



НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ПРОГРАМНИХ СИСТЕМ

ISSN 1727-4907

ПРОБЛЕМИ ПРОГРАМУВАННЯ

НАУКОВИЙ ЖУРНАЛ

PROBLEMS
OF PROGRAMMING
SCIENTIFIC JOURNAL

2021
№ 2

MAIN SECTIONS :

- *Information Systems*
- *Software Environment and Tools*
- *Expert and Intelligent
Information Systems*
- *Mashine Learning Models
and Methods*

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.isoftware.kiev.ua>

Редакційна колегія

Головний редактор
П.І. Андон (Україна)

Заступник головного редактора
О.П. Ігнатенко (Україна)

Секретар редколегії
В.О. Єгоров (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. Куссіль	(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

Затверджено до друку вченою радою Інституту програмних систем НАН України.
Протокол № 4 від 29.04.2021 р.

Редактор В.О. Єгоров
Комп'ютерна верстка В.П. Бумажний

Підписано до друку __.06.2021. Формат 60x84/8. Папір офс. Ум. друк. арк. ____
Обл.-вид. арк. __ Тираж 120 прим. Ціна договірна. Замовл.

Віддруковано ТОВ «Про формат»
вул. Маршала Жукова, 45 б, м. Київ, 02166



ПРОБЛЕМИ ПРОГРАМУВАННЯ

науковий журнал

№ 2

квітень - червень

2021

Заснований у березні 1999 р.

ЗМІСТ

Інформаційні системи

- Рогущина Ю.В., Гладун А.Я.* Розробка тезаурусу домену як сукупності концепцій онтології з використанням семантичної подібності та елементів комбінаторної оптимізації 3
- Родін Є., Сініцин І.* Базова модель захисту прикладних задач розподіленої інформаційної системи 16
- Захарова О.* Визначення ступеня семантичної подібності з використанням апарату дескриптивних логік 24
- Гладун А.Я., Хала К.О.* Онтологічний підхід до аналізу метаданих в домені інформаційної безпеки 34

Інструментальні засоби та середовища програмування

- Григорян Р.Д., Юрчак О.І., Дегода А.Г., Людовик Т.В.* Спеціалізоване програмне забезпечення для моделювання множинного керування та модуляцій гемодинаміки людини 42
- Рагозін Д.В., Дорошенко А.Ю.* Розширений аналіз швидкодії програм за допомогою Valgrind 54

Експертні та інтелектуальні інформаційні системи

- Косовець М., Товстенко Л.* Особливості використання штучного інтелекту при розробці архітектури інтелектуальних відмовостійких радіолокаційних систем 63

Моделі та методи машинного навчання

- Шевченко В.Л., Лазоренко Я.С., Боровська О.М.* Інтенаційна виразність тексту при програмному озвучуванні 76
- Тріантафіллу А.А., Матешко М.А., Шевченко В.Л., Сініцин І.П.* Алгоритм та програмне забезпечення для визначення жанру пісні задля створення музичного хіта 85

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 OF PROGRAMMING

scientific journal

№ 2

April – June

2021

Founded in March, 1999

CONTENTS

Information Systems

- J. Rogushina, A. Gladun* Development of domain thesaurus as a set of ontology concepts with use of semantic similarity and elements of combinatorial optimization 3
- Y. Rodin, I. Sinitsyn*. Security basic model for applied tasks of the distributed information system 16
- O. Zakharova* Defining degree of semantic similarity using description logic tools 24
- A. Gladun, K. Khala* Ontology-based semantic similarity to metadata analysis in the information security domain 34

Software Environment and Tools

- R. Grygoryan, O. Yurchak, A. Degoda, T. Lyudovyk* Specialised software for simulating the multiple control and modulations of human hemodynamics 42
- D. Rahozin, A. Doroshenko*. Extended performance accounting using Valgrind tool 54

Expert and Intelligent Information Systems

- M. Kosovets, L. Tovstenko*. Specific features of the use of artificial intelligence in the development of the architecture of intelligent fault-tolerant radar systems 63

Mashine Learning Models and Methods

- V. Shevchenko, Y. Lazorenko, O. Borovska* Intonation expressiveness of the text at program sounding 76
- A. Triantafillu, M. Mateshko, V. Shevchenko, I. Sinitsyn* Algorithm and software for determining a musical genre by lyrics to create a song hit 85

J.V. Rogushina, A.Ya. Gladun

DEVELOPMENT OF DOMAIN THESAURUS AS A SET OF ONTOLOGY CONCEPTS WITH USE OF SEMANTIC SIMILARITY AND ELEMENTS OF COMBINATORIAL OPTIMIZATION

We consider use of ontological background knowledge in intelligent information systems and analyze directions of their reduction in compliance with specifics of particular user task. Such reduction is aimed at simplification of knowledge processing without loss of significant information. We propose methods of generation of task thesauri based on domain ontology that contain such subset of ontological concepts and relations that can be used in task solving. Combinatorial optimization is used for minimization of task thesaurus. In this approach, semantic similarity estimates are used for determination of concept significance for user task. Some practical examples of optimized thesauri application for semantic retrieval and competence analysis demonstrate efficiency of proposed approach.

Keywords: domain ontology, task thesaurus, semantic similarity, combinatorial optimization

Introduction

A lot of intelligent applications need in background knowledge about domain. Modeling of domain is often realized by ontologies. But processing of unconditioned ontologies is a complex and hard problem. For many tasks it is reasonable and acceptable to use various simplified domain models, for example, thesaurus of domain that is based on domain ontology but contains the lesser part of domain terms and does not contain relations between them.

Every concept of domain ontology is characterized by properties, relations with other concepts and individuals and other characteristics. We propose to define some initial subset of ontology concepts and then define such other concepts of this ontology that are semantically similar to concepts from initial subset in context of user task. This extended set of terms can be considered as a domain thesaurus and be used for user task solving. We propose to use combinatorial optimization methods (particularly the knapsack task) for development of the optimized domain thesaurus that has minimum quantity of concepts but covers all task-specific needs.

Thesaurus and ontologies as means of domain knowledge representation

By definition, “thesaurus” is the study of term usage in given domains associated to a hu-

man activity. A *term* is a sequence of words used in a given domain and which makes sense in this domain. In ontological analysis term corresponds to some concept of ontology. Therefore, thesaurus can be used for domain description.

Domain thesaurus is a sort of terminological base: it is a collection of terms with some set of relations among them. Now many thesauri for medical domain, mathematics, computer science, etc. domains are developed. They are used for unification of terminology, for common interpretation of domain knowledge, for integration of independently developed intelligent software and knowledge bases etc. Thesauri can be used as a bridge from a terminological base to document indexing and for normalization of indexing terms.

Elements of thesaurus can be extracted from natural language (NL) text by means of linguistic analysis. Manual thesaurus building is a hard task and needs much time. But in this way one can guarantee a good quality of the collected terms. Automatic thesaurus building needs less human workforce but the quality is not guaranteed. It relies on the content and structuring of document sources, and also on the methods of NL processing. Another problem deals with selection of NL texts pertinent to analyzed domain.

Domain ontologies. We consider that any human activity that consists of solving dif-

ferent tasks is a characteristic of activity domain. Task solving needs special knowledge, the same for all the tasks that can be represented verbally. Therefore we can speak about special vocabulary of every domain that is used for specification of tasks and their solutions in this domain. A *domain* is considered as a set of the tasks that are solved by specialists of this domain. In process of the task solving all solving subjects (persons, software agents, etc.) use a finite set of objects and a finite set of relations among them. These sets are formed as a result of agreements about understanding among members of the domain community. In the field of the distributed knowledge management the term “ontology” is used for explicit conceptualization of some domain [1]. The focus of ontologies is not only the domain terminology, but also the inherent ontological structure. It shows which objects exist in the application domain, how they can be organized into classes, called concepts, and how these classes are defined and related.

Every domain has phenomena that people allocate as conceptual or physical objects, connections and situations. With the help of various language mechanisms such phenomena contacts to the certain descriptors (for example, names, noun phrases).

At present the usefulness of domain ontologies is generally recognized and causes their wide use. But the elements and the structure of domain ontologies are not defined uniformly in different applications.

Now three main approaches to define domain ontology are used in intelligent information systems (IIS). They are connected with the ways of ontological analysis application and deal with different sciences.

The first one – *humanitarian* approach – suggests definitions in terms understood intuitively but cannot be used for solving of technical problems.

The second one – computer approach – is based on some computer languages (such as OWL, DAML+OIL) for representation of domain ontology and applied software. It realizes the processing of knowledge represented in these languages. Such approach is the most useful for development of knowledge bases (KBs) for IIS.

The third one – mathematical approach – defines the domain ontologies in

mathematical terms or by mathematical constructions. This approach is too complex for applied IIS and is used for finiteness of ontology processing algorithm and estimation of their execution time.

Usually at first step of domain ontology building the humanitarian approach is used, then the mathematical model of ontology is constructed, and at last its software realization is developed.

Till now no generally accepted universal definition of domain ontology has been suggested. In [2] different definitions are analyzed. On the meaningful level domain ontology will be understood as a set of agreements (domain term definitions, their commentary, statements restricting a possible meaning of these terms, and also a commentary of these statements). Domain ontology is:

- the part of domain knowledge that is not to be changed;
- the part of domain knowledge that restricts the meanings of domain terms;
- a set of agreements about the domain;
- an external approximation represented explicitly of a conceptualization given implicitly as a subset of the set of all the situations that can be represented.

All these meanings of the notion of domain ontology supplement each other.

For the successful development of IIS it is necessary to present user knowledge about domain of her/his interests in some form suitable for computer processing. The specifications of high-level domain are formed by integration of the domain structures of low-level domains. It is important to achieve an interoperability of domain knowledge representation. Ontological approach is an appropriate tool for solution of this task. Ontology is an agreement about common use of concepts that contains means for representing the subject knowledge and agreements on methods of reasons. It can be considered as the certain description and reflection of the world in some specific spheres of interest. Ontology in the most general representation consists of: 1) domain terms; 2) relations between these terms that define links of domain classes and individuals; 3) rules of their use and interoperation that limit meanings of terms in the context of particular do-

main [3]. The formal model of domain ontology O is an ordered triple $O = \langle X, R, F \rangle$, where X – finite set of domain concepts; R – finite set of the relations between concepts of the given subject domain; F – finite set of interpretation functions of given concepts and relations.

Domain ontology is a special kind of knowledge base that contains semantic information about some domain in interoperable and formalized representation. It is a set of definitions in some formal language of declarative knowledge fragment focused on common repeated use by the various applications and tasks.

Ontological commitments are the agreements aimed at coordination and consistent use of the common dictionary. The agents (human beings or software agents) that jointly use the dictionary do not feel necessity of common knowledge base: one agent can know something that other ones don't know. Agent that handle the ontology is not required the answers to all questions that can be formulated with the help of the common dictionary.

