). P. 157–163.
 18. Fung P., Kong R. The Integrative Five-Fluid Circulation System in the Human Body. *Open J. of Mol. Integr. Physiol.* 2016. 6. P. 45–97. doi: 10.4236/ojmip.2016.64005.
 19. Grygoryan R.D. Comprehension of individual adaptation mechanisms: endogenous tuning of constants determining optimal physiological states. *Slovak int. scientific j.* 2019. 32. P. 67–72.
 20. De Mello WC, Frohlich ED. On the local cardiac renin angiotensin system. Basic and clinical implications. *Peptides*. 2011. 32. P. 1774–1779.

References

1. Brands M.W. Chronic Blood Pressure Control. *Compr. Physiol.*, 2012, V.2, P. 2481-2494.
2. Grygoryan R.D. The optimal circulation: cells' contribution to arterial pressure. 2017, Nova Science, N.Y., 298 p.
3. Grygoryan R.D., Sagach V.F. The concept of physiological super-systems: New stage of integrative physiology. *Int. J. Physiol. and Pathophysiology*, 2018: 9,2,169-180.
4. Grygoryan R.D. The unknown aspects of arterial pressure. *Znanstvena misel journal*, 2019, 33:19-23.
5. Sparks M.A., Crowley S.D., Gurley S.B., Mirotsoy M., Coffman T.M. Classical Renin-Angiotensin system in kidney physiology. *Compr Physiol.* 2014;4(3):1201–1228. doi:10.1002/cphy.c130040.
6. Grygoryan R.D., Lissov P.N. A software-simulator of human cardiovascular system based on its mathematical model. *Problems in programming*. 2004, №4. С.100-111 (Rus)..
7. Grygoryan R.D., Lissov P.N., Aksenova T.V., Moroz A.G. The specialized software-modeling complex “PhysiolResp”. *Problems of programming*, 2009, 2:140-150 (Rus).
8. Larrabide I., Blanco P.J., Urquiza S.A., Dari E.A., Ve'nere M.J., de Souza e Silva N.A., Feijo' R.A. HeMoLab – Hemodynamics Modelling Laboratory: An application for modelling the human cardiovascular system. *Computers in Biology and Medicine*. 2012, V. 42, P. 993–1004.
9. Ježe F., Kulhánek T., Kalecký K., Kofránek J. Lumped models of the cardiovascular system of various complexity. *Biocybernetics and Biomedical Engineering*. 2017, V. 37, Issue 4, P. 666-678.
10. Kulhánek T., Kofránek J., Mateják M. Modeling of short-term mechanism of arterial pressure control in the cardiovascular system: Object-oriented and acausal approach. *Computers in Biology and Medicine*, V. 54, 2014, P. 137–144.
11. Mahdi A., Sturdy J., Ottesen J.T., Olufsen M.S. Modeling the Afferent Dynamics of the Baroreflex Control System. *PLoS Comput Biol*. 2013 Dec; 9(12): e1003384.
12. Raphan T., Cohen B., Xiang Y., Yakushin S. B. A Model of Blood Pressure, Heart Rate,

- and Vaso-Vagal Responses Produced by Vestibulo-Sympathetic Activation. *Front Neurosci.* 2016, V.10, P. 96–104.
13. Fernandez de Canete J., Luque J., Barbancho J., Munoz V. Modelling of long-term and short-term mechanisms of arterial pressure control in the cardiovascular system: an object-oriented approach. *Comput Biol Med.* 2014;47:104-12. doi: 10.1016/j.combiomed.2014.01.006.
 14. Grygoryan R.D., Degoda A.G., Dzhurinsky Y.A., Kharsun V.S. A simulator of pulsatile heart. Симулятор пульсирующего сердца. *Problems of programming.* 2017, №4. С.98-108 (Rus.).
 15. Grygoryan R.D., Degoda A.G., Kharsun V.S., Dzhurinsky Y.A. A simulator of mechanisms of acute control of human hemodynamics. *Problems of programming,*2019;1:90-98. doi: <https://doi.org/10.15407/pp2019.01.090>. (Rus.)
 16. Stewart J.M. Update on the theory and management of orthostatic intolerance and related syndromes in adolescents and children. *Expert Rev. Cardiovasc. Ther.* 2012, V. 10(11), P. 1387–1399.
 17. Kawada T., Yamamoto K., Kamiya A., Ariumi H., Michikami D., Shishido T., Sunagawa K., Sugimachi M. A derivative-sigmoidal model reproduces operating point-dependent baroreflex neural arc transfer characteristics. *Jpn. J. Physiol.* 2005; V.55(3). P. 157–163.
 18. Fung P., Kong R. The Integrative Five-Fluid Circulation System in the Human Body. *Open J. of Mol. Integr. Physiol.* 2016; 6, 45–97. doi: 10.4236/ojmip.2016.64005.
 19. Grygoryan R.D. Comprehension of individual adaptation mechanisms: endogenous tuning of constants determining optimal physiological states. *Slovak int. scientific J.*, 2019, 32:67-72.
 20. De Mello W.C, Frohlich E.D. On the local cardiac renin angiotensin system. Basic and clinical implications. *Peptides.* 2011;32:1774–1779.

Об авторах:

Григорян Рафик Давидович,
заведующий отделом,
доктор биологических наук.
Количество научных публикаций в
украинских изданиях – 136.
Количество научных публикаций в
зарубежных изданиях – 45.
Индекс Хирша – 9.
<http://orcid.org/0000-0001-8762-733X>,

Дегода Анна Григорьевна,
старший научный сотрудник,
кандидат физико-математических наук.
Количество научных публикаций в
украинских изданиях – 13.
Количество научных публикаций в
зарубежных изданиях – 1.
Индекс Хирша – 3.
<http://orcid.org/0000-0001-6364-5568>,

Джуринский Егор Антонович,
инженер-программист.
Количество научных публикаций
в украинских изданиях – 3.
<http://orcid.org/0000-0002-1636-1447>.

Место работы авторов:

Институт программных систем
НАН Украины,
03187, Киев,
проспект Академика Глушкова, 40.
Тел.: (044) 526 5169.
E-mail: rgrygoryan@gmail.com,
anna@silverlinecrm.com,
y.a.dzhurynskyi@gmail.com

Получено 05.11.2019

К.І. Кузьміна, Т.М. Сьомик, Л.М. Шереметова

ІННОВАЦІЙНА ПЕДАГОГІЧНА КОМП'ЮТЕРНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ ШКІЛЬНОГО КОЛЕКТИВУ В АСПЕКТІ ЗНАТЬ ПРО БІОСОЦІАЛЬНУ СТРУКТУРУ ОСОБИСТОСТІ ЙОГО ЧЛЕНІВ

У роботі обговорюються сучасні комп'ютерні технології підтримки функціонування шкільного колективу на базі індивідуально-типологічного підходу для підвищення ефективності навчання і збереження біосоціального здоров'я його членів. З позиції принципів цілісності, системності, динамічності, ритмологічності і міждисциплінарності розглянуто соціопсихологічний динамічний портрет колективу, його індивідуальний адаптаційний інформаційний простір (ІАП) – раціональне для життєдіяльності середовище (дизайн). Технологічна схема пропонує для розгляду принципово нові сервісні компоненти: біосоціальна культура школяра (знання своїх адаптаційних можливостей), моніторинг для постійної гармонізації колективу, персонолог-міждисциплінарник, більшою мірою відповідальний за психологічний клімат середовища. Індивідуально-типологічний підхід для розуміння механізмів індивідуальної (а не середньостатистичної) адаптації людини, тандем «професійна компетенція – біосоціальна культура» разом з активною комп'ютеризацією можуть стати «проривним моментом» в сучасному людинознавстві і людиноведенні, тим самим істотно підвищуючи рівень економічного потенціалу і культури країни. Ключові слова: біосоціальна культура, індивідуально-типологічний підхід, здоров'язберігаюча комп'ютерна технологія, оптимізація шкільного колективу, психологічний клімат, профкомпетентність, персонолог-міждисциплінарник, моніторинг, топ-менеджмент, ефективність навчання.

1. Актуальність проблеми і мета роботи

Сучасна епоха – це час соціальних і природних потрясінь, що змінюють свідомість людей, час інтелектуальних революцій. Людина має продовжувати свій шлях від *homo erectus*, *homo sapiens* до *homo intelligence* – людини з високим рівнем біосоціальної культури [1 – 4], творчого, реалізаційного потенціалу, ефективності індивідуального і, особливо, колективного розуму. В основу останнього мають бути покладені такі високорангові людські якості: емпатія, взаємодія як мистецтво компромісу, тактичне і стратегічне мислення, високоінформативний інноваційний реалізаційний потенціал.

Науково-освітня сфера має бути престижною, а отримання нових знань – ключовим фактором для функціонування держави та її стратегічної безпеки. Надзвичайно актуальною сьогодні (і ми всі очевидці) є необхідність забезпечення високої якості професійної підготовки керівників лідерів. Інтелект має стати суспільним пріоритетом. Політикам необхідно це зрозуміти і підтримувати науку щедро і розумно.

Системоутворюючим ядром цілісності людини взагалі і компетентності вчителя, зокрема, є його професійний потенціал, пріоритети життєвих цінностей та ідеалів, біосоціальна культура, як очевидна платформа забезпечення високого рівня професійної компетенції [2, 5]. Педагогіка як наука про виховання, освіту і навчання сьогодні може базуватися на численних знаннях, у тому числі нових, про людину і людство.