Every domain with the certain subject of research has its own terminology, original dictionary used for discussion of typical objects and processes of this domain. The library, for example, involves the dictionary relating to the books, references, bibliographies, magazines etc. Thus, pattern of domain is discovered by its dictionary (the set of NL words that are used in this domain). Clearly, however, that the specificity of domain is shown not only in the appropriate dictionary. Besides, it is necessary:

- to provide strict definitions of grammar managing of combining the dictionary terms into the statements,
- to clear logic connections between such statements.

Only when this additional information is accessible, it is possible to understand both nature of domain objects and important relations established between them.

Task thesauri. For description of some domain is always used the certain set of terms X . Each of terms designates or describes some concept or idea from this domain. Aggregate of terms that describes this domain with pointing the semantic relations between terms

is a thesaurus. Such relations in thesaurus always specify the presence of semantic connection between terms. If user needs to solve some task then he/she selects some subset of X dealt with this task. This subset can be considered as a task thesaurus.

The term “thesaurus” for the first time was used still in XIII century by B.Datiny as the name of the encyclopedia. In translation from Greek “thesaurus” means treasure, riches. The thesaurus is the complete systematized data set about some field of knowledge allowing the human or the computer to orient in it. Intelligent information technologies (IIT) consider thesaurus as a dictionary that contains descriptors of the certain field of knowledge with ordering of their hierarchical and correlative relations. These descriptors are represented into thesaurus in alphabetic order but they also are grouped semantically.

Usually thesauri developed for IIS do not contain definitions of terms. Some thesauri can group terms in X (monolingual, bilingual or multilingual) in a hierarchical taxonomy of concepts, others present them in alphabetical order or by a sphere of science.

Task thesaurus is a collection of the domain terms with indication of the semantic relations between them deal with some particular task. Formal model of thesaurus Th is a pair $Th = \langle T_{Th}, R_{Th} \rangle$, where T_{Th} is a finite subset of the domain terms, $T_{Th} \subseteq X$, where R_{Th} is a finite subset of the relations between these domain terms, $R_{Th} \subseteq R$. Task thesaurus can be considered as a special case of domain ontology.

The expressiveness of the associative relationships in a thesaurus vary and can be as simple as “related to term” as in term A is related to term B [4].

Thesaurus databases, created by international standards, are generally arranged hierarchically by themes and topics.

Formal definition of task thesaurus is a list of terms (single-word or multi-word) important to user task in fixed domain enlarged by the set of related terms for each term from the list.

The structure of thesauri is controlled by international standards that are among the most influential ever developed for the library and information field. The main three standards

define the relations to be used between terms in monolingual thesauri (ISO 2788:1986), the additional relations for multilingual thesauri (ISO 5964:1985), and methods for examining documents, determining their subjects, and selecting index terms (ISO 5963:1985). ISO 2788 contains separate sections covering indexing terms, compound terms, basic relationships in a thesaurus, display of terms and their relationships, and management aspects of thesaurus construction. The general principles in ISO 2788 are considered language- and culture-independent. As a result, ISO 5964:1985 refers to ISO 2788 and uses it as a point of departure for dealing with the specific requirements that emerge when a single thesaurus attempts to express “conceptual equivalencies” among terms selected from more than one natural language [5].

Until recently term “thesaurus” was used as a synonym of term “ontology”, however now in IISs with the help of the thesauri frequently describe domain lexicon in a semantic projection, and ontologies apply for semantics and pragmatics modeling in a projection to representation language [6]. The models either of ontologies or of thesauruses include (as the basic concepts) the terms and connections between these terms.

Spheres of task thesauri use in IIS. Ontologies that differ by expressiveness, volume, language etc. are widely used in IIS as a source of background knowledge about domain, users and their beliefs about information processing and representing. Task specifics defines the restrictions on used ontologies. Many researchers differentiate ontologies depending on the complexity of relationships provided by them into “light weight ontologies” and “heavyweight ontologies” [7].

Examples of lightweight ontologies are controlled vocabularies, thesauri and informal taxonomies. Controlled vocabularies are represented by list of domain terms. Taxonomies add hierarchical relations (i.e. “is-a” relation) between terms of controlled vocabularies, and therefore we can estimate some semantic similarity of terms by number of steps between them in this hierarchy. Thesauri add additional information to the terms in taxonomies, including preferred names, synonyms and relations to other terms (e.g. “see also”).

A lot of thesauri are created for various spheres of human activities – medical domain, mathematics, computer science, etc. Thesaurus can be created for single information resource (IR), natural language (NL) document or the set of documents. It can contain all words of source or some subset of them (for example, nouns, words of reference vocabulary or concepts of domain ontology). Thesaurus terms can be extracted from text by means of linguistic analysis or manually.

Now thesauri are widely used in semantic search [8], e-learning [9], competence analysis [10], and personification of information processing in IIS. User models on base of ontologies can support “personal ontology view” (POV) – ontological representation of individual beliefs about domain conceptualization [11].

Heavyweight ontologies contain not only hierarchical term relation but also domain-specific ones with various sets of characteristics (e.g. transitive or reflexive) that can be used for logical reasoning. Processing of heavyweight ontologies demands more time and calculation facilities but such ontologies are much more expressive as compared with lightweight ontologies. Therefore we try to propose methods that are aimed at automated generation of lightweight ontologies (such as task thesauri) on base of heavyweight ontologies according to needs of particular user task.

Constructing of task thesauri. Construction of task thesaurus includes such main steps (Fig. 1):

1. Definition of user task. At first user has to define particular task that is needed in background knowledge and to fix description of this task (by natural language, in some structured form or by the set of keywords).

2. Selection of domain ontology. Thesauri construction is based on use of domain ontologies of the appropriate areas. Therefore user needs an appropriate ontology $O = \langle X, R, F \rangle$ that can be retrieved from some ontology repository with the help of matching with user interests description or constructed (manually or semi-automatically) specially for this task.

3. Generation of the set of thesaurus concepts. The main part of task thesauri $Th = \langle T_{Th}, R_{Th} \rangle$ construction consists in building of set $T_{Th} \subseteq X$ where every $t_i \in T_{Th}$

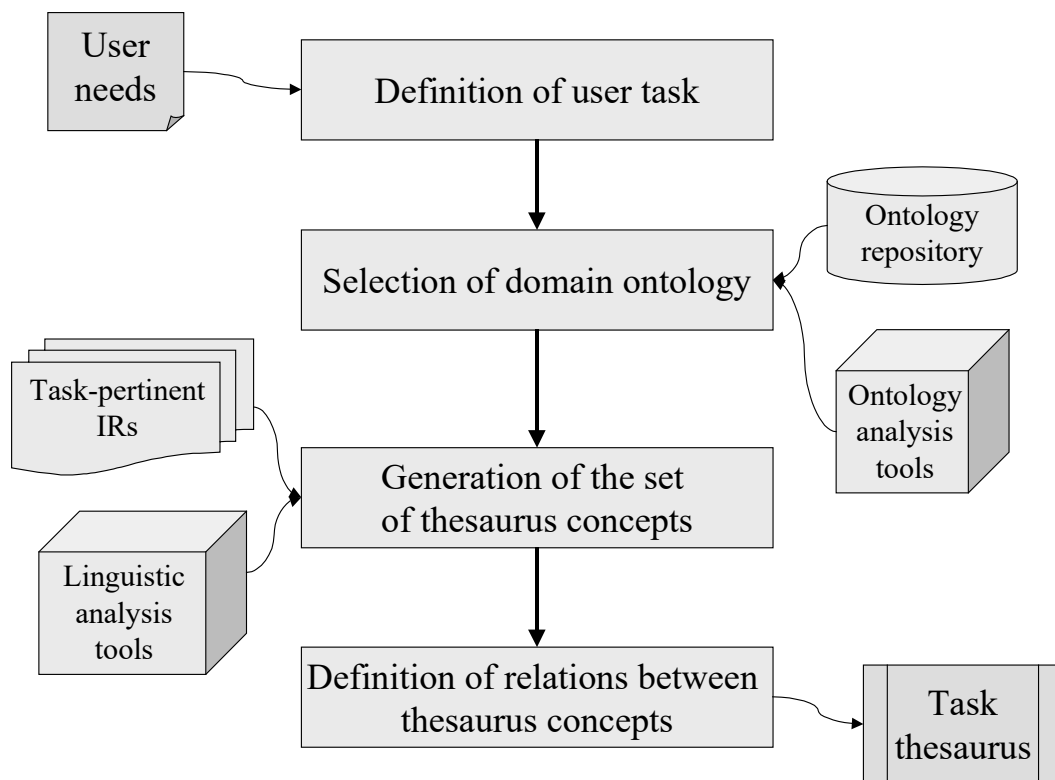


Fig. 1. Main steps and sources of task thesaurus construction

has some semantic matches with some element $w_i \in W_{ut}$ of user task description W_{ut} that $\forall t_i \in T_{Th}, i = 1, n, \exists w_i \in W_{ut}$. This set can be enriched by processing of pertinent IRs (user should independently select the set of IR that he/she considers relevant to domain of his/her interests). Every IR is described by not empty set of the textual documents connected with this IR - text of content, metadata, results of indexing etc. Task thesaurus is formed as a result of the automated analysis of these documents (the user actions are reduced to constructing of semantic bunches - by linking of each word of the formed thesaurus with some term of domain ontology. Algorithm of NL processing for thesaurus building is proposed in [12].

4. Definition of relations between thesaurus concepts. This step provides identification of hierarchical (“class-subclass”, “class-individual”, “is-a”) and synonymic (“see also”) relations from $R_{Th} \subseteq R$ between concepts from $T_{Th} \subseteq X$. These relations can be imported from domain ontology, be extracted from pertinent IRs or be defined manually by user.

In general, task thesaurus can be extended by thesauri of other pertinent IRs and user can edit it manually. This approach is used if task definition is too small and insufficient for

retrieval of necessary data but user has some additional information about task (Fig. 2).

This approach provides generation of task thesaurus if user has any information about task, and this thesaurus contains all domain concepts important for task. But such thesaurus can contain a lot of concepts that are not used in task solving. It increase the volume of thesaurus and causes complications of task solving by IIS

Statement of the problem

For the purpose to reduce the time of task solving and complexity of analysis we propose to construct task thesaurus Th available for solving of user task that contains a minimum subset of terms of domain ontology

$X \cdot |T_{Th_{min}}| \leq |T_{Th_j}|, j = \overline{1, m}$ where Th_j are all possible task thesauri that contain all information from ontology that can be used for task solving and $|A|$ is a number of elements of the set A (sufficiency of information is defined by user and can be estimated by analysis of IIS results).

Development of such minimized thesaurus Th_{min} can be based on semantic similarity between domain concepts. They deal

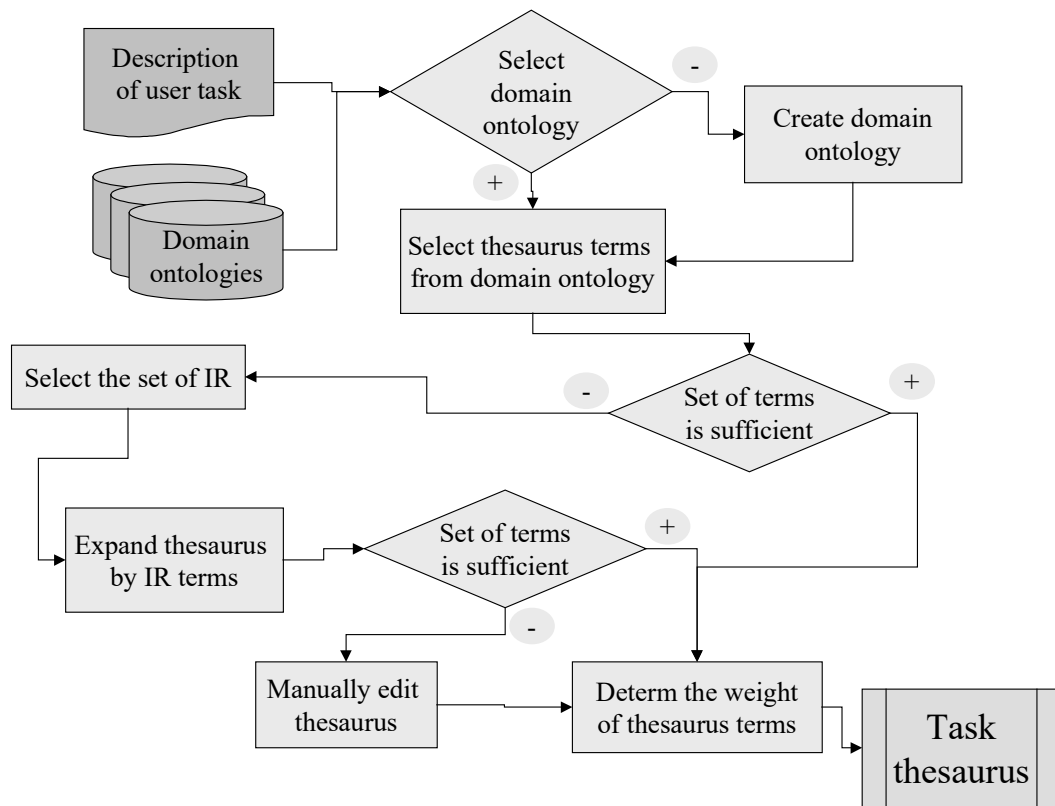


Fig. 2. Generalized algorithm of task thesaurus generation

with user task concepts and on use of combinatorial optimization methods. Such set can be constructed by combinatorial methods as a comparison of all possible subsets.

Combinatorial methods and knapsack task

Similarity estimates are used in recognition tasks for matching various sets of concept properties; individuals and relations with reference definition of used demands. The accuracy of the matching result depends on adequately selected similarity measures.

Combinatorial optimization uses modeling of processed data with finite numerical sequences. The result is evaluated by correlation approach where an expression that define a total product of the values of these sequences establishes the dependence of input information on the combinatorial configuration (objective function argument) [13].

Mathematical formulation of the general problem of combinatorial optimization. Combinatorial optimization problems are usually defined on one or more basic sets, for example $A = \{a_i\}, i = \overline{1, n}$ and $B = \{b_i\}, i = \overline{1, m}$,

n is the number of elements of the set A , m is the number of elements of the set B , the elements of which have any nature [14].

There are two types of combinatorial optimization tasks. In problems of the first type, each of these basic sets is represented in the form of a graph, the vertices of which are elements, and each edge corresponds to the weight of the edge $c_{lt} \in R, l = \overline{1, n}, t = \overline{1, m}$, R is the set of real numbers. There are connections between the elements of these sets A and B , the numerical value of which called scales are set as matrices.

In the second type of task there are no connections between the elements of given set, and the weights are the numbers $v_i \in R, i = \overline{1, n}$ that correspond to some properties of these elements. The numerical values of elements are defined by finite sequences of data.

Knapsack task definition. In this work we use the methods developed for solution of combinatorial task that is known as “knapsack task”. This task is formulated as a combinatorial optimization problem like that: a set of items is given, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less

than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who processes fixed-size knapsack and must fill it with the most valuable items. The problem often arises in resource allocation where the decision-makers have to choose from a set of non-divisible projects or tasks under a fixed budget or time constraint, respectively.

The knapsack problem is a NP-complete combinatorial optimization problem. It got its name from the ultimate goal: to put into knapsack as many valuable things as possible, on conditions that the capacity of the knapsack is limited. Different variations of the knapsack problem can be encountered in economics, applied mathematics, cryptography, logistics, and so on.

The classical formulation of the problem is formulated as follows: there is a set of objects (terms), and each of them has two parameters, weight (significance) and location in the taxonomy of terms. In general, the problem can be formulated as follows: from some given set of items with properties “value” and “weight” we need to select a subset with the maximum total cost, while adhering to the limit on the total weight (adaptation to semantic models is necessary) [15].

A knapsack that has a capacity V must be packed in such a way with n inseparable items with species values $\underline{B} = \{b_i\}, i = 1, n$ and capacities $B = \{b_i\}, i = 1, n$ that the total cost of packaged items would be maximal, and their total capacity would not exceed the value [16]. For the task of thesaurus optimizing we consider items represented by various natural language (NL) information objects (IO). Their values are defined by significance of terms in IO, and capacities – by volume of IO.

Knapsack task can be reduced to the combinatorial optimization task because the knapsack task is given on one set of objects $A = \{a_1, \dots, a_n\}$, there are no connections between the elements a_j of this set, and the input data are given by the elements of the sets B and E that characterize the properties $a_j \in A$, i.e. the problem belongs to the second type of optimization problem. The argument of the objective function is

a combination without repetitions. We set the sequence of sets B and E by numerical functions $\phi(j)|_1^n = (\phi(1), \dots, \phi(n))$ and $\varphi(j)|_1^n = (\varphi(1), \dots, \varphi(n))$. We set combinatorial function

$$\beta(f(j), w^k)|_1^n = (\beta_1(f(1), w^k), \dots, \beta_n(f(n), w^k)),$$

where $\beta_j(f(j), w^k) = 1$, if element a_j is selected from the set A , and $\beta_j(f(j), w^k) = 0$, otherwise. The objective function is reduced to an expression $F(w^k) = \sum_{j=1}^n \beta_j(f(j), w^k) \phi(j)$.

Knapsack task consists in finding of such combination $w^{k^*} \in W$ for which the objective function $F(w^{k^*}) = \max_{w^k \in W} F(w^k)$,

$$\text{if } \sum_{j=1}^n \beta_j(f(j), w^{k^*}) \varphi(j) \leq V,$$

$$k, k^* \in \{1, \dots, 2^n - 1\}.$$

Some variants of knapsack task can be separated:

5. Knapsack task: no more than one copy of each item.

6. Bounded knapsack task: no more than the specified number of copies of each item.

7. Unbounded knapsack task: Arbitrary number of copies of each item.

8. Multiple-choice knapsack task: Items are divided into groups, and only one item can be selected from each group.

9. Multiple knapsack task: There are several knapsacks, each with its maximum weight. Each item can be put in any knapsack or left.

10. Multi-dimensional knapsack task: instead of weight, several different resources are given (for example, weight, volume and packing time). Each item spends a given amount of each resource. It is necessary to choose a subset of items so that the total cost of each resource does not exceed the maximum for this resource, and the total value of items is maximum.

11. Quadratic knapsack task: the total value is given by a non-negative quadratic form [13].

Methods of knapsack task solution. As mentioned above, the knapsack task be-

longs to the class of NP-complete tasks, and there is no polynomial algorithm to calculate it in a reasonable time. Therefore, solving the knapsack task needs to choose between precise algorithms that are not suitable for “large” knapsacks, and approximate ones that work quickly, but do not guarantee the optimal solution to the problem.

Computationally, various approaches have been proposed for solving the knapsack tasks. All these algorithms can be classified into two categories, 1) exact algorithms, and 2) heuristics or meta-heuristics ones [17]. Exact methods for MKP began several decades ago and include branch-and-bound method, special enumeration techniques and reduction schemes, Lagrangean methods and surrogate relaxation methods.

Exhaustive search. As other discrete problems, the problem of the knapsack can be solved by complete processing of all possible solutions. Under the problem conditions there are N items that can be placed in a knapsack, and we need to determine the maximum value of the cargo with weight that does not exceed W .

There are two options for each item: the item is placed in a knapsack, or the item is not placed in a knapsack. Then the search for all possible options has a time complexity of $O(2^N)$, that allows to use it only for a small number of items [18]. As the number of items increases, the problem becomes unsolvable by this method in a reasonable time.

The method of branches and borders is a variation of the method of exhaustive search with the difference that deliberately non-optimal branches of the search tree of complete search are excluded. As well as a method of exhaustive search, it allows to find the optimum decision and therefore concerns exact algorithms.

The original algorithm, proposed by Peter Kolesar in 1967, suggests arranging items by their specific value (in terms of relation of value to weight) and building an exhaustive search tree. Its improvement consists in the process of building a tree for each node: the upper limit of the value of the solution is evaluated, and the construction of the tree continues only for the node with the maximum score [19]. When the maximal upper limit is found

in the tree leaf, the algorithm ends its work. The ability of the branch and boundary method to reduce the number of search options relies heavily on input data. It is expedient to apply it only if the specific values of items differ significantly [20].

Methods for solving the knapsack problem are subdivided into exact and approximate ones. If exact solution needs too much time then approximate solution may be sufficient for practical application.

Approximate methods for the knapsack problem include:

1. An example of bulleted list is as following.

- greedy algorithms;
- ant colony algorithms;
- genetic algorithms.
- The greedy algorithm for the knapsack problem is as follows:
 - the set of items Q is ordered by decreasing the «specific value» of items,
 - then, starting from the empty set, objects from the ordered set items are successively added to the approximate solution Q' (initially this set is empty);
 - each attempt of adding of item to the knapsack is accompanied with comparison of its weight with empty volume of the backpack;
 - the process of constructing an approximate solution to the knapsack problem is ended when all items are considered.

The *ant colony* algorithm is based on the analysis of ant behavior. This algorithm performs the same actions that ants can perform when searching for paths to an object. For each ant, the action of taking an item depends on three components: the ant’s memory, importance of item and the virtual pheromone trace. An ant’s memory is a list of items taken by an ant that cannot be analyzed iteratively. It is also necessary to include in the list those items that break restrictions on the volume of the backpack. Importance of item is the value inverse of the volume of the item. Ant Colony Optimization (ACO) is a meta-heuristic. And it has been applied to many hard discrete optimization problems. Recently, some researchers have proposed several different ACO algorithms to solve the multidimensional knapsack problem (MKP), which is an NP-hard combinatorial optimization problem.