Зокрема, це новий системний підхід до розуміння діючих в природі закономірностей, їх загального характеру – універсалізації принципу розвитку; погляд на світ як на єдине ціле, що складно розвивається, що підкоряється єдиним закономірностям (Г.Б. Лейбніц). Відповідно до його вчення про монади кожна проста субстанція є відображенням загальних якостей універсуму. При цьому розглядається новий стиль наукового мислення – нелінійність, біоритмологія, взаємозв'язок підсистем, їх кооперативний ефект, сила самоорганізації (І. Пригожин), можливість і необхідність міждисциплінарного соціопсихо-

фізіологічного підходу до вирішення, у тому числі, і педагогічних проблем, інформаційного арсеналу комп'ютерних засобів для їх реалізації у вигляді здоров'я-зберігаючих технологій [2, 4, 5].

Не менш важливі сьогодні для наукового світогляду сучасної теоретичної педагогіки і медицини, тісно з нею пов'язаної, здорова людина – це адекватно мотивована людина; знання законів взаємодії коливальних процесів, біологічної синхронізації як системоутворюючої функції у живій природі і її енергозабезпечення.

Новий термін «біофілософія» відкриває можливість формування цілісного образу живого з розумінням специфіки природно-біологічного в «соціальной людині», обумовлює творчу поведінку людини, гуманізацію її діяльності, гармонійні людські відносини [3].

Творчість розглядається як найбільш розвинена форма розвитку, як спосіб істинного гуманізму: від суспільства «трудящих» до суспільства «творящих», до людини соціально активної і соціально відповідальної [3]. Духовність і інтелегентність розглядаються як вектори дійсних гуманістичних і практичних реалізацій.

Звідси мета роботи – *розробка сучасної комп'ютерної технології підтримки функціонування шкільного колективу на базі індивідуально-типологічного підходу для підвищення ефективності навчання і збереження біосоціального здоров'я його членів.*

Концептуально розглядаються 3 блоки комплексних завдань:

- а) біосоціальна культура людини [5];
- б) індивідуально-типологічний адаптаційний профіль особистості і колективу в цілому [4];
- в) індивідуальний адаптаційний інформаційний простір – ІАП, реалізовані за допомогою комп'ютерних систем [1, 2, 4–7].

а) Біосоціальна культура людини

Поняття «біосоціальна культура» відображає пласт знань про цілісну організацію людини, її гено-фенотиповий базис, рівні самоорганізації та самосвідомості.

Це знання про себе, свої можливості, типологічні властивості (сила, рухли-

вість, врівноваженість нервових процесів), швидкісні властивості аналізаторів, тип реагування (стенічний, астенічний, гіпо- та гіперстенічний) і спрямованості захворювань (невротичний тип, психосоматичний і соціально-поведінковий); особливості вищих психічних функцій при переробці інформаційного потоку (увага, свідомість, мислення), стилі міжособистісної поведінки, професійний тропізм, способи біологічного захисту, інтроверсію-екстраверсію; про індивідуалізований корекційний підхід з використанням знань про структуру особистості при реалізації режимів праці і відпочинку.

б) Індивідуально-типологічний адаптаційний профіль особистості і колективу в цілому

Другий блок завдань відображає динаміку цілісного адаптаційного образу учня і шкільного колективу в цілому, **континуум його функціональних станів.** Останній діагностує рівень психоемоційного напруження, депресивних тенденцій, еустресу і дистресу, періодів оптимальної працездатності (сови-жайворонки), активометрії, психоклімату, стилів навчання та ін.

в) Індивідуальний адаптаційний інформаційний простір – ІАП

З урахуванням вище описаного технологія оптимізації педагогічного процесу, підвищення його ефективності з меншою психофізіологічною вартістю ще в більшій мірі знаходить відображення в **3-му** блоці завдань – **ІАП.** Цей термін сформульований нами і відображає комфортний, «добрий», раціональний для діяльності (навчання і виховання) простір (дизайн), побудований на основі знань про структуру особистості та функціональний стан того, хто навчається.

Поняття "дизайн" сьогодні – це не тільки вміння "оформити" предметно-просторове оснащення нашого життя, але це мистецтво нового етапу розвитку цивілізації [8]. Дизайн переріс у загальну категорію "проектну культуру" і затверджується як нова професія – дизайн середовища (колористична палітра, гармонізація форми, композиційне рішення структури сере-

довища, виразність і смисловий зміст елементів і т. д.).

Середовищний дизайн – єдиний вид проектування (зокрема, і при реалізації педагогічних завдань), який розглядає всю сукупність умов і обставин людського буття. Якщо на тлі цих знань розглянути знання про типи особистості людини (наприклад, холерик, флегматик, меланхолік, сангвінік), то визріє принципово **новий напрямок знань: "Типологія особистості – типологія середовищного дизайну"**. Сьогодні – це найбільш ємна сфера проектної діяльності, що має власну типологію своєї кінцевої продукції, свої методики і способи їх проектування.

У тандемі з типологією особистості людини (пацієнта), для якого потрібна ця продукція, платформа, що розглядається, генетично адекватна і перспективна для створення корисних високоінтелектуальних здоров'язберігаючих технологій, зокрема для реалізації у тому числі інноваційних програм з сімейної медицини [6] і електронної бібліотеки [9].

Створений інформаційний продукт (модель) для пацієнта, або так зване ІАП, повинен мати високий рівень комфортності та адекватності глибинній генотипічній організації людини з урахуванням особливостей її вищих психічних функцій, центрально-вегетативної регуляції, особливостей енергозабезпечення й інтоксикації організму.

Наприклад, на підставі цього для **інтроверта** (в тому числі і при комп'ютерному навчанні) – це гомогенне візуальне середовище з домінуванням ахроматичних кольорів у колористичній палітрі та інше.

Для **екстраверта** – все навпаки: різноманітний ілюстративний матеріал, динамічність агресивного (активного) середовища, еkleктичний (анархічний) дизайн. Такий індивідуальний інформаційний простір оптимізує, захищає від монотонності, підвищує рівень активації головного мозку екстраверта, що суб'єктивно сприймається ним як комфорт.

Для профілактики звикання, перенасичення і монотонії (особливо для користувача з сильною нервовою системою) не-

обхідно після певного терміну, не змінюючи основного змісту, вводити стенічні (активні) елементи.

В аспекті сказаного **особливо важливим є питання про етику**, основним закликом якого є «не нашкодь». Надзвичайно коректними необхідно бути з пацієнтами молодого віку, у яких багато інформаційних процесів протікають на підсвідомому рівні. Мають бути виключені неестетичні асоціації, які крім огиди нічого не викликають. Нині на TV часто зустрічаються немислимі блюзнірства, особливо для дітей емпатійних (співпереживаючих) інтровертів, формуючи неврози і нав'язливості, виражена перенасиченість інформаційних середовищ (особливо в період підкресленої необхідності концентрації уваги) динамічними колірними, контрастно-колірними вишукуваннями сучасних технічних можливостей без диференціації головного і другорядного.

2. Про структуру особистості і колективу

Сьогодні дуже потрібна грамотна мобілізація важливого **економічного ресурсу** – колективів, їх гармонійна структуризація, вдосконалення міжособистісних відносин, організація і керування трудовим колективом, підвищення рівня мотивації і ціннісних орієнтацій, тобто це проблематика гармонізації та функціонування соціуму на науково-обґрунтованій біосоціальной платформі.

В історії людинознавства за минуле сторіччя, практично за весь її науковий період тривали спроби узагальнити, а точніше, осмислити по-новому, звівши в єдине ціле досягнення багатьох дослідників особистості. При цьому важливі об'єктивні дані експериментальних досліджень і можливість «за деревами побачити ліс», цілісність людини в неосяжному світі психологічних побудов [1–5].

Психологи вважають, що саме існування людської цивілізації залежатиме від **біосоціальной грамотності людства і його розуміння самого себе і навколишнього середовища**.

При цьому перспективне міждисциплінарне вивчення поведінкових реакцій людини і їх забезпечення: гено-, фено-, фізіо-, біохімічне та інше, яке активно відходить від середньостатистичного підходу [5].

Час підготував об'єктивний ґрунт для систематизації, уніфікації різноаспектних біосоціальних знань про людину в теоретичному плані. А *міждисциплінарний лонгitudний соціопсихофізіологічний комп'ютерний експеримент*, заснований на гено-фенотипологічній платформі, привніс принципово нове в розуміння цілісності людини, раціональне і ефективне використання цієї платформи в усі періоди її життєвого циклу.

З численних варіантів визначення особистості провідними психологами сьогодні більше імпонує таке: Г. Олпорт (1937) "**Особистість** – це внутрішня динамічна організація психофізіологічних систем, яка формує патерни поведінки, мислення і емоцій, характерні для даної людини" [10].

Колектив розглядається як таке уособлення соціальних відносин, які забезпечуються динамічною організацією психофізіологічних груп індивідуумів і визначають унікальний спосіб адаптації до середовища [11].

Обговорення питань, пов'язаних з розумінням структури людини, спрямоване до знань про її *рисунки і типи* [10–12]. У наш час розглядаються 3 пласта знань: теоретико-емпіричні про основні положення про природу людини [10]; теоретико-експериментальні про нейрофізіологічні механізми, об'єктивні показники умовно-рефлекторної діяльності (школа Павлова, Теплова, Небілицина, Анохіна та ін.) [12–15].

На початку ХХ століття з'явилися роботи міждисциплінарного напрямку (соціоніка), в яких розглядаються можливості побудови психологічної моделі особистості для моделювання міжособистісних взаємин [16, 17], структурування психоінформаційного простору, розгляду типів інформаційного метаболізму колективів, організацій, націй, етносів, держав, їх психоінформаційної структури і взаємодії [16, 17 та інше]. Починаючи з 80-х років

минулого століття моделювання соціальних систем і суспільства з використанням типології Юнга [18] додало ряд надзвичайно необхідних знань у розуміння механізмів індивідуальної адаптації людини за допомогою вищих психічних функцій. Так існують чоловіча та жіноча складові нації та етносу і відповідно стереотипи їх «чоловічої» і «жіночої» поведінки; соціоніка як інтегральний напрям дозволяє побачити особливості аспектів інформаційного потоку, розглянути періодичну систему типів інформаційного метаболізму, голографічну модель функцій інформаційного метаболізму в зв'язку з асиметрією півкуль головного мозку, різні види інтелекту (іраціональний, раціональний), біоритмічні і енергетичні рівні людини та ін. [19].

Кожна риса людини представлена континуумом положень (або властивостей особистості) з двома екстремумами (наприклад, "свобода" розташована на одному полюсі першого континууму, а "детермінізм" – на іншому полюсі цього ж континууму, тип особистості може визначатися 2-ма, 3, 6, 8, 16 індивідуальними властивостями (Айзенк – 3, ММРІ – 13, Юнг – 8, Кеттел – 16, Павлов – 3, 12).

На ниві сучасної актуальності здорового глузду важливі роботи Адлера [10] про структуру особистості, який побудував свою типологію за принципом двомірної схеми «соціальний інтерес» і «ступінь активності», що збігається з поняттям «рівень енергії» і грає конструктивну або деструктивну роль тільки в поєднанні з соціальним інтересом. Соціальний інтерес, як барометр нормальності, показник якості життя. Еволюція змушує людей жити спільно з перспективою гармонійного суспільства, де кожна людина зобов'язана, має вміти грамотно, культурно спілкуватися з іншими; не тільки взаємодіяти, а й *взаємодіяти* з ними.

Особливо корисна для практики Адлерівська терапія – у співпраці, посиленні соціального інтересу. Сьогодні це когнітивна терапія – лікування знаннями. Юнг ближче всіх стояв до розуміння цілісності структури особистості з різним

ступенем вираженості її компонентів. Поведінка людини завжди є результатом тієї чи іншої *конфігурації особистісних рис* [10].

Структура особистості може бути описана за допомогою концепції *типу особистості* – сукупності безлічі різних рис, яка створює самостійну категорію з чітко визначеними межами. Айзенк [14] був переконаний, що для пояснення більшої частини поведінкових проявів людини необхідно не більше 3-х суперрис (які він назвав типами). Було виявлено 2 основних типи – інтроверсія-екстраверсія і нейротизм-стабільність, і чотири категорії людей. Найбільш важливим аспектом теорії Айзенка є його спроба встановити нейрофізіологічну канву в плані особливостей нервової системи.

Особистісний розвиток відбувається протягом усього життя. Важлива *стадійність* Еріксона, що підкріплюється відомою інвертованою V-подібною кривою адаптації, і вагомий внесок усвідомлюваного в самоорганізацію людини на основі вищих психічних механізмів: зворотної аферентації, самоактивації, волі, гальмівних нейрофізіологічних механізмів і ін.

За Келлі [10] (когнітивний напрямок) люди – думаючі істоти, людина по суті – вчений, дослідник, інтерпретатор того, що навколо неї.

Інша спрямованість робіт по структурі особистості, зокрема, вивчення темпераменту, ведуть нас до припущень Гіппократа (V ст. до н.е.) про відмінності людей по співвідношенню чотирьох основних "соків" організму – крові, флегми, жовтої та чорної жовчі з подальшим виділенням типів: *холерик, меланхолік, сангвінік і флегматик* (Клавдій Гален, II в н.е.). Подальші дослідження в області диференціальної психофізіології отримали відображення в працях І.П. Павлова, Б.М. Теплова, В.С. Мерліна, В.Д. Небиліцина. В основу індивідуальних відмінностей покладені фундаментальні властивості нервових процесів збудження і гальмування: сила, їх врівноваженість і рухливість.

Айзенк [14] вважає, що нейротизм, екстраверсія, інтроверсія обумовлені генетично, а крайні їх форми ведуть до патології.

За Собчик Л.Н. [20] надмірність показників стеничної саморегуляції загрожує дезадаптацією з *соціально-поведінкового типу*, протилежне співвідношення означає відмову від самореалізації і переважання соціального тиску на особу, формує *невротичний патерн дезадаптації*; при перенапруженні тих і інших механізмів, відбувається їх взаємне "гасіння" і формується *біологічний варіант дезадаптації* з психосоматичними розладами.

Гігієна розумової праці має передбачати, крім усього іншого, оптимальний харчовий режим і, в першу чергу, білковий. Дослідження інших харчових режимів – вуглеводних, мінеральних і жирових підтвердили їх причетність до *типологічної організації метаболічних процесів* в організмі і конституції людини [21].

Необхідно зазначити величезний внесок М.Н. Ліванова і його школи [22] в розшифровку феномена просторової синхронізації і її різних проявів. Надзвичайно важливо розглядати *синхронізацію як загальнобіологічну закономірність*, що лежить в основі часової самоорганізації процесів живої системи [22, 23].

Енергетичні характеристики систем досягають своїх екстремальних значень у стані синхронізації, відображаючи *механізм судової готовності і судової активності* субстрату [24].

А.Б. Коган [24] пов'язує такий режим (глобальної синхронізації, коли мозок працює як єдине ціле) з об'єднанням великих аналізаторних областей кори в ритмі "холостого ходу", коли структури гіперсинхронізовані, напружені, а вихід – безрезультатний.

Поняття *"біосоціальна культура"* людини [2] – це перш за все знання про себе, свої долереалізуючі тенденції, можливості, резерви, стійкість до стресу, вміння адекватно формувати міжособистісні відносини і багато іншого. За суттю – це знання свого міждисциплінарного соціопсихофізіологічного портрету [1], який суттєво розширює можливості пізнання людиною цього світу, що знижує ціну індивідуальної адаптації та веде до гідного довголіття [1, 2, 4, 5].

3. Розроблення підходу до створення інформаційної технології соціопсихофізіологічної підтримки для вивчення особливостей функціонування шкільного колективу

За суттю, уніфікація і систематизація знань з досліджуваного питання привела до розуміння того, що це цілісно-орієнтована «проривна» сучасна наукова технологія (методологія, методики і програма) вивчення індивідуально-типологічної адаптації школяра на базі індивідуально-типологічного підходу.

Далі (рисунок) показана схема технологічної реалізації проблеми: «Розроблення теоретичних основ для сучасної комп'ютеризованої організації трудового колективу», що досить повно й точно, науково-обґрунтовано відображає “образ” колективу для подальшого вивчення моделі за допомогою реалізованих у комп'ютерах алгоритмів. Це дає можливість глибоко і докладно вивчати об'єкти в динаміці і експерименті. Методологія математичного (а тепер вже й інформаційного) моделювання – це немінуча складова науково-технічного прогресу. Потрібні надійні засоби переробки інформаційної “сировини” – наші алгоритми в точні знання аналізу і прогнозу, прийняття рішень і контролю за їх виконанням.

Моделювання породжує чіткий план дій і умовно його можна розбити на три етапи: модель, алгоритм, програма.

1 етап – модель це теоретичний міждисциплінарний портрет колективу на основі знань про динамічний соціопсихофізіологічний портрет кожного члена колективу;

2 етап – вибір алгоритму для реалізації моделі на комп'ютері;

3 етап програми, що переведе модель і алгоритм на доступну комп'ютерну мову (“електронний” еквівалент об'єкта, що вивчається).

В даний час дозрів фундамент для *цілісного, системного, міждисциплінарного* розуміння людини, особливо професіонала і колективу в цілому на підставі **індивідуально-типологічного** (а не

середньостатистичного) підходу, з такими високоранговими якостями, як емпатія (почуття співпереживання), мистецтво компромісу і взаємодії, стратегічне і тактичне мислення, раціональність, швидкість, творчий, реалізаційний і контролюючий потенціал.

Дана технологічна схема включає також особливо важливий фрагмент для реалізації «цілісності» колективу – його динамічну структуру (фази зростання, розвитку, перенапруження, в цілому довголіття).

За суттю у роботі розглядається орієнтація на принципово нову науково-обґрунтовану реалізацію цілей у різних аспектах сучасного людинознавства і людиноведення, що забезпечують **тандем високої професійної компетентності колективів і їх біосоціального здоров'я.**

Соціопсихофізіологічний портрет старшокласника виступає **міждисциплінарним** (соціопсихофізіологічним) динамічним портретом особистості з урахуванням його професійної спрямованості, ділових якостей, умінь, ступеня його соціально-психологічної адаптованості. На підставі цього можна охарактеризувати функціонування цілого класу та спрямованість його біосоціального здоров'я.

Психофізіологічне дослідження дозволяє диференційовано визначати сумісність учнів, згуртованість колективу, здатність до ефективної взаємодії, деталізувати взаємини членів групи, ставлення до лідера.

Економічна вигода (як кінцевий показник) в класі може рекламуватися у вигляді високої і якісної успішності, адекватної адаптації у колективі і житті, високого рівня біосоціальної культури школярів.

Комп'ютерні **здоров'язберігаючі** системи – це міждисциплінарна основа для розвитку нових уявлень про теорію особистості, норму, довголіття, спрямованість дезадаптації (захворювань), адекватність відображення інформації.

Це гармонізація людино-комп'ютерних взаємин, розробка комп'ютерів нових поколінь, розгляд механізмів «соціальної психіки» на основі симбіозу з новими науками (психогенетика, біоритмологія, біоісторія, біоекономіка, біофілософія, біополітика та інше).