Special importance is given to local information. It is expressed in a heuristic desire to take an object (the smaller the object, the greater the desire) to put it in a backpack. The virtual trace of the pheromone on the item confirms the ant experience dealt with attempt to process it. To study the entire space of objects, it is necessary to ensure the evaporation of the pheromone: at the beginning of the optimization, the amount of pheromone is taken equal to a small positive number, the number of ants can be assigned equal to the number of items.

Stochastic optimization techniques like evolutionary algorithms, simulated annealing etc., which rely heavily on computational power, have been developed and used for optimization. Among these, evolutionary algorithms, which are randomized search techniques aimed at simulating the natural evolution of asexual species, are found to be very promising global optimizers. The *genetic* algorithm used for knapsack problem is based on the evolutionary principles of heredity, variability and natural selection. This algorithm works with a population of individuals and encodes their chromosomes (genotype) for possible solution to the problem (phenotype).

At the beginning of the algorithm, the population is formed randomly. In order to assess the quality of solutions, the fitness function is used to calculate the fitness of each individual. According to the results of the evaluation of individuals, the most adapted of them are selected for crossing. As a result of crossing of selected individuals by using a genetic crossover operator new population is formed.

The multidimensional 0-1 knapsack task is a NP-hard combinatorial optimization problem. The problem is an extension of the standard 0-1 knapsack problem with many constraints while the standard 0-1 knapsack problem has only one constraint. The objective of this approach is to maximize the sum of the values of the items to be selected from a given set by taking into account multiple resource constraints.

All these methods can be used for solution of various problems defined in terms of knapsack task. For minimized thesaurus constructing we need in some estimates that define quantitatively the importance of each domain concept for user task in particular

IIS. We propose to use ontology-based semantic similarity measures of domain concepts for these purposes.

Semantic similarity and criteria of its estimations

Task thesaurus allows to define that subset of domain which is interesting for user in solving a task as a subset of ontology terms that is generated as certain sub-graph of ontology. Such sub-graph can contain, for example, the concepts which are linked to selected terms with selected subset of relations. They should have some properties with defined values or concepts that are semantically similar to selected terms of ontology.

We define *semantically similar concepts* (SSC) as a subset of the domain concepts joined by some relations, properties, attributes or any other characteristics (for example, joint use or identical elements). There are several ways to build SSC that can be used separately or together. Generation of SSC starts from selection of non-empty initial set of concepts. Then various approaches support retrieval of other concepts that are semantically similar to concepts from initial set. User can define SSC manually according to personal beliefs about domain.

More often SSC is generated automatically by processing concept links with initial set of concepts (by some subset of the ontological relations) or with the help of matching concept properties. Such processing defines semantic similarity estimation between analyzed concept and concepts from initial set of SSC.

A lot of different approaches used now to quantifying the semantic distance between concepts are based on ontologies that contain these concepts and define their relations and properties. The source [21] classifies methods and their software realizations of such semantic similarity measuring. Methods are grouped by parameters used in estimations and differ within the groups by calculation of these parameters.

Estimations of semantic similarity. Usually generation of task thesaurus starts from the set of task keywords. Domain ontology can be used to define other domain concepts that

have semantic links with these keywords. All concepts of ontology have some nonzero value of semantic closeness (they are connected one with the other at least by superclass “Thing”). Therefore we have to define what relations of ontology are important for task, what similarity estimations are used and what threshold value of similarity is acceptable.

The similarity of two entities can be defined on base of information about direct and indirect superclasses of these concepts; and instances of these concepts. The most commonly used way of semantic similarity evaluation in taxonomy lies in measuring the distance (path length from one node to another) between concept nodes – semantic similarity is defined as inverse function to the shortest path length. If elements are connected by multiple paths between them the shortest path length is used. This approach is used also for analysis of thesauri [22]. However, this approach is based on hypothesis that all relations between taxonomy concepts represent equal distances, but real taxonomies have great variability of distances covered by the same taxonomic relation, especially if some taxonomy subsets are much denser than others. Some researchers calculate similarity estimates on base of singular taxonomic relations “*is-a*” and exclude other types of relations.

For example, the source [23] considers ontology as a directed graph. Ontology concepts correspond to graph nodes, and universal and domain-specific relations (mainly taxonomic “*is-a*”) correspond to graph edges. Estimation of semantic similarity between concepts is calculated as a minimum path length that connects the corresponding ontological nodes: $SS_{Rada} = \min |path(c_1, c_2)|$. Similarity estimation proposed by Wu and Palmer [24] is based on the analysis of the path between concepts and their depth in the hierarchy: $SS_{WP} = 2H / (N_1 + N_2 - 2H)$, where N_1 and N_2 are calculated as a number of “*is-a*” relations between concepts c_1 and c_2 to the lowest common generic object (subsumer) c , and H is the number of “*is-a*” relations between c and the *root* of taxonomy.

Other researchers take into account also relations “*part-of-part*” [25].

An alternative way of evaluating semantic similarity in a taxonomy, based on the

concept of informational content, which is also not sensitive to the different sizes of distances between relations is offered in [26]. Important factor in the similarity of taxonomy concepts is the degree of their information sharing that defines the number of highly specific terms that is applied to both of these concepts. Measures of similarity based on information content determine the similarity of two concepts. It is defined as information content of their lowest common generic object (subsumer).

In general, all semantic similarity estimates provide some function S : that defines quantitative value of similarity for all concepts of domain ontology. Input information for S includes: domain ontology O , initial set of concepts $C_0 \subseteq X$ and analyzed concept $c_i \in X$,

$$\forall c_i \in X \exists S(O, C_0, c_i) = w_i \geq 0.$$

Optimization of task thesaurus

To reduce task thesaurus $Th = \langle T_{Th}, R_{Th} \rangle$ by methods of combinatorial optimization we have to represent its characteristics in terms of knapsack task. We analyze the set T_{Th} of concepts that are contained in this thesaurus. For this analysis we propose to use:

- the set of task thesaurus concepts $T_{Th} = \{t_k\}, k = \overline{1, p}$;
- domain ontology O that was used as a base for Th generation;
- initial set of task concepts $C_0 \subseteq T_{Th} \subseteq X$ (these concepts have to be placed into all variants of task thesaurus);
- function of semantic similarity estimation S that defines significance of concept for user task;
- values of some selected semantic similarity estimation for all elements of T_{Th} : $w_i = S(O, C_0, c_i) \geq 0$ that can be used as a value from knapsack task;
- length of concept name $l_i = |c_i| \geq 0$ defined as a number of symbols in this name that can be used as a weight from knapsack task;
- user defined memory capacity that is given for thesaurus storage.

We understand that memory needed for thesaurus storage is not a problem now. For NP-complete combinatorial optimization size of processed data it defines the calculation time. Therefore we try to add to C_0 concepts with bigger values of semantic similar-

ity according to one of knapsack task solution methods till then their length l_i is less than the free space in memory for thesaurus. Selection of optimization method and function of semantic similarity estimation depends on task specifics and user needs.

Practical use of optimized task thesaurus

Practical use of optimized task thesaurus. Approach to generation and optimization of task thesaurus for IIS we test on problem of personified information retrieval. Intelligent retrieval system “MAIPS” [27] use thesauri generated semi-automatically on base of domain ontologies selected by users. This IIS use task thesaurus defined by user to filter retrieval results received from retrieval systems. Every user can select one or more domain ontologies and generate one or more task thesauri for each of them. Moreover, users can combine thesauri based on different ontologies by set-theoretic operations. Now we enrich functions of MAIPS dealt with thesauri by optimization operation (Fig. 3). Thesauri in MAIPS are visualized by tag cloud where font size represents the significance of concept for user task.

We compare the time and quality of retrieval with usual task thesaurus and with optimized one and draw a conclusion that processing of optimized thesaurus distinctly accelerates data processing. And use of this filtering of concepts with low semantic similarity estimates not influences substantially retrieval results.

Prospects for further use. We consider that use of combinatorial methods to form optimized user profiles that meet user conditions can be applied in various IIS that work with sets of competencies [30, 31], [28].

For example, if for a certain problem it is necessary to use a set of competencies K , then the problem is solved by construction of minimized set of items (courses, learning disciplines, experts, employees, etc.) P such that $\bigcup_{i=1}^n c(p_i) \subseteq K$, where $c(p_i)$ is a set of competencies of the i -th participant R . Now we plan to include appropriate service into the advisory system “Advisort” [29].

Acknowledgements

The article was prepared in the framework of basic research “Development of meth-

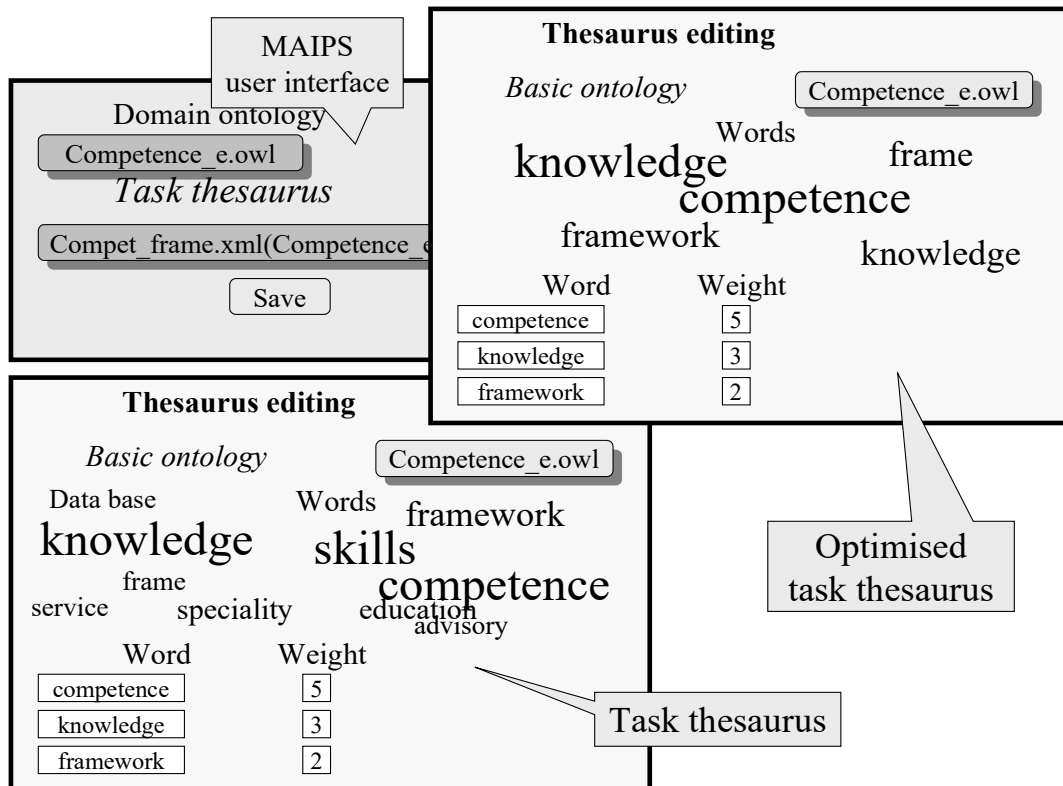


Fig. 3. Use of optimized task thesaurus in MAIPS

ods and tools for integrating combinatorial optimization and semantic modeling for information technology,” topic code VF 170.26 (2017-2022), which was performed at the International Research Center for Information Technology and Systems of the National Academy of Sciences of Ukraine and the Ministry of Science and Education. Results of research work №: III-2-17 “Methods and tools for creating intelligent service-oriented information and support systems in the Semantic Web environment” provided by Institute of Software Systems of National Academy of Sciences of Ukraine were used in theoretical and practical parts of this research.