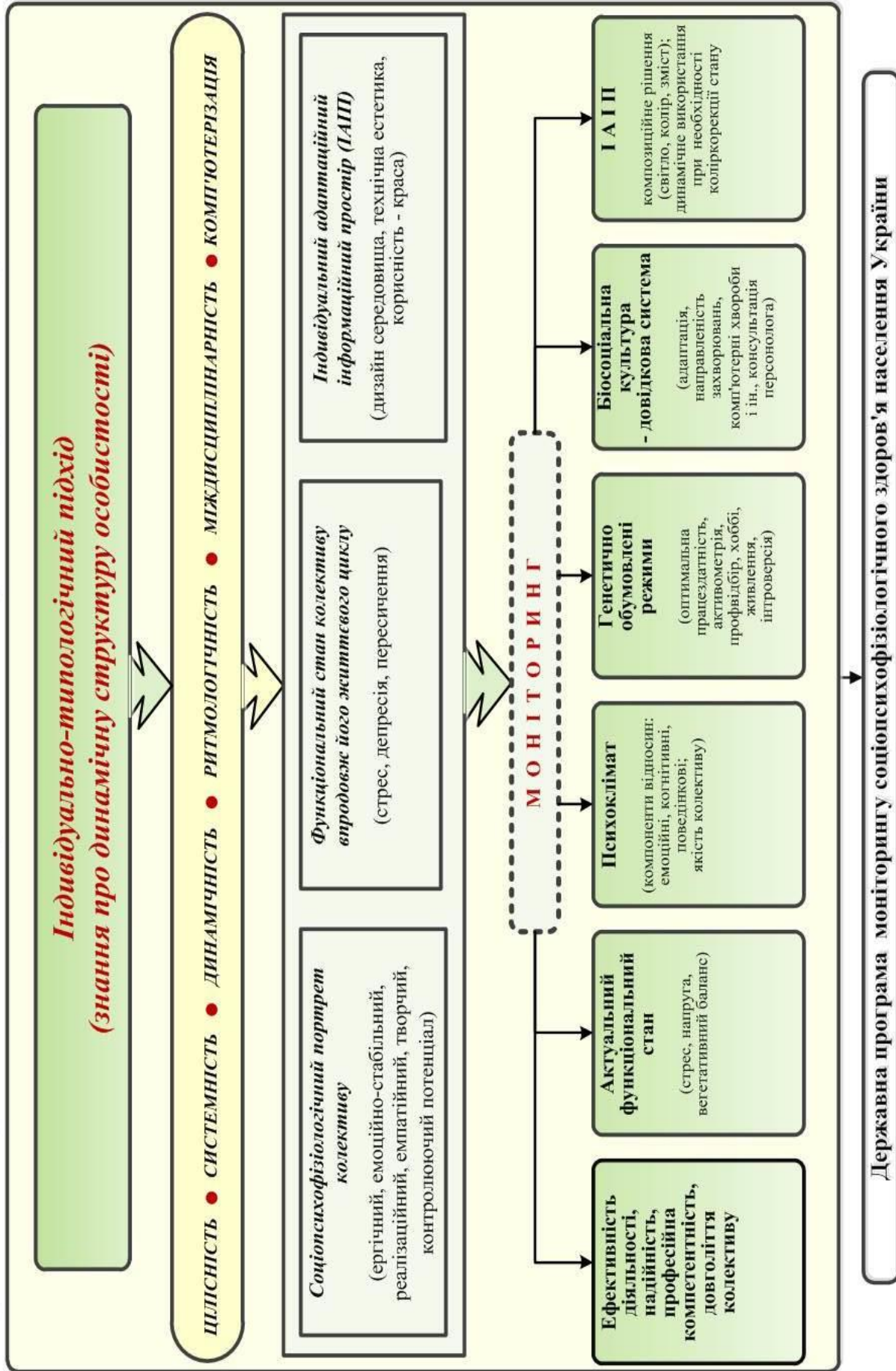


Рисунок. Розроблення теоретичних основ для сучасної комп'ютеризованої організації трудового колективу

Персонологія дуже молода галузь знань, продукт XIX століття, вже досягла повноліття, заявивши про себе як життєздатна генералізована галузь всеаспектних досліджень.

Індивідуально-типологічний підхід з урахуванням вищеперерахованих принципів і широкої комп'ютеризації може бути **«проривним»** етапом у плані вирішення 2-х найважливіших сьогодні проблем – *«професійної компетентності» людства і його біосоціальної культури»*.

Наступні 3 блоки завдань при роботі теоретичних основ для сучасної комп'ютеризованої організації трудового колективу – *соціопсихофізіологічний портрет колективу, функціональний стан колективу впродовж його життєвого циклу, індивідуальний адаптаційний інформаційний простір (ІАП)*.

Перший блок визначає ергічний, емоціолабільний, реалізаційний, емпатійний, творчий, контролюючий потенціал.

Другий блок – різні види функціональних станів колективу впродовж його життєвого циклу.

Третій блок – формування індивідуального адаптаційного інформаційного простору (ІАП), що несе концептуально нові знання (дизайн середовища, технічна естетика, корисність-краса) про гармонізацію професійної компетентності і біосоціального здоров'я.

Далі розглядається практична реалізація проблеми у вигляді принципово інноваційного блоку в таких системах – «блок моніторингу». Останній включає ряд інформаційноємних показників ефективності діяльності колективу (і кожного його члена) у зв'язку з актуальним станом його психофізіологічного здоров'я з метою їх оптимізації. Такими показниками виступають ефективність діяльності, рівень стресу, психоклімат і ряд генетично обумовлених режимів оптимальної працездатності (активоμετρία, профвідбір, хобі, харчування, інтроверсія), від яких суттєво залежить успішність діяльності і її адекватна «вартість» (рисунок).

Одним з інноваційних, високо наукових для адекватної реалізації даної системи є рівень *біосоціальної культури* (реа-

лізований у вигляді довідкової системи), яка відображує високу грамотність кожної людини, знання своїх резервних можливостей, поведінки в стресі, вміння і т. д.; також представлена інформація про особливості індивідуальної адаптації, спрямованість захворювань, комп'ютерні хвороби. Передбачаються постійні консультації з персонологом-міждисциплінарником (соціопсихофізіологом) в плані глибокого розуміння біосоціальної культури та механізмів індивідуальної адаптації (природний сон, спілкування, харчування).

Науково обґрунтованим, високоактуальним для забезпечення необхідної професійної і соціальної адекватності «комп'ютеризованого» колективу виступає заключний (рисунок) фрагмент ІАП, який здійснює композиційні рішення простору (світло, колір, зміст); можливе динамічне використання при необхідності коліркорекції стану та інше.

За суттю комп'ютеризована організація колективу безпосередньо пов'язана з давно відомою проблемою розробки дружнього інтерфейсу. В даний час це переконливий технічний засіб для спілкування з користувачем, хоча людина представлена в ньому вкрай слабо.

Розглянута технологія може стати цінним пластом психофізіологічних знань при побудові для користувачів індивідуалізованих моделей ПК і дружніх інтерфейсів з високим рівнем комфортності і адекватності їх глибинній гено-фенотипічній організації.

Розглянута програма з створення теоретичних основ для сучасної комп'ютеризованої організації трудового колективу (шкільного) може стати **інноваційною науково обґрунтованою інформацією для розробки інвестиційних державних програм моніторингу біосоціального здоров'я населення України**.

Висновки

Виходячи з того, що особистість є структурно-функціональною одиницею нації, знання про типологію особистості та її динамічну структуру формує принципово новий *антропосоціальний погляд на керування суспільством*, виходячи з його гено-

фенотипових властивостей [17–19].

Тому енергію індивіда необхідно направити в потрібне, соціально прийнятне русло. Таким руслом, самим головним керуючим фактором, джерелом життєстійкості людини і нації є **праця**.

Індивідуально-типологічний підхід до пізнання людини має розглядатися в руслі державної програми охорони здоров'я, і особливо здоров'я здорової людини, і орієнтувати на підвищення рівня біосоціальної культури нації, її морально-етичного та економічного потенціалу.

Технологія базується на **глибоких наукових міждисциплінарних засадах** вивчення адаптації людини в ХХІ столітті, кричущій необхідності актуальних реалізацій продуктів комп'ютерної біоінформатики. **Біосоціальна культура** (знання про себе, свої індивідуально-типологічні адаптаційні можливості) має стати пріоритетною **компетенцією людини ХХІ століття**.

Централізований підхід, який задав би правильний тон і реалізував науково-обґрунтовану методологію сьогодні просто необхідний як «проривний» для раціональної організації колективів різних рангів, їх ефективності та довголіття.

Є надія, що достатня зрілість міждисциплінарних знань про людину, динамічну структуру її особистості, індивідуальний адаптаційний інформаційний простір (ІАП), готовність сучасних **здоров'язберігаючих технологій і** комп'ютерних систем зорієнтують дослідників і керуючі ланки країн на адекватні програми з людинознавства і людиноведення, зокрема, створення науково-практичних центрів біосоціальної культури людини, сімейної медицини, професійної компетентності для вищих ешелонів керування, центрів гармонізації колективів різних рангів та інше.

Насьогодні закладено фундамент для **цілісного, міждисциплінарного** розуміння людини на основі **індивідуально-типологічного** (а не середньостатистичного) підходу.