References

1. Gruber T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5, (1993), pp. 199-220.
2. Kleshche A., Artemjeva I. A Structure of Domain Ontologies and their Mathematical Models. URL: www.iacp.dvo.ru/es/.
3. Gladun A., Rogushina J. Mereological Aspects of Ontological Analysis for Thesauri Constructing. In *Buildings and the Environment*, Nova Science Publishers. (2010), pp. 301-308.
4. The differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a metamodel. URL: www.metamodel.com/article.php?story=20030115211223271.
5. Matthews B.M., Miller K., Wilson M.D. A Thesaurus Interchange Format in RDF. – URL: www.limber.rl.ac.uk/External/SW_conf_thes_paper.htm.
6. Thesaurus Links URL: https://www.w3.org/2001/sw/Europe/reports/thes/thes_links.html.
7. Lassila O., McGuinness D. The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science*, 6(5), (2001).
8. Gladun, A., & Rogushina, J. Use of Semantic Web technologies in design of informational retrieval systems. In book *Building and Environment*, Nova Science Publishers, (2010), pp. 289-299. http://www.novapublishers.org/catalog/product_info.php?cPath=23_67_742&products_id=10117
9. Gladun, A., & Rogushina, J. Use of Semantic Web technologies in design of informational retrieval systems. In book *Building and Environment*, Nova Science Publishers, (2010), pp. 289-299. URL: <https://www.researchgate.net/profile/Anatoly-Gladun/publication/287721726>.
10. Gladun, A., & Rogushina, J. Distant control of student skills by formal model of domain knowledge. *International Journal of Innovation and Learning*, 7(4), (2010), pp. 394-411.
11. Use_of_semantic_web_technologies_in_design_of_informational_retrieval_systems/links/569ff66c08ae4af52546d9cc/Use-of-semantic-web-technologies-in-design-of-informational-retrieval-systems.pdf.
12. Gladun A., Rogushina J. “Distant control of student skills by formal model of domain knowledge”. *International Journal of Innovation and Learning*, 7(4), (2010), pp. 394-411.
13. Rogushina J., Priyma S. Use of competence ontological model for matching of qualifications. *Chemistry: Bulgarian Journal of Science Education*, Volume 26, №2, (2017), pp. 216-228.
14. Kalfoglou Y., Schorelmmmer M. “Ontology mapping: the state of the art.” *The Knowledge Engineering Review* 18(1), (2003), pp. 1–31.
15. Gladun A., Rogushina J. Semantic search of Internet information resources on base of ontologies and multilinguistic thesauruses. *Information Theories & Applications*, Vol.14, (2007), pp. 48-55.
16. Timofieva N. On some approaches to estimating the optimal solution of combinatorial optimization problems, *USiM, Control systems & computers*, №3 (281). (2019), pp. 3–13. URL: doi.org/10.15407/csc.2019.03.003. (in Ukrainian).
17. Sergienko I., Kaspshitskaya M. Models and methods of solving combinatorial optimization problems on a computer. - K.: Naukova dumka, 1981, 281 p. (in Russian)
18. Sigal I., Ivanova A. *Introduction to Discrete Application Programming: Models and Computing. algorithms.*: M.: Fizmatlit, 2002, 237 p.
19. Korbut A., Finkelstein Yu. *Discrete programming.* - M.: Nauka, 1969. 368 p. (in Russian)
20. Kilincli Taskiran G., *An Improved Genetic Algorithm for Knapsack Problems* (Doctoral dissertation, Wright State University). 2010. URL: core.ac.uk/download/pdf/36754668.pdf.
21. Okulov S. *Programming in algorithms.* Binom. Laboratory of Knowledge, 2007. - ISBN 5-94774-010-9. (in Russian)
22. Martello S., Toth P. *Knapsack problems: algorithms and computer implementations.* John

- Wiley & Sons Ltd., 1990. - P. 29.50. - 296 p. ISBN 0-471-92420-2.
23. Burkov V., Gorgidze I., Lovetsky S. Applied problems of graph theory / ed. J. Gorgidze - Tbilisi: Computing Center of the USSR Academy of Sciences, 1974. 231 p. (in Russian)
 24. Taieb M., Aouicha A. H., Hamadou M. B. "Ontology-based approach for measuring semantic similarity." *Engineering Applications of Artificial Intelligence*, 36, (2014), pp. 238-261.
 25. Rada R., Bicknell E. Ranking documents with a thesaurus. *JASIS*, V.10(5), (1989), pp. 304-310.
 26. Rada R., Mili H., Bicknell E., Blettner M. Development and application of a metric on semantic nets. *IEEE transactions on systems, man, and cybernetics*, 19(1), (1989), pp. 17-30.
 27. Wu Z., Palmer M. Verbs semantics and lexical selection. *Proceedings of the 32-nd Annual Meeting on Association for Computational Linguistics, ACL'94*, Association for Computational Linguistics, Stroudsburg, PA, USA, (1994), pp. 133-138.
 28. 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).
 29. 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* 11, (1999), pp. 95-130.
 30. Rogushina J. Analysis of Automated Matching of the Semantic Wiki Resources with Elements of Domain Ontologies. *International Journal of Mathematical Sciences and Computing (IJM-SC)*, 3(3), (2017), pp. 50-58.
 31. Rogushina J., Gladun A., Pryima S., Strokan O. Ontology-Based Approach to Validation of Learning Outcomes for Information Security Domain. *CEUR Vol-2577, Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security"*, (2019), pp. 21-36. URL: ceur-ws.org/Vol-2577/paper3.pdf.

About authors:

Rohushina Julia Vitalievna,
Candidate of Physical and Mathematical Sciences,
Senior Research Fellow.
Number of scientific publications in Ukrainian
publications - 150.
Number of scientific publications in foreign publi-
cations - 31.
<http://orcid.org/0000-0001-7958-2557>,

Gladun Anatoliy Yasonovych,
Candidate of Technical Sciences, Senior Research
Fellow.
Number of scientific publications in Ukrainian
publications - 160.
Number of scientific publications in foreign publi-
cations - 45.
<http://orcid.org/0000-0002-4133-8169>,

Affiliation:

Institute of Software Systems, National Academy
of Sciences of Ukraine.
03680, Kyiv, Ukraine.
Academician Glushkov Avenue, 44.
Tel: 066 550 1999.
E-mail: ladamandraka2010@gmail.com,

International Research and Training Center of In-
formation Technologies and Systems, National
Academy of Sciences of Ukraine and Ministry of
Education and Science of Ukraine.
03187, Kyiv, Ukraine.
Academician Glushkov Avenue, 40.
Tel: 044 502 6366.
E-mail: glanat@yahoo.com

Received: 11.05.2021

Y.S. Rodin, I.P.Sinitsyn

SECURITY BASIC MODEL FOR APPLIED TASKS OF THE DISTRIBUTED INFORMATION SYSTEM

The tasks of modelling and the components of the basic model of applied task protection of a distributed information system have been considered. The measurement and relationship of security parameters, protection, new and reference attacks, anomalies, and threat environments have been proposed. The conditions of threats, attacks and, consequently, inconsistencies in the results of applied tasks are proved. At the beginning of the article the concept of a distributed information system, system of applied tasks, modern trends of zero-trust architecture in building information security systems are discussed. Further, it gives an overview of existing methods of detection and counteraction to attacks based on reference knowledge bases. To improve the level of security it is proposed to analyze the causes of attacks, namely hazards and threats to the system.

Attacks, hazards and threats are considered as structured processes that affect the internal and external environment of the system of the applied tasks with a further impact on the output of these tasks. The concepts of security level and security level of a distributed information system are introduced, as well as the concepts of applied task, environment, and user contradictions. As the logical metrics of discrepancy detection the apparatus of semantic analysis is proposed, which based on the reference knowledge base, the apparatus of text transformations should be applied at the stage of loading of applied task and describe the input and output data, requirements to the environment of the task solution.

The result of the research is the proposed method for identifying additional data about hazards, threats, attacks, countermeasures to attacks, applied task-solving. This data is generated from the reference and augmented textual descriptions derived from the proposed contradictions. By building additional reference images of threats, attacks, countermeasures, it becomes possible to prevent the activation of new attacks on the distributed information system.

Keywords: information, security, anomaly, attack, model, applied task, distributed system, semantics.

Introduction

A distributed information system (RIS) is now the backbone of the infrastructure of any organization dealing with electronic information resources. A RIS is a continuously operating mechanism of interconnected distributed general and application software and hardware, interconnected by telecommunication means. As a single RIS user the organization is interested in maintaining its own applied task. To solve the applied task a RIS consolidates certain resources, which together would be called an application system for solving the user's PSZ problem. It is objective and economically reasonable that the user is only interested in protecting his applied task and the PSZ allocated to this task. Creation of demilitarized zones, physical perimeters of protection of the organization's infrastructure environment, corporate systems for recognition of unauthorized intrusion during remote work of organization's employees does not make sense. The article

formalizes the PSZ parameters required to build a basic model of information security of a distributed system.

Problem statement

The rapid dynamics of new cyber threats poses new challenges to managers and developers of information security protections. An important one is the effective identification of new, previously unrecorded threats and attacks, as well as the adaptation of protocol protections for new attacks.

In this paper we consider the task of determining the possibility of danger and threat in relation to the application system solving the user's task. It identifies and analyzes the causes of hazards (Nb_i) and threats (Zg_i), analyzes the relationship between Nb_i , Zg_i and attacks (At_i), affecting the occurrence of anomalies in the PSZ. A formalization of an adaptive modeling approach to information security system design is proposed.

Analysis of recent studies and publications

With the migration of enterprise software to cloud locations and many infrastructure services to off-the-shelf cloud services and products (SaaS, IaaS, PaaS) in many organizations, a new architecture for building information security is emerging - the adaptive security approach. The main principles of adaptive architecture are proactive continuous threat monitoring, risk auditing, and a shift from single tunneling access of valuable resources to contextual access [1]. The contextual approach to providing access to resources involves defining the requirements for the applied task that orders the resources, continuously monitoring and adapting these requirements.

Each PZa_i for which information resources are commissioned, which the PSZ will use to implement its operational processes, must consolidate relevant resources, which also include a description of the data and RIS requirements. These are:

- the input information that characterizes the PSZ system and is used for its functioning,
- general requirements for the functionality of the resources and their parameters characterizing the RIS system,
- the required overall security levels and security values of the individual designated processes, which can be implemented within one or some individual PZa_i of the PSZ system,
- the requirements for ensuring the required safety value of the operation of the individual tasks of the $PZa_i \ni PSZ$,
- the required measure of recoverability of the individual PZa_i in the case of successful completion of an attack on PZa_i by an appropriate attack.

An important function of PSZ defense processes is the recognition and identification of attacks At_i and recognition of anomalies An_i , which activate attacks in PSZ [2]. Anomaly recognition An_i and attack recognition At_i are quite different from each other. Attack recognition At_i is implemented based on the use of descriptions of the reference images of attacks $E(At_i)$, which are in the database of the system RSB . Recognition of anomalies An_i is implemented based on the analysis of devia-

tions of the parameters of the environment in which they occur, from their threshold values, or $\Delta_j^d(P_i)$. One of the features, which is associated with the difference between $E(At_i)$ and $\Delta_j^d(P_i)$, is that in $E(At_i)$ besides the list of parameters P_i , which can characterize An_i , there are some factors that describe in a certain approximation of the relationship between the parameters $P_i(An_i)$ and parameters $P_i(At_i)$. The second feature, which characterizes the possibility of a relationship between $E(At_i)$ and $\Delta_j^d(P_i)$, is that in $E(At_i)$ the number of parameters P_i should be not less than the number that is necessary to identify the corresponding attack. Since $E(At_i)$ represents some structure, $E(At_i)$ can be represented in the form $\{L = [E_i(At_i)]\} \rightarrow L(E_i)$, for which the relation $E(At_i) = L_i(P_1, \dots, P_k)$ holds, where L_i is the logical function P_i of the parameters characterizing At_i . The relation between An_i and At_i at the logical level can be described by the relation

$$\{[An_i, E(At_i)] \Rightarrow \neg At_i\} \vee \{[An_i, E(At_i)] \Rightarrow [[An_i \Rightarrow L(At_i)] \rightarrow (An_i \Rightarrow At_i)]\}.$$

If at the final stage $[An_i, E(At_i)] \Rightarrow At_i$, then in RSB for At_i there exists an algorithm for counteracting At_i , which we will denote by Ap_i . Such algorithms may differ from each other in characteristics that determine their capabilities and type:

- Algorithm Ap_i that completely neutralizes the impact of the counterattack of At_i by neutralizing all the functions implemented by the attack of At_i ,
- Algorithm Ap_i , which neutralizes the capabilities of Zg_i to activate the retaliatory attack, or $Ap_i(At_i) \rightarrow \neg Fk_i(At_i)$, where Fk_i is the activation function of At_i ,
- Algorithm Ap_i , which partially counteracts the negative impact of At_i on the object of the attack.

The first type Ap_i implements counter attacks that are in the active state. For example, At_i , which has several interrelated stages of its implementation, encounters counteraction in the second, third or other stages of the implementation process, regardless of whether the attack at each stage realizes its impact on the object, which can result in unacceptable changes in functional parameters. Attack localization in this case

is determined by the presence or absence of impermissible modification in the *PSZ* environment. The second type of counteraction Ap_i is to eliminate the vulnerability points of the system, which were used Zg_i to introduce and activate the attack. The third type of algorithm Ap_i , which counteracts At_i , which is activated, is that the counteraction to the attack is implemented only when the effect of the attack on the corresponding object is manifested in unacceptable changes in the functional parameters of the object of attack. This type of attack counteraction can result in blocking some functions in the attacked object, which is *PSZ*.

In addition to the above, other types of algorithms Ap_i can have a fairly wide range of counteraction At_i . The implementation of countermeasures also depends on:

- the completeness of the information about the identified At_i , which must be in $E_i(At_i)$,
- the type of attack At_i , which has been activated in the object of the attack,
- the way of recognizing At_i for which there is no $E_i(At_i)$ in the *RSB*, and other factors.

A rather large number of scientific and technical publications are devoted to descriptions of attacks and ways to counter attacks of various types [3-5].

Task statement

The purpose of the article is to present the elements of information hazard of an applied task of a distributed information system by a model of sequential processes of impact on incoming, outgoing data, and processes of calculation of the problem.

The article explores ways to use the semantic expert system apparatus to identify inconsistencies (anomalies) in the system based on existing records of incident history and countermeasures.

Lets consider aspects related to enhancing the security of *RIS* (and above all *PSZ* application systems), concerning the problems of countering threats Zg_i and analyzing the hazards Nb_i that Zg_i generates concerning *PSZ*. This raises the following challenges.

1. Analyzing the ability to detect and recognize threats based on attack data that have been activated concerning *PSZ*.

2. Establishing (based on Zg_i data) the possibility of influencing Nb_i by *RSB* means in order to prevent the possible initiation of the process of forming the corresponding threat Zg_i .

3. Analyzing particular aspects of the coexistence of Nb_i , Zg_i , and At_i with *PSZ* objects that may be affected by the eventual retaliatory attacks.

In the paper, by using methods of mathematical logic, the elements of the basic security model to be monitored by *RSB* are theoretically laid out.

Presentation of the basic material of the study

Most attacks can be considered as $(Pr_i(At_i))$ processes implemented in a certain sequence, and the whole process of attack implementation can be represented at the level of logical implementation At_i . The corresponding processes $Pr_i(At_i)$ can be conventionally considered as realized in steps, which would be considered as elements of the At_i realization process. In this case, it can be written $Pr_i(At_i) = \{l_1, \dots, l_n\}$, where l_i is a single-step logic realization formula $Pr_i(At_i)$, $l_i = \{x_1 * \dots * x_k\}$, x_i is a logical variable of the l_i formula, "*" is an arbitrary logical function. When l_i passes to the level of dependencies, at which x_i takes values in the given fields of their definition, l_i passes to the form of analytical, discrete, tabular or other forms of dependencies description, which are interpreted by logical functions, can be written in the form $[l_i = \{x_1 * \dots * x_k\}] \rightarrow [F_i^r(x_1 \circ \dots \circ x_k)]$, where " \circ " are operators, which correspond to the chosen function of dependencies description between x_i and x_j . Let us introduce the notation of additional elements, which are directly related to the elements At_i , Nb_i , and Zg_i . The first of these additional elements are the individual fragments of the object of attack, which are, in the first place, the selected components with *PSZ*, these components will be denoted by the symbol v_i . The next additional element will be a specialist Sp_i in the implementation of unauthorized cooperation with *PSZ* using v_i , which represents the potential opportunity to contribute to the success of the implementation of At_i . Let us assume that the model of Sp_i functioning process is

a system of logical inference $I(L, R)$, where L is a system of logical inference rules, R is a system of heuristic inference rules, which can be included in the system I , if necessary. Heuristic elements are used in problem solving processes when inference rules of classical inference system (for example, Gentzen's inference system) are not enough for problem solving [6]. In the context of this problem, we will define a heuristic rule as an inference rule or some dependency based on the interpretation of some fragment of the domain which is not shown on the level of logical dependencies, since the corresponding heuristic rules describe individual features of links or dependencies in such a fragment. In this case, we can write down such a formal model of hazard: $Nb_i = [M(Sp_i) = I(L, R, D)]$, where D is the input data used in the given model to realize the processes of its functioning. The result of functioning $M(Sp_i)$ is the occurrence of Zg_i or $I(L, R, D) \rightarrow Zg_i$. Proceeding from the given variant of description Nb_i as a model of $I(L, R, D)$, we can state that Zg_i is a certain structure At_i which is created based on the use of logical L and heuristic R rules, in which the description of the purpose of functioning Zg_i is formed, which can be written in the form $Zg_i = A_i(L, R, C)$, where C is the purpose of functioning of At_i . As a result of the functioning of Zg_i , it is necessary to obtain a description of the attack - a software product, which is transferred and activated using v_i , or without v_i directly in PSZ . To solve this problem, a library of known software implementations of the corresponding types of attacks is used. From the library a program of the corresponding $Ap_i(At_i)$ algorithm and features of its implementation is selected, which has the closest target C_i^* in relation to the target C_i , which is formed in the Zg_i . Based on the determination of the difference between the targets C_i^* and C_i , the necessary modification of the corresponding program is formed. Thus, Zg_i , forms the attack, which, based on the data on RSB and v_i , is transmitted to PSZ in the form of software implementation of At_i , which is activated. In this case, we will consider the stage of attack object analysis, which is implemented by Zg_i to obtain information about the relevant v_i in order to identify vulnerable points.

In most cases the Nb_i and Zg_i functions are implemented by the corresponding Sp_i specialists. But, in the case of the need to work with distributed systems, which include a large number of individual v_i , it is necessary to automate the relevant processes. Nb_i and Zg_i systems can be considered as objects of influence, which is carried out by means of protection and countermeasures against attacks. Means of protection should prevent the occurrence of attacks [5, 7]. To do this, the following tasks must be solved. Revealing the possibility of an attack by Nb_i , Zg_i and identifying signs of PSZ , which may indicate that the attack will occur.

1. Determining the inevitability of an attack in case of Zg_i initiation.

2. Calculating how many and under what conditions different attacks may occur in relation to v_i if Zg_i is initiated by Nb_i danger.

As part of the experiment we will conduct a theoretical analysis of the possibility of obtaining the information needed to solve the problems of counteraction Zg_i . This requires to obtain data on the identified attacks, to determine the causes of their occurrence in order to eliminate (to a greater or lesser extent) the corresponding causes and to counteract the threats that cause the possibility of attacks in the form of dangerous programs and other factors that can lead to disruption of PSZ processes. Obviously, it is quite difficult to cover all possible causes of At_i or Zg_i , focused on the implementation of the negative impact on the PSZ . Therefore, let us limit ourselves to the functional space of PSZ , reflecting the goals of the creation of the corresponding PSZ and its interaction with the subject area of interpretation of the corresponding $W_i(PSZ)$ system.

Definition 1. The external environment of the $W_i(PSZ)$ system is all digital media that have direct access to the RIS system, regardless of the type of communication channel.

The $W_i(PSZ)$ interpretive domain will be called the external environment PSZ . The $W_i(PSZ)$ environment, which we will denote by $H(W_i)$, can be of the following types:

- 1) an environment $H(W_i)$ that uses information obtained from the PSZ system in its processes, we will call a *passive environment* $H^p(W_i)$.

2) The environment $H(W_i)$, which cooperates with PSZ by forming information presented to the inputs of PSZ , and uses the corresponding results received from PSZ to implement its external processes, will be called an *active environment* $H^A(W_i)$.

The corresponding system of processes functioning in W_i we will denote by V^ASZ and V^PSZ and call their external task systems. Assume that the RIS system, together with RSB , is PSZ friendly. Therefore, we can assume that an arbitrary $Zg_i(PSZ)$ can occur only in $H(W_i)$. Let us consider statements concerning threats and hazards associated with passive and active external task systems.