Література

- Кузьмина К.И. Психологические механизмы индивидуальной адаптации организма при действии различных экзогенных факторов: дис. докт. биол. наук. Киев, 1995. 356 с.
- Кузьмина К.И. Человек и его биосоциальная культура. *Теория и практика управления социальными системами: философия, психология, педагогика, социология*. Харків. 2005. № 1. С. 109 – 120.
- Новіков Б.В. Творчість як спосіб здійснення гуманізму. Монографія. К.: Видавець ПАРАПАН, 2005. 332 с.
- Кузьміна К.І., Сьомик Т.М., Куниця Т.А. Особливості дослідження психологічного клімату в колективі. Глава 5. *Коллективная монография «Перспективные тренды развития науки: образование и воспитание, спорт»*. Входит в РИНЦ SCIENCE INDEX Одесса: Куприенко СВ. 2016. С. 115 – 128.
- Кузьмина К.И., Семик Т.М. Здоровье обеспечивающие технологии и компьютерные системы в современном человековедении для рациональной организации жизненного цикла человека. *Проблемы програмування*. 2017. № 3. С. 172 – 193.
- Кузьмина К.И., Оноприенко В.Н., Козак Н.С., Семик Т.М., Андон Т.А. Семейная медицина сегодня и проблема ее дальнейшей интеллектуализации с помощью информационных технологий и компьютерных систем. *Теория и практика управления социальными системами*. Харків: НТУ «ХПИ». 2012. № 2. С. 56 – 67.
- Кузьмина К.И., Семик Т.М., Андон Т.А. Современные информационные технологии для изучения механизмов индивидуальной психофизиологической адаптации человека. *Проблемы програмування*. 2008. № 2-3. С. 695 – 702.
- Дизайн. Иллюстрированный словарь-справочник / Минервин Г.Б., Шимко В.Г., Ефимов А.В. и др./под общ. ред. Б. Минервина, В.Г. Шимко. М.: «Архитектура. – С». 2004. 228 с.
- Кузьмина К.И., Семик Т.М., Андон Т.А. Социопсихофизиологические знания – инновационный резерв повышения качества электронных библиотек (ЭБ). *«Инновации в науке»*: материалы XII Международной заочной научно-практической конференции. Часть I. (17 сентября 2012 г.). Новосибирск: Изд. «Сибирская ассоциация консультантов», 2012. С. 31 – 43.
- Хьелл Л., Зиглер Д. Теории личности. 3-е изд. СПб: Питер, 2013. 607 с. (Серия «Мастера психологии»).
- Копець Л.В. Психологія особистості: Навч. посіб. для студентів вищ. навч. закл. – 2-е

- вид. К.: Вид. дім «Києво-Могилянська академія». 2008. 458 с.
12. Теплов Б.М. Новые данные по изучению свойств нервной системы человека. Типологические особенности высшей нервной деятельности человека. М.: Медицина, 1963. Т. 3. С. 34 – 46.
 13. Небылицын В.Д. Психофизиологические исследования индивидуальных различий. М.: Наука. 1976. 336 с.
 14. Eysenck H.S. The biological basis of personality. Illinois: Springfield. 1967. 202 p.
 15. Strelay J. Temperament I typ uklady nerwowego. Warszawa: Panstwowe Wydawnictwo Naukowe, 1974. 231s.
 16. Букалов А.В. Интегральная соционика, типы коллективов, государств, этносоционика. *Соционика, ментология и психология личности*. Киев. 1998. № 5. С. 13 – 17.
 17. Аугустинавичюте А. О дуальной природе человека. Международный ин-т соционики. *Соционика, ментология и психология личности*. Киев. 1996. № 2 (5). С. 2 – 12.
 18. Юнг К. Психологические типы. М.: «Университетская книга» АСТ, 1997. 709 с.
 19. Букалов А.В. Голографическая модель ФИМ и ее связь с полушариями головного мозга. *Соционика, ментология и психология*. Киев. 2001. № 3. С. 7 – 11.
 20. Собчик Л.Н. Психология индивидуальности. *Теория и практика психодиагностики*. СПб: изд. “Речь”. 2008. 624 с.
 21. Губко А.Т. Проблема типа нервной системы: научное исследование. Ин-т психологии им. Г.С. Костюка АПН Украины. К.: Світогляд. 2008. 300 с.
 22. Ливанов М.Н. Пространственная организация процессов головного мозга. М.: Наука. 1972. 252 с.
 23. Степанова С.И. Биоритмологические аспекты проблемы адаптации. М.: Наука. 1986. 244 с.
 24. Коган А.Б. О принципах нейрональной организации рабочих механизмов управления функциональной системой. *Принципы системной организации функций*. М.: Наука. 1973. С. 125 – 130.
 - social systems: philosophy, psychology, pedagogy, sociology. Kharkiv. 2005. N 1. P.109 – 120.
 3. Novikov B.V. Creativity as a way of realizing humanism. Monograph. K.: Publisher PARAPAN, 2005. 332 p.
 4. Kuzmina K.I., Syomik T.M., Kunitsya T.A. Features of the study of psychological climate in the collective. Chapter 5. The collective monograph "Perspective trends in the development of science: education and upbringing, sport". Included in the RSCI SCIENCE INDEX Odessa: Kuprienko SV. 2016. P. 115 – 128.
 5. Kuzmina K.I., Syomik T.M. Health-saving technologies and computer systems in modern human science for the rational organization of the human life cycle. *Problems of Software*. 2017. N 3. P. 172 – 193.
 6. Kuzmina K.I., Onoprienko V.N., Kozak N.S., Syomik T.M., Andon T.A. Family medicine today and the problem of its further intellectualization with the help of information technologies and computer systems. *Theory and Practice of Social Systems Management*. Kharkiv: NTU "KhPI", 2012. N 2. P. 56–67.
 7. Kuzmina K.I., Syomik T.M., Andon T.A. Modern information technologies for studying the mechanisms of individual psycho-physiological adaptation of a person. *Problems in Programming*. 2008. N 2-3. P. 695–702.
 8. Design. Illustrated dictionary-reference book / Minervin G.B., Shimko V.G., Efimov A.V. and others / under total. ed. B. Minervina, V.G. Shimko. M.: “Architecture. – C”, 2004. 228 p.
 9. Kuzmina K.I., Syomik T.M., Andon T.A. Sociopsychophysiological knowledge – innovative reserve for improving the quality of electronic libraries (EL). "Innovations in Science": materials of the XII International Correspondence Scientific and Practical Conference. Part I. (September 17, 2012). Novosibirsk: Izd. “Siberian Association of Consultants”, 2012. P. 31–43.
 10. Hyeell L., Ziegler D. Personality theory. 3rd ed. SPb: Peter, 2013. 607 p. (A series of "Masters of Psychology").
 11. Kopets L.V. Psychology specialty: A manual for students in higher education – 2nd edition. K.: Publishing House. “Kyiv- Mogilyan Academy”, 2008, 458 p.
 12. Teplov B.M. New data on the study of the properties of the human nervous system. Typological features of the higher nervous activity of human. М.: Медицина, 1963. Vol. 3. P. 34 – 46.

References

1. Kuzmina K.I. Psychophysiological mechanisms of individual adaptation of the organism under the action of various exogenous factors: dis. Dr. biol. sciences. Kiev, 1995. 356 p.
2. Kuzmina K.I. Human and his biosocial culture. Theory and practice of management of

13. Nebylytsyn V.D. Psychophysiological studies of individual differences. M.: Science, 1976. 336 p.
14. Eysenck H.S. The biological basis of personality. Illinois: Springfield, 1967. 202 p.
15. Strelay J. Temperament I typ uklady nerwowego. Warszawa: Panstwowe Wudawnictwo Naukowe, 1974. 231 p.
16. Bukalov A.V. Integral Socionics, types of groups, states, ethnoscionics. Socionics, mentology and personality psychology. Kiev. 1998. N 5. P. 13 – 17.
17. Augustinavichyute A. About the dual nature of human. International Institute of Socionics. Socionics, mentology and personality psychology. Kiev. 1996. N 2 (5). P. 2 – 12.
18. Jung K. Psychological types / K. Jung. M.: “University Book” AST, 1997. 709 p.
19. Bukalov A.V. Holographic model of FIM and its connection with the cerebral hemispheres. Socionics, Mentology and Psychology. Kiev. 2001. N 3. P. 7 – 11.
20. Sobchik L.N. Psychology of individuality. Theory and practice of psychodiagnostics. SPb: ed. “Speech”, 2008. 624 p.
21. Gubko A.T. The problem of the type of the nervous system: a scientific study. Institute of Psychology. G.S. Kostyuk APS of Ukraine. K.: Svitoglyad, 2008. 300 p.
22. Livanov M.N. Spatial organization of brain processes. M.: Science, 1972. 252 p.
23. Stepanova S.I. Biorhythmic aspects of the problem of adaptation. M.: Science, 1986. 244 p.
24. Kogan A.B. On the principles of neuronal organization of working mechanisms for managing a functional system. Principles of system organization of functions. M.: Science, 1973. P.125 – 130.

Про авторів:

Кузьміна Клариса Іванівна,
доктор біологічних наук,
провідний науковий співробітник
Кількість наукових публікацій в
українських виданнях – понад 130.
Кількість наукових публікацій в
зарубіжних виданнях – понад 10,

Сьомик Тетяна Михайлівна,
кандидат біологічних наук,
старший науковий співробітник
Кількість наукових публікацій в
українських виданнях – понад 100.
Кількість наукових публікацій
в зарубіжних виданнях – понад 6,

Шереметова Людмила Миколаївна,
вчителька.

Місце роботи авторів:

Інститут програмних систем НАН України,
03187, м. Київ,
проспект Академіка Глушкова, 40.
Тел.: (044) 524 26 70,
(044) 526 31 83.

Школа № 80,
01103 м. Київ,
бульвар Дружби Народів, 12-Б
E-mail: somik@isofts.kiev.ua

Одержано 18.07.2019

УДК 681.3

Модель паралельних програм з обмеженням ресурсів / Д.В. Рагозін. – С. 3 – 10.

Сучасні паралельні програми виконуються в складних та ресурсно-обмежених середовищах, тому до них застосовуються підвищені вимоги щодо споживання ресурсів та стабільності виконання, особливо для довгих у часі обчислень. Для допомоги у визначенні ресурсних обмежень для паралельних програм розроблено формальну модель паралельних програм з ресурсними обмеженнями. Модель визначає часові та ресурсні обмеження і може бути використана як для дрібнозернистого, так і для крупнозернистого паралелізму. Для високих ступенів паралелізму (незалежні процедури, великі ітерації циклів, великі обчислювальні блоки для графічних, нейромережових обчислень та обробки відео) інтерпретація моделі можлива під час виконання програми для унеможливлення блокувань при розподілі ресурсів. Розглянуто сучасні фреймворки, які можуть інтегрувати у свій склад базові засоби перевірки відповідності множини паралельних програм наявним обчислювальним ресурсам під час виконання програм. Розглянуто задачі реального світу – вивід у нейромережах, обробка відео, обчислення на графічних картах, які матимуть зиск з впровадження таких моделей.

Ключові слова: формальна модель, паралельні обчислення, паралельні обчислення на графічних процесорах.

УДК 681.3

Модель паралельних програм с ограничением ресурсов / Д.В. Рагозин. – С. 3 – 10.

Современные параллельные программы исполняются в сложных и ресурсно-ограниченных средах, поэтому к ним предъявляются повышенные требования к потреблению ресурсов и стабильности выполнения, особенно для долгих по времени вычислений. Для помощи в определении ресурсных ограничений для параллельных программ разработана формальная модель параллельных программ с ресурсными ограничениями. Модель определяет временные и ресурсные ограничения и может быть использована для как мелкозернистого, так и для крупнозернистого параллелизма. Для высоких степеней параллелизма (независимые процедуры, большие итерации циклов, большие вычислительные блоки для графических, нейросетевых вычислений и обработки видео) интерпретация модели возможна во время исполнения программы для обеспечения невозможности блокировок при распределении ресурсов. Рассмотрены современные фреймворки, для которых возможна интеграция в свой состав базовых средств проверки соответствия множества параллельных программ наличным вычислительным ресурсам при исполнении программы. Рассмотрены задачи реального мира – вывод в нейросетях, обработки видео, вычисления на графических картах, для которых будет выгодно применение таких моделей.

Ключевые слова: формальная модель, параллельные вычисления, параллельные вычисления на графических процессорах.

Использование онтологий для расширения возможностей поиска и навигации в онлайн-версии «Большой украинской энциклопедии» / Ю.В. Рогушина, И.Ю. Гришанова. – С. 11 – 27.

Цель работы – анализ средств семантического поиска в Wiki-ресурсах и разработка направлений расширения его возможностей на основе использования современных средств управления знаниями. В статье анализируются преимущества, которые обеспечивают пользователям использование семантической разметки в информационных ресурсах Web для более удобного и быстрого доступа к контенту. Рассматриваются способы использования онтологий для поиска и навигации в семантизированных Wiki-ресурсах. Приведено определение Wiki-онтологии, которая формализует информацию о структуре базы знаний информационного ресурса (как обычного, так и семантизированного), который базируется на технологии Wiki. Проанализированы теоретические основы определения меры семантического подобия между понятиями на основе онтологий и таксономий. Предложены методы построения средств навигации в Wiki-ресурсах, которые базируются на нахождении страниц и используют семантически близкие понятия. Продемонстрирована реализация предложенного подхода в онлайн-версии «Большой украинской энциклопедии» (e-VUE), которая обеспечивает расширение функционала энциклопедического издания путем предоставления средств доступа к контенту на семантическом уровне. Обоснованы методологические принципы организации базы знаний e-VUE и категоризации ее статей, обеспечивающие использование оценок семантической близости между понятиями для поиска подобных понятий.

Ключевые слова: семантические Wiki-ресурсы, семантический поиск, онтология, семантическое подобие, Большая украинская энциклопедия.

Use of ontologies for search and navigation improvement in the online version of the “Great ukrainian encyclopedia” / J. Rogushina, I. Grishanova. – P. 11 –27.

The aim of the work is to analyze the means of semantic search in Wiki resources and to develop directions for expanding its capabilities through the use of modern tools of knowledge management. The article analyzes the benefits that the use of semantic markup of the Web information resources provides for users by more convenient and quick access to content. The techniques of application of ontological analysis for search and navigation in semantized Wiki resources are considered. The definition of Wiki ontology that formalizes the information about the structure of the knowledge base of information resource (both ordinary and semantized) based on Wiki technology is proposed. The theoretical foundations of determining the measure of semantic similarity between domain concepts based on ontologies and taxonomies are analyzed. Methods of design of navigation tools oriented on Wiki resources are proposed. These methods are based on retrieval of pages that use semantically related concepts. The implementation of the proposed approach in the online version of the Great Ukrainian Encyclopedia (e-VUE) is described. Such approach supports an extension of the functionality of the encyclopedic resource by providing means of access to content on a semantic level. The methodological principles of the organization of the e-VUE knowledge base and the categorization of its articles that ensure the use of semantic proximity estimations between domain concepts for discovery of such concepts are substantiated.

Key words: semantic Wiki resources, semantic search, ontology, semantic similarity, Great Ukrainian Encyclopedia.

Использование тезаурусов для поиска сложных информационных объектов в Web на основе онтологий / Ю.В. Рогущина. – С. 28 – 52.

Предложена онтологическая модель взаимодействия между объектами и субъектами семантического поиска в Web, охарактеризованы ее основные элементы, рассмотрены методы ее пополнения и применение для фильтрации информации, которая соответствует персонифицированным потребностям пользователей. Проанализированы типы отношений между экземплярами и классами этой модели и их характеристики, которые могут влиять на временную сложность обработки знаний, представленных на основе этой модели. Одним из важных элементов предложенной модели являются тезаурусы, которые отображают знания относительно задач, для решения которых пользователи ищут информацию, и относительно информационных ресурсов, в которых такие сведения могут содержаться. Обоснована целесообразность применения частных случаев онтологии – тезаурусов – для нахождения семантически подобных информационных объектов. Рассмотрены виды тезаурусов, которые используются для семантического поиска, предложены методы их пополнения и проанализированы их характеристики. В работе предложен алгоритм автоматизированного построения простого тезауруса, который формируется на основе онтологии предметной области и естественного языкового описания задачи пользователя, и методы генерации составных тезаурусов, которые пертинентны новым задачам пользователя, по множеству простых тезаурусов, построенных пользователем ранее. Оцениваются выразительность и вычислительная сложность предложенных методов, которая зависит от свойств онтологии предметной области и от объема описания задачи. Рассмотрены методы использования семантически размеченных Wiki-ресурсов как источника знаний для построения онтологий предметных областей и связанных с

Use of the sauruses for search of complex information objects on Web on the basis of ontologies / J. Rogushina. – P. 28 – 52.

An ontological model of interaction between objects and subjects of the Web semantic search is proposed, its basic elements are characterized, methods for its replenishment and use for filtering information that is pertinent to the personalized needs of users are considered. The types of relations between the instances and classes of this model and their characteristics that can have an influence on the time complexity of processing knowledge presented on the basis of this model, are analyzed. One of the important elements of the proposed model are thesauri that represent knowledge regarding tasks for which users are looking for information, and about information resources where such information can be contained. The expediency of using particular cases of ontology – thesauruses – to find semantically similar information objects is substantiated. The types of thesauruses that are used for semantic search are considered, methods for their replenishment and utilization are proposed, and their characteristics are analyzed. In this paper we proposes an algorithm for the automated construction of simple thesaurus, which is formed on the base of the domain ontology and a natural language description of the user's problem, and methods for generating composite thesauri that are pertinent to new user tasks which are based on the set of simple thesauruses previously built by user. The expressiveness and computational complexity of the proposed methods which depend on the domain ontology characteristics and on the size of the problem description are estimated. Methods of use of the semantically marked Wiki resources as a source of knowledge for constructing ontologies of subject areas and associated typical information objects are considered.

Key words: semantic search, information object, ontology, task thesaurus, semantic markup.

ними типичных информационных объектов.

Ключевые слова: семантический поиск, информационный объект, онтология, тезаурус задачи, семантическая разметка.

УДК 004.94

Классификация метаданных больших данных / О. Захарова. – С. 53 – 74.

На сегодняшний день накоплено много данных разной структуры (или вообще не структурированные) и происхождения, их объемы растут экспоненциально. Проблема заключается в том, что существующее программное и аппаратное обеспечение не в состоянии справиться с таким количеством, появляющихся с огромной скоростью разнообразных типов данных. Большие данные стали слишком сложными и динамичными для обработки, хранения, анализа и управления с помощью традиционных средств. Это обусловило возникновение новых платформ и подходов для работы с данными, и, вместе с тем, четкое понимание того, что для решения задач больших данных эти необработанные данные должны быть дополнены метаданными. Метаданные в данном случае являются средством классификации, упорядочивания и характеристики данных и их содержания. Их главная особенность – это упорядоченная структура. Благодаря структурированному виду, метаданные доступны для чтения не только человеку, но и компьютеру. Таким образом, они могут обрабатываться автоматически и использоваться для индексирования, поиска, объединения, автоматизированной обработки, классификации больших данных и т. д. Построение эффективных систем управления метаданными, прежде всего, требует их согласованной общей классификации с учетом типов источников (способов получения) данных, формирующих контент, задач, решаемых на разных этапах жизненного цикла, существующих форматов представления данных, принципов разумной эффективности, так как часто размер метаданных существенно превышает объем

UDC 004.94

Big data metadata classification / O. Zakharova. – P. 53 – 74.