Assertion 1. Threat $Zg_i(PSZ)$ can arise and exist only in environments V^ASZ, V^PSZ .

Suppose that some $Zg_i(PSZ)$ has arisen outside $W_i(PSZ)$. Since the $W_i(PSZ)$ environment is closed and complete, any $Zg_i(PSZ)$ that originated outside $H^P(W_i)$ or $H^A(W_i)$ must interact with $W_i(PSZ)$. For IS , the functioning of any negative processes is realized by activating the intrusion algorithms (Ar_i) and using information access channels to PSZ .

If $Zg_i(PSZ)$ does not enter $H(W_i)$, then for $Zg_i(PSZ)$ to enter $H(W_i)$, as some intrusive $Ar_i(An_i)$, the latter must have access to the channel of communication with objects with $H(W_i)$. But $H(W_i)$ is a closed environment, which means that $H(W_i)$ provides access only in cases of authorized extension of $H(W_i)$ or in cases where An_i occurs in $H(W_i)$ itself. Since An_i is formed in Nb_i , which is included in $H(W_i)$, and the latter is closed, this contradicts the assumption that $Zg_i(PSZ)$ has access to $H(W_i)$. In the case where $H(W_i)$ is extended by some fragment of $h_i \ni H(W_i)$, the $H^P(W_i)$ and $H^A(W_i)$ systems must identify the corresponding fragment of $h_i(W_i)$ before such an extension can activate its $Pr_i[h(W_i)]$ processes. $H(W_i)$ is friendly to RIS and PSZ . Thus, if $h_i(W_i) \rightarrow \neg H(W_i)$, then $h_i(W_i)$ is identified as $Ar_i(An_i)$, which confirms that the statement is correct.

Let us consider the case when the information contradiction σ^I between $W_i(VSZ)$ and PSZ occurs, which we formally write in the form $\sigma^I[W_i(VSZ) \& PSZ] \geq [\sigma^I(\delta a_i)]$, where δa_i is some threshold value of the contradiction σ^I . Informational contradiction σ^I is the most general type of contradictions because it can

include contradictions: logical σ^L , structural σ^S and semantic σ^C .

In order to find the σ^I contradiction dimension it is used the means that determine the degree of consistency of the information received from RIS with the information expected by the user, the role of which is almost increasingly played by a separate information system, which we will denote by $P_i(EP)$ symbols. An example of such a mechanism for determining $\sigma^C[PSZ, P_i(EP)]$, where $P_i(EP) \ni W_i$, can be the use of representations of the magnitude of the semantic significance of the two corresponding components [8]. For example, a logical contradiction is defined based on the use of known representations of it from mathematical logic, if the objects concerning which it is defined are descriptions at the level of their logical interpretation [8].

Assertion 2. $Nb_i(PSZ)$ hazards can form as a result of contradictions between VSZ and PSZ .

The notion of Nb_i is always associated with certain contradictions between Nb_i and the object towards which it is directed. In general, Nb_i is essentially a contradiction in relation to the object of influence, which in this case is PSZ , which can be written in the form of

$$\{\sigma^I[PSZ, P(EP)] \geq \delta(\sigma^I)\} \rightarrow Nb_i(PSZ). \quad (1)$$

In many cases it is assumed that $Nb_i(Q_i)$, where Q_i is the object of influence, can arise without $\sigma^I(ZQ_i, Q_i)$ contradiction, but for some other reason, where ZQ_i is an object external to Q_i . Such a premise is overly broad.

Let us assume that relation (1) is a precondition for the emergence of Nb_i . Identification of σ^I is an analysis of the $Pr_i[P(EP), VD]$ process, where VD is the input from PSZ . The magnitude of the contradiction is determined by implementing a control of the input data resulting from the operation of the $Pr_i[P(EP), VD]$ process. After performing this control, the following types of results can be obtained:

- The $Pr_i[P(EP), VD]$ process completed successfully,
- The $Pr_i[P(EP), VD]$ process did not complete successfully.

In the second case there may be the following results.

- There are deviations in the results obtained R caused by the $Pr_i[P(EP), VD]$ process itself.

- deviations in the results are caused by the use of false VD , or $Te(Pr_i) \rightarrow [\neg R(Pr_i) \vee \neg R(VD)]$, where Te is the process testing.

Under *Assertion 2*, the case of $Te(Pr_i) \rightarrow \neg R(VD)$ is relevant. Since it is assumed that PSZ works correctly, the relation is valid: $VD(PSZ) \rightarrow H(VD)$ if takes place:

$$\{[H(PSZ) \rightarrow H(VD)][[P(EP), H(VD)] \rightarrow R[Pr_i[P(EP)]]] \rightarrow [P(EP) \rightarrow Nb_i(PSZ)]$$

The interpretation of the above deduction in bringing *Assertion 2* to the qualitative level is as follows: If $Pr_i(PSZ)$ produces VD_i , or $Pr_i(PSZ) \rightarrow VD_i$ without a $P(EP)$ consumer in $W_i(PSZ)$, then in $W_i(PSZ)$ the existence of PSZ is invisible. We can write the following relation:

$$\{[Pr_i(PSZ) \rightarrow VD_i][P(EP), VD_i] \rightarrow \{ \neg Pr_i[P(EP)] \rightarrow [P(EP) \rightarrow Nb_i(PSZ)] \}$$

If PSZ produces VD_i and there is a $P(EP)$ using VD_i in $W_i(PSZ)$, then in the case of $\{[P(EP), VD_i] \rightarrow \neg Pr_i[P(EP)]\}$, there is a $P(EP) \rightarrow Nb_i(PSZ)$ relation, which proves the statement.

The important task is to determine whether $Zg_i(PSZ)$ can occur when there is a Nb_i hazard in $H(W_i)$. An arbitrary hazard represents some object or process, or other factor, which, by its nature, should not be intended to create threats to objects that may be in its environment. It is reasonable to regard Nb_i as some factor within the framework of such interpretations:

1. Nb_i hazards are created artificially, or under the influence of natural factors in a way that causes the possibility of adverse effects on objects in the environment.

2. Some object Q_i , which is formed artificially from the hazards, may be, in terms of processes of its functioning $Pr_i(Q_i)$, incompatible with the already existing objects of the general environment.

Conclusions

It follows from the above that the presence of Nb_i is caused not so much by the nature of the factors taken separately, as by the negative nature of the possible interaction of Nb_i with the potential ExQ_i objects of its environment. The emergence of Nb_i in $H(W_i)$ can be described as follows: $[(Q_i) \rightarrow \neg(PSZ)] \rightarrow Pr_i(Q_i) \rightarrow Nb_i$.

The above relation reflects the conditions for the existence of Nb_i on the binary level. Since there is a task to provide transformations $Pr(Q_i) \rightarrow Nb_i$, in order to use a binary interpretation the process and must be divided into separate components and provided them with an appropriate interpretation. Therefore, let us assume that the following takes place:

$$Pr_i(Q_i) \rightarrow Ea[Pr_i(Q_i)] \rightarrow Ed[Pr_i(Q_i)] \rightarrow Er[Pr_i(Q_i)] \rightarrow Nb_i$$

where Ea is a stage to arise, Ed is a stage to develop, Er is a stage to ripen. The introduction of such stages requires the identification of attributes or ranges of values of parameters that characterize the corresponding stage of the $Pr_i(Q_i)$ process. The allocation of the corresponding stages can be realized based on the analysis of each type of $Pr_i(Q_i) \rightarrow Nb_i(PSZ)$. In this case different $Pr_i(Q_i)$ at different stages can be combined into separate classes. The emergence of different stages in $Pr_i(Q_i)$ may not mean that the corresponding $Pr_i(Q_i)$ will lead to the emergence of Nb_i .

Let us assume that the emergence of different stages of $Pr_i(Q_i)$ is associated with changes in the values of contradiction between $Pr_i(Q_i)$ and $Pr_i(PSZ)$. Therefore, it is necessary to consider ways of determining the different levels of contradiction $\sigma(Q_i, PSZ)$.

The first level of contradiction σ_1 corresponds to a situation when the use of a result of $Pr_i(Q_i) \rightarrow R_{fi}(Q_i)$ functioning in $H(W)$ does not cause disturbances in $Pr_i(PSZ)$, and the contradiction appears in the occurrence of information redundancy within $Pr_i(PSZ)$ due to the transfer to $Pr_i(PSZ)$ of a result of $R_{fi}(Q_i)$, or:

$$\{[Pr_i(Q_i) \rightarrow R_{fi}(Q_i)] \& [R_{fi}(Q_i) \rightarrow Pr_i(PSZ)]\} \rightarrow [Pr_i(PSZ) \leftrightarrow Pr_i(PSZ, R_{fi}(Q_i))].$$

This measure of $\sigma^a(Q, PSZ)$ corresponds to the situation when the presence in $Pr_i(PSZ)$ of the result of $R_{fi}(Q_i)$ functioning $Pr_i(Q_i)$ does not lead to changes in the process of functioning of the potential $P(EP)$ consumer due to his use of the received result of $Pr_i(PSZ)$ functioning.

The second level of contradiction σ_2 corresponds to a situation where the use in $Pr_i(PSZ)$ of the result of $R_{fi}(Q_i)$ functioning leads to the formation of data in $Pr_i(PSZ)$ that are different from those expected by the $P(EP)$

user, but they are non-critical or do not lead to negative consequences of $P(EP)$ functioning.

The third level of contradiction σ_3 corresponds to a situation where the use in Pr_i (PSZ) of $R_{fi}(Q_i)$ results leads to the formation of data in Pr_i (PSZ) that are unacceptable, or critical for $P(EP)$, but their use in Pr_i (PSZ) does not lead to unacceptable changes in Pr_i (PSZ); this situation is interpreted as the occurrence of Nb_i .

The fourth level of contradiction σ_4 corresponds to the situation when the use in Pr_i (PSZ) of the results of $R_{fi}(Q_i)$ causes disastrous consequences in Pr_i (PSZ), because in this case the danger is formed.

In the framework of the formulated problem it is proposed to consider the processes connected with the interaction of external objects Q with Nb_i as separate stages, and also different levels of contradictions are introduced, which are the main signs of the possibility of occurrence of dangers and threats Nb_i and Zg_i .

The concept of adaptive approach to building an information security system so far raises the question of contextual access to resources and continuous monitoring of information system security indicators, but does not answer what processes and indicators of information system state should be checked [1, 9]. This paper proposes formal markers for monitoring the state of security and protection of an information distributed system.

Prospects for future developments

Information security systems are no longer capable of operating protection perimeters and modeling breaches based on perimeter crossing by unauthorized users or software [10].

Actual today is: monitoring of anomalies, threats, attacks, countermeasures, formation of reference knowledge bases based on the analysis of contradictions in the system of applied tasks, stages of development of threats, attacks within specific resources, actions of users (personified or individual information systems) [11,12]. The paper outlines a formalized basis of parameters for assessing anomalies in distributed systems.