Now there are a lot of data of different structure (or not structured at all) and origin, their volumes are growing exponentially. The problem is the existing software and hardware are not able to cope with so many different types of data appearing with great speed. Big Data has become too complex and dynamic to process, store, analyze and manage with traditional tools. It caused the appearance of new platforms and approaches for working with data, and at the same time, an understanding of the fact that to solve big data problems, these raw data must be supplemented with metadata. Metadata in this case is a means of classifying, organizing, and characterizing data and its content. Their main advantage is an ordered structure. Due to it, metadata is readable not only by a person, but also by a computer. Thus, they can be processed automatically and used for indexing, searching, combining, automated processing, classification of big data, etc. The creation of effective metadata management systems, first of all, requires their coordinated general classification that take into account the types of data sources (methods of their obtaining) that form the content, tasks solved at different stages of the life cycle, existing formats of data presentation, principles of reasonable efficiency, since often metadata size significantly exceeds the amount of described data (even big). Therefore, the aim of this work is to analyze existing sources of big data, methods for creating and processing the corresponding metadata, as well as software products that allow them to be processed in a certain way, and building the classification of metadata on the basis of the analysis.

Key words: big data source, metadata management, Hadoop, metadata classifi-

описываемых данных (даже больших). Поэтому, цель данной работы состоит в анализе существующих источников больших данных, способов создания и обработки соответствующих метаданных, а также программных продуктов, позволяющих их обрабатывать определенным образом, и построении на основе выполненного анализа классификации метаданных.

Ключевые слова: источники больших данных, управление метаданными, Nadoop, классификация метаданных, анализ метаданных, сервисы работы с метаданными, создание, просмотр, редактирование метаданных, метаданные изображений, метаданные аудио-файлов, метаданные видео-файлов, метаданные хранилищ, метаданные в социальных сетях.

УДК 004.822, 681.3, 519.81

Подход к построению и использованию модели предметной области рисков стратегических решений организации / Е.П. Ильина, О.А. Слабоспицкая. – С. 75 – 91.

Предложена концептуальная модель предметной области рисков стратегических решений в системах организационного управления (СОУ), которая является результатом операций выбора, дополнения и интерпретации, обусловленных спецификой соответствующего объекта риска и выполненными относительно элементов нормативного концептуального базиса оперирования рисками в других сферах деятельности. Модель содержит концепты: *Решение, Последствия для Решения и для СОУ, Источник, Фактор, Аспект и Событие риска, Уязвимость, Опасное воздействие, Затрагиваемое Свойство* и их взаимосвязи. На основе предложенной модели жизненного цикла стратегического решения и обобщения известных практик стратегического управления рассмотрена система мероприятий менеджмента рисков стратегических решений, представленных как действия относительно опасных воздействий, угроз, уязвимостей, объектов риска и последствий. Продемон-

stration, metadata analysis, services for processing metadata, creation, reviewing, editing of metadata, metadata of images, metadata of audio files, metadata of video files, Data Warehouse metadata, social metadata.

UDC 004.822, 681.3, 519.81

An approach for problem area elaboration and usage of organization's strategic decisions risks / E.P. Ilina, O.O. Slabospitskaya. – P. 75 – 91.

A Conceptual Model is proposed for Problem Area of Strategic Decisions' Risks within Organizational Management Systems (OS) being a result of selection, addition and interpretation operations caused by the proper Risk Object specifics and applied to the elements of regular conceptual basis for risk treatment within the other domains. The Model includes such the Concepts: *Decision, Consequences for both Decision and OS; Source, Factor, Aspect and Event of Risk, Vulnerability, Hazard, Affected Property* and their interrelations. Based on the Model proposed for Strategic Decision life cycle and integration of well-known Strategic Management practices the System of Strategic Decisions Risks Management Measures is considered being represented as actions with hazards, threats, vulnerabilities, risk objects and consequences. The Usage of Conceptual Model proposed is demonstrated for the Method elaboration of knowledge ambiguity- caused Risks Assessment for the decisions of Expertise using different business groups and

стрировано использование предложенной концептуальной модели при разработке метода оценки рисков, вызванных неоднозначностью знаний, для решений экспертиз с представителем различных деловых групп и с использованием модели «Дерево ценности». Метод основан на модели, где формализовано множество факторов риска, вызванного альтернативностью знаний. Источниками факторов являются элементы постановки экспертной проблемы. Виды факторов – это виды некогерентного соотношения знаний, уязвимости связаны с характером действий экспертов, опасные воздействия вызваны используемыми системами экспертных оценок, создающими события риска в ходе процесса жизненного цикла решения. Каждый из аспектов риска детализируется свойствами решения, которые являются мишенями риска вследствие событий риска. Метод предоставляет оценки риска для вывода об удовлетворительности экспертизы и возможности выбора наименее рискованной формы организации ее процесса. Развитие и использование предложенной концептуальной модели рисков стратегических решений перспективно для создания средств поддержки процессов оборонного планирования механизмами онтологически базированного оперирования опытом относительно проблемных ситуаций и антикризисных мероприятий.

Ключевые слова: экспертное принятие решений, стратегическое решение, антикризисная программа, менеджмент рисков, концептуальный базис анализа рисков, риск неоднозначности, дерево ценности, оборонное планирование в условиях неопределенности.

УДК 004.4(075.8)

Застосування машинного навчання в програмній інженерії: огляд / О.Г. Мороз, Г.Б. Мороз. – С. 92 –110.

Сьогодні програмне забезпечення є однією з основних технологій, що сприяють розвитку суспільства. Тому його якість є головною вимогою як

«Value Tree» model. The Method is based on the model, where the set of knowledge ambiguity-caused Risk Factors is formalized. The Factors' Sources are the elements of expert problem statement. Factors' types are the types of non-coherent knowledge interrelations, Vulnerabilities relate to experts activity nature, Hazards are caused by the systems of expert estimates in use producing Risk Events during the process of decision life cycle. Each Risk Aspect is specialized with decision Properties being Risk targets due to Risk Events. The Method produces Risk estimates to conclude about expertise satisfactoriness and to enable selection the least risky form of its process organization.

Enhancing and using of Conceptual Model proposed for Strategic Decisions Risks is advantageous for tools elaboration to support Defense Planning processes with the mechanisms of ontology based experience operation concerning with Problem Situations and Anti-crisis Measures.

Key words: expert decision making, strategic decision, anti-crisis program, risk management, risk analysis conceptual basis, ambiguity risk, value tree, defense planning under uncertainty.

UDC 004.4(075.8)

Application of machine learning in software engineering: an overview / O.H. Moroz, H.B. Moroz. – P. 92 – 110.

Today, software is one of the main technologies contributing to the development of society. Therefore, its quality is a major requirement for both the global soft-

для світової індустрії програмного забезпечення, так і програмної інженерії, яка займається всіма аспектами підвищення якості і надійності програмних продуктів на всіх етапах їх життєвого циклу. Для вирішення проблем програмної інженерії все актуальнішим стає застосування методів штучного інтелекту. У статті подано короткий опис методів машинного навчання, таких як штучні нейронні мережі, машини опорних векторів, дерева рішень, індуктивне логічне програмування та інші. Також наводяться приклади застосування цих методів для вирішення деяких проблем прогнозування та оцінки якості в програмній інженерії, даються рекомендації для застосування алгоритмів машинного навчання до розв'язання задач програмної інженерії. Огляд буде корисний дослідникам і практикам в якості стартової точки, оскільки він окреслює важливі та перспективні напрямки досліджень. Це зрештою призведе до ефективнішого вирішення проблем програмної інженерії, забезпечуючи кращі, надійніші та економічно вигідніші програмні продукти.

Ключові слова: програмна інженерія, програмне забезпечення, машинне навчання, нейронні мережі, машини опорних векторів, дерева рішень.

УДК 612.51.001.57+519.6

Симулятор механізмів довготривалого регулювання гемодинаміки людини / Р.Д. Григорян, А.Г. Дегода, Е.А. Джури́нський. – С. 111 – 120.

Розроблено комп'ютерний симулятор (КС) фізіологічних механізмів довготривалої регуляції стану серцево-судинної системи (ССС) людини. Основою КС є кількісні математичні моделі (ММ): а) впливів центральної реніна́нгіотензинової системи на характеристики ССС; б) динаміки загального обсягу крові. Тестові дослідження ММ в автономному режимі показали адекватність моделей. Планується об'єднати цю модель з створеної раніше моделлю строкової регуляції ССС. На основі комплексної моделі

ware industry and software engineering, which deals with all aspects of improving the quality and reliability of software products at all stages of their life cycle. To solve software engineering problems, the use of artificial intelligence methods is becoming increasingly relevant. The article presents a brief description of machine learning methods such as artificial neural networks, support vector machine, decision trees, inductive logic programming and others. Also, examples of the application of these methods to solve some problems of forecasting and quality assessment in software engineering are presented, recommendations for applying machine learning algorithms to solving problems of software engineering are given. The review will be useful by researchers and practitioners as a starting point, because it identifies important and promising areas of research. This will ultimately lead to more effective solving of software engineering problems, providing better, more reliable and cost effective software products.

Key words: software engineering, software, machine learning, neural networks, support vector machine, decision trees.

UDC 612.51.001.57+519.6

A simulator of mechanisms of long-term control of human hemodynamics / R.D. Grygoryan, A.G. Degoda, E.A. Dzhurinsky. – P. 111 – 120.

A computer simulator (CS) of the physiological mechanisms of long-term control of the state of human cardiovascular system (CVS) is developed. The bases of CS are quantitative mathematical models (MM) describing: a) the effects of the central renin-angiotensin system on the characteristics of the CVS; b) the dynamics of the total blood volume. Test studies of MM in offline mode showed the adequacy of the models. It is planned to combine this model with the previously created model simulating the urgent reg-

буде розроблена спеціалізована програмна технологія для підтримки комп'ютерних симуляцій з різних аспектів фізіології кровообігу здорової людини. Спеціальне імітаційне дослідження має виявляти причини різних сценаріїв розвитку гіпертонічної хвороби. Програма написана на мові C ++17.

Ключові слова: математична модель, фізіологія, кровообіг, артеріальний тиск, регуляція, інформаційна технологія.

УДК 612.0 + 159.938 + 57 : 007

Иновационная педагогическая компьютерная технология оптимизации школьного коллектива в аспекте знаний о биосоциальной структуре личности его членов / К.И. Кузьмина, Т.М. Семик, Л.Н. Шереметова. – С. 121 – 131.

В работе обсуждаются современные компьютерные технологии поддержки функционирования школьного коллектива на базе индивидуально-типологического подхода для повышения эффективности обучения и сохранения биосоциального здоровья его членов. С позиций принципов целостности, системности, динамичности, ритмологичности и междисциплинарности рассмотрен социопсихологический динамический портрет коллектива, его индивидуальное адаптационное информационное пространство (ИАИП) – рациональная для жизнедеятельности среда (дизайн). Технологическая схема предлагает для рассмотрения принципиально новые сервисные компоненты: биосоциальная культура школьника (знание своих адаптационных возможностей), мониторинг для постоянной гармонизации коллектива, персонолог-междисциплинарник, в большей степени ответственный за психологический климат среды. Индивидуально-типологический подход для понимания механизмов индивидуальной (а не средне-статистической) адаптации человека, тандем «профессиональная

ulation of CVS. Based on the integrated model, specialized software technology will be developed to support computer simulations on various aspects of the physiology of the healthy person's blood circulation. A special simulation study should identify the causes of different scenarios for the development of hypertension. The program is written in C ++17.

Key words: mathematical model, physiology, blood circulation, blood pressure, regulation, information technology.

UDC 612.0 + 159.938 + 57 : 007

Innovative pedagogical computer technology optimization school community in the aspect of knowledge about the biosocial personality structure of its members / K.I. Kuzmina, T.M. Syomik, L.N. Sheremetova. – P. 121 – 131.

The article discusses modern computer technologies to support the functioning of the school collective on the basis of an individual-typological approach to improve the efficiency of training and preserve the biosocial health of its members. From the standpoint of the principles of integrity, consistency, dynamism, rhythmology and interdisciplinary, the sociopsychophysiological dynamic portrait of the collective, its individual adaptation information space (IAIS) – a rational environment for life (design), was considered. The technological scheme offers for consideration fundamentally new service components: the biosocial culture of the student (knowledge of his adaptation capabilities), monitoring for the constant harmonization of the collective, a personologist – interdisciplinary, more responsible for the psychological climate of the environment. Individually typological approach to understanding the mechanisms of individual (rather than average) human adaptation, the tandem “professional competence – biosocial culture” together with active computerization can become a “breakthrough moment” in modern human knowledge and humanity, thereby significantly increas-

компетенция – биосоциальная культура» вместе с активной компьютеризацией могут стать «прорывным моментом» в современном человекознании и человековедении, тем самым существенно повышая уровень экономического потенциала и культуры страны.

Ключевые слова: биосоциальная культура, индивидуально-типологический подход, здоровье сберегающая компьютерная технология, оптимизация школьного коллектива, психологический климат, профкомпетентность, персолог-междисциплинарник, мониторинг, топ-менеджмент, эффективность деятельности.

ing the level of economic potential and culture of the country.

Key words: biosocial culture, individual-typological approach, health-saving computer technology optimization of the school collective, monitoring, personologist-interdisciplinary learning effectiveness.

ДО УВАГИ АВТОРІВ!

У журналі "Проблеми програмування" публікуються наукові матеріали, які раніше не публікувалися в інших виданнях.

Мова статті: українська, російська, англійська. Обсяг статті — від 6 до 16 сторінок формату А4.

Документ зберігається у форматі doc або docx. Ім'я подається транслітерацією, як прізвище автора (авторів), наприклад, «Petrenko.doc».

Автори можуть користуватися електронною поштою і також телефаксом для ділової переписки та передачі до редакції тексту статті та правки при коректурі. E-mail редакції: tsok@isofts.kiev.ua. FAX: +380 (44) 526 6263, Телефон: 526 5065.

1. Оформлення файлу з текстом статті.

При підготовці файлу використовуються: стиль нормальний (звичайний) або normal; шрифт Times New Roman, розмір шрифту 12 пт.; міжрядковий інтервал – 1,0; абзацний відступ – 1,25 см; вирівнювання – по ширині. У тексті не допускається вирівнювання пропусками; розстановка переносів – автоматична. Формат паперу А4, розміри полів документа – 20 мм. Текст статті після анотації має бути оформлений у 2 колонки, ширина яких – 7,86 см, а пробіл між ними – 1,27 см.

2. Послідовність розміщення та оформлення матеріалу статті.

УДК: індекс за універсальною десятковою класифікацією.

Автори: ініціали та прізвища авторів, курсив (світлий).

Заголовок 1 (назва статті): не містить аббревіатур та строго відповідає змісту статті. Шрифт 15 пт, напівжирний, регістр верхній.

Анотація (мовою статті): 50–100 слів, не містить аббревіатур, зрозумілих із змісту статті. Шрифт 10 пт, звичайний.

Ключові слова (мовою статті): не більше 10 слів, не містить аббревіатур, зрозумілих із змісту статті, подаються в називному відмінку, розділені комами. Шрифт 10 пт, звичайний.

Заголовок 2 (назва розділу): шрифт 14 пт, напівжирний; абзац із центральним вирівнюванням, без переносів. Заголовки нижчого рівня (пункти і т. п.) у самостійний абзац не виділяються і проходять першим реченням текстового абзацу, шрифт 12 пт, напівжирний.

Основний текст статті, має такі необхідні елементи:

постановка проблеми в загальному вигляді і її зв'язок з важливими науковими або практичними завданнями;

аналіз останніх досліджень і публікацій, у яких розпочато рішення даної проблеми і на які спирається автор, виділення невирішених раніше частин загальної проблеми, яким присвячується дана стаття;

формулювання цілей статті (постановка задачі);

виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів;

висновки з даного дослідження і перспективи подальших розробок у даному напрямку;

подяка (за наявності такої).

Формули створюються в редакторі Microsoft Equation 3.0 або MathType. Формули, на які є посилання в тексті, повинні мати наскрізну нумерацію. Номер формули друкується в круглих дужках біля краю правого поля. Розмір основного шрифту редактора формул – 12 пт. Розміри символів у формулах: звичайний – 12 пт, великий індекс – 9 пт, дрібний індекс – 7 пт, великий символ – 18 пт, дрібний символ – 11 пт. Не допускається масштабування формульних об'єктів.

Рисунки мають бути створені вбудованим редактором Word Picture або експортовані з прикладних програм Windows у графічних форматах (bmp, psx, gif, jpg або tif). Рисунки розташовуються по центру. Нумерація рисунків здійснюється відповідно до порядку

згадування у тексті. Нумеровані підписи розміщуються під рисунком з позначенням «Рис. », далі вказується номер рисунка і текст підпису.

Таблиці мають бути підготовлені стандартним вбудованим в Word інструментарієм «Таблиця». Таблиці нумеруються за порядком згадування. На номер таблиці повинно бути посилання в тексті. Номер таблиці вказується в окремому рядку з вирівнюванням по правій стороні (наприклад, «Таблиця 1»). Назви таблиць розміщуються над таблицею з вирівнюванням по центру. Мінімальний розмір шрифту в таблицях – 11 пт.

Література: нумерований список джерел згідно ДСТУ 8302:2015 від 01.07.2016 р., шрифт 11 пт, відступ: спеціальний, навислий, 0,63 см.

Література англійською мовою (References): список використовуваних джерел згідно **Harvard Style**. Джерела з заголовками на латиниці наводяться без перекладу. Для літератури джерел на мовах, що не використовують латинський алфавіт, необхідно забезпечити переведення назв джерел і вказати після них у дужках мову оригіналу. Прізвища та ініціали авторів, слід транслітерувати за правилами як для закордонного паспорта. Приклади оформлення бібліографічних посилань згідно з вимогами **Harvard Style** наведені в багатьох публікаціях, наприклад, за електронною адресою http://www.staffs.ac.uk/assets/harvard_referencing_examples_tcm44-39847.pdf

Дані про авторів: мають починатися рядком «Про авторів:», напівжирний курсив. Далі вказуються для кожного з авторів ПІБ повністю, наукове звання, посада, адреса, кількість публікацій в українських виданнях (приблизна), кількість публікацій в зарубіжних індексованих виданнях (приблизна), індекс Хірша (за наявності), обов'язково номер ORCID (сайт ORCID <http://orcid.org/>).

Дані про місце роботи авторів: починаються рядком «Місце роботи авторів:», напівжирний курсив. Далі вказуються місце роботи, адреса, телефон, факс, електронна пошта, контактний телефон.

3. Оформлення файлу з анотаціями.

Файл з анотаціями містить інформацію двома мовами (наприклад, якщо стаття написана на українській мові, то анотації та ключові слова – на російській та англійській мовах) та має бути оформлений у дві колонки: УДК (шрифт – 8 пт); назва статті (шрифт – 12 пт, напівжирний); прізвища та ініціали авторів (шрифт – 12 пт); текст анотації, ключові слова (шрифт – 10 пт).

Вимоги до анотації англійською мовою: обсяг від 100 до 250 слів, інформативність, оригінальність (не є калькою української або російськомовної анотації), змістовність (відображає основний зміст статті і результати досліджень), структурованість (дотримується логіки опису результатів у статті).

Документ зберігається у форматі doc або docx. Ім'я подається транслітерацією, як прізвище автора (авторів), наприклад, «Petrenko_Annot.doc».

Примітка: Підписний індекс журналу "Проблеми програмування" – **90853**.