The models of applied task processes, environment, hazards, threats, attacks, anomalies, and contradictions of distributed system objects functioning are proposed. The theoretical foundations of using a semantic analysis expert system to monitor anomalies, determine deviations from standards, form new knowledge base images of threats, attacks, countermeasures are highlighted. In the future, the proposed methods can be applied in the development of software for monitoring and protection of distributed information systems. The experience of using a semantic expert system to analyze and use the knowledge accumulated by IDS/IPS systems is of interest. The results of the study can be used in the construction of fuzzy rules of relationships of vulnerabilities, threats, attacks, countermeasures, consequences for further use of fuzzy logic apparatus of information security risk management.

References

1. Risk Adaptive Approach, Gartner. (2018). <https://www.gartner.com/teamsiteanalytics/servePDF?g=/imagesrv/media-products/pdf/Forcepoint/Forcepoint-1-4YCDU8P.pdf>.
2. Joint Task Force. (2018). Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-37, Rev. 2. <https://doi.org/10.6028/NIST.SP.800-37r2>.
3. Lukatsky, A. I. (2001). Detection of attacks. SPb.: BHV-St-Petersburg, 624. (In Russian)
4. Zaytsev, O. I. (2006). ROOTKITS, SPYWARE/ADWARE, KEYLOGGERS & DACKDOORS: detecting and protecting. SPb.: BHV-St-Petersburg, 304. (In Russian)
5. Guide for Conducting Risk Assessments. (2012). NIST SP 800-30, Rev. 1. National Institute of Standards and Technology. September, 2012. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf>.
6. Kleene, Stephen. (1973). Mathematical Logic : monogr. Moskva: Mir, 1973. (In Russian)
7. IOTW: World's Third Largest Music Company Falls Prey To Magecart Attack. (2020). 2020/11/09, 1–2. <https://www.cshub.com/attacks/articles>.

8. Korostil, Olga, Korostil, Yurii. (2015). Usin text models in systems of control of social objects. Scientific Journals Maritime University of Szczecin: Akademia Morska w Szczecinie, 42(114), 112–117. ISSN 1733-8670.
9. Common Criteria for Information Technology Security Evaluation. (2017). CCMB-2017-04-001. <https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf>.
10. Zagorodnyy, A., Borovska, O., Svistunov, S., Sinitsyn, I., Rodin, Y. (2014). Creation of an integrated information resource protection system in the national grid infrastructure. Kyiv: Stal, 373. (In Ukrainian)
11. CISO Strategies & Tactics For Incident Response. (2020). August, 2020, 7–11. <https://www.cshub.com/executive-decisions/reports/ciso-strategies-tactics-for-incident-response>.
12. Scott, Rose, Oliver, Borchert, Stu, Mitchell, Sean, Connelly. (2020). Zero Trust Architecture. NIST Special Publication 800-207. August, 2020, 6–35. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf>.

Received: 30.04.2021

About authors:

Rodin Eugene

junior researcher

The number of articles in the national database – 5

ORCID: <https://orcid.org/0000-0003-2416-8572>

Sinitsyn Igor

chief of department

The number of articles in the national database – 80

ORCID: <https://orcid.org/0000-0002-4120-0784>

Affiliation:

Institute of Program Systems of the National Academy of Sciences of Ukraine.

Academician Glushkov Ave., 40, building 5, Kyiv, Ukraine, 03187

(044) 526-55-07, +380674070962

E-mail: yevheniy.s.rodin@gmail.com

Institute of Program Systems of the National Academy of Sciences of Ukraine

Academician Glushkov Ave., 40, building 5, Kyiv, Ukraine, 03187

(044) 526-41-08, +38067-2261313

E-mail: ipsinitsyn@gmail.com

O. Zakharova

DEFINING DEGREE OF SEMANTIC SIMILARITY USING DESCRIPTION LOGIC TOOLS

The purpose of this study is to determine effective approaches to define the value of semantic similarity of information. The special functions to determine quantitative indicators of a degree of semantic similarity of the information allow ranking the found information on its semantic proximity to search request/template. Forming such measures should take into account many aspects from the meanings of the matched concepts to the specifics of the business-task in which it is done. A combination of semantic and structural approaches is appropriate when constructing the similarity functions. This allows to do descriptions of the concepts more detail, and the impact of syntactic matching can be significantly reduced by using more expressive descriptive logics to represent information and by moving the attention to semantic properties.

The focus of this research is in the methods for evaluating similarity of concepts. Values of similarity between individuals and between a concept and an individual are defined by finding the most specific concept for individual(s) and evaluating the similarity between the appropriate concepts. Using some of defined measures is demonstrated on a geometry ontology application.

Key words: semantic similarity of information, a similarity value, least concept subsumer, the most specific concept, the most specific is-a ancestor, similarity measures, features-based models, semantic-network based models, information content based models, existential concepts similarity.

Introduction

The task of discovery of concepts that are semantically similar and evaluating a degree of their similarity is very important both for resolving applied problems (discovery of semantic web services, effective semantic search of information, data categorization, etc.), and for more general problems in the information technologies area, as, for example, integration of ontologies/knowledge, information search, etc. There are a lot of approaches that try to resolve the problems of finding similarity by methods of text analysis or using special vocabularies as Wordnet [1], for example. As a rule, in such approaches only atomic concepts are considered, but more complex ones are out of the question. In addition, the cases of identifying similarities between individuals and between an individual and a concept are omitted. Also, note, that measures of information similarity should be based on semantics because the purely syntactic approach is too weak to ensure that standard inferences are executed, especially if expressive descriptive logics (for example, *ALC*) are considered as a language for knowledge representation. It is clear that algorithms and functions of similarity measures must be effective. If

they are too complex, they can't provide the desired result in a reasonable period of time and become commonly used.

Last time many studies have appeared that emphasize the feasibility of using ontologies and based on them functions of semantic similarity to compare concepts and / or individuals that can be obtained through the integration of heterogeneous sources of information [2,3,4,5].

The main purpose of this study is the analysis of methods, models and approaches for creating quantity indicators that evaluate a similarity degree of knowledges represented with descriptive logic (DL) tools, their classification and application.

Types and levels of similarity determination

Information/knowledge similarity may be considered and defined on different levels. Namely, we can identify:

- 1) **Conceptual level** – determination of similarity between concepts;
- 2) **Knowledge level** – determination of the similarity between instances of concepts;
- 3) **Mixed level** – determination of the similarity between an instance and a concept.

Similarity measures, as a rule, use the basic Set Theory and they are based on objects commonality. In particular, the basic criteria to determine such measures can be formulated as follows: the value of similarity between objects is not only a result of their common features but it is, also, a result of their differences. This criteria corresponds to the theoretical-informational definition of similarity. The objects, in this case, are concepts and instances of concepts.

We consider approaches for defining similarity measures and corresponding models for evaluating on the each level. But, first, we introduce some definitions used by the most existing models.

Basic concepts and definitions

Definition 1. LCS (Least Concept Subsumer) [23, 24] – the least common subsumer of concepts. Let L is DL. A description of the concept E in DL \mathcal{L} is a LCS of concepts' descriptions C_1, \dots, C_n in \mathcal{L} (shortly $LCS(C_1, \dots, C_n)$), if:

1) $C_i \sqsubseteq E$ for $i = 1, \dots, n$ and

2) E is the least description of \mathcal{L} -concept that meets the first condition, so as, if E_0 is description of L -concept such that $C_i \sqsubseteq E_0$ for all $i = 1, \dots, n$, then $E \sqsubseteq E_0$.

At once, it should be noted that LCS doesn't exist for everyone DL used to represent knowledge, but if LCS exists, it is unique to the point of equivalence. All measures that will be discussed below are based on DL \mathcal{ALC} . As shown in [6], LCS always exists for \mathcal{ALC} DL and it is defined by the concepts' disjunction. In the case if the logic doesn't support the disjunction operator, LCS is calculated by selecting general concept names in its descriptions (within the concepts of the universum and existential constraints for the same role), not taking into account TBox as a whole [6]. But, in this case the result of LCS evaluation may be very common. Based on these considerations, LCS is calculated relative to TBox, on the basis of which the concepts are defined [7].

Taking into account the TBox, LCS definition can be reformulated as follows.

Definition 2. Let \mathcal{L}_1 and \mathcal{L}_2 are descriptive logics such as \mathcal{L}_1 is sub-logic of \mathcal{L}_2 , so \mathcal{L}_1 includes less constructors which are used to build expressions. For given TBox of \mathcal{L}_2 logic

\mathcal{T} , $\mathcal{L}_1(\mathcal{T})$ is set of concept descriptions that can include concepts defined in \mathcal{T} . C_1, \dots, C_n are concept descriptions from $\mathcal{L}_1(\mathcal{T})$, so $LCS(C_1, \dots, C_n)$ in $\mathcal{L}_1(\mathcal{T})$ w.r.t. TBox \mathcal{T} is the description of the most specific $\mathcal{L}_1(\mathcal{T})$ concept that includes C_1, \dots, C_n on TBox \mathcal{T} . In particular, it is such description of $\mathcal{L}_1(\mathcal{T})$ - concept D , as:

1) $C_i \sqsubseteq_{\mathcal{T}} D$ for $i = 1, \dots, n$ and

2) If E is a description $\mathcal{L}_1(\mathcal{T})$ - concept such as $C_i \sqsubseteq_{\mathcal{T}} E$ for all $i = 1, \dots, n$, $D \sqsubseteq E$.

If LCS for TBox doesn't exist (for example, in the case of cyclic TBox), its approximation is calculated. It is named Good Common Subsumer (GCS) [25] w.r.t. TBox and it exists for general TBox. GCS is calculated by defining the least conjunction of concepts and their objections which can include a conjunction of concept names of top level for each considered concept and the same conjunction of concepts constituting the rank of existential and universal constraints on the same role. GCS is the most specific covering than LCS calculated unrelated to Tbox. But, in a general case, it includes (or it is equivalent) LCS calculated w.r.t. TBox [7].

MSA (Most Specific is-a Ancestor)

[8] – the most specific ancestor in the hierarchy of the taxonomy. It is defined as binary relation on concepts taxonomy, but semantically it is similar to LCS. Both calculate the most specific generalization of input concepts (w.r.t. the operator of subsume). Their difference is next. MSA works on a taxonomy of concepts and returns one concept, which contains two original concepts (there is their is-a ancestor) and it does not include anyone else what meets the same requirements. LCS is a description that covers input concepts, and, as a result, returns all concepts included in it. If concepts only related by generic relations (TBox is a taxonomy) then LCS is reduced to one ancestor and $LCS(C_1, C_2) = MSA(C_1, C_2)$.

MSC is the most specific concept. It is unary relation on a set of individuals of ABox.

Definition 3. [25] Let given ABox \mathcal{A} and a is an individual from this ABox, then the most specific concept for a w.r.t. ABox \mathcal{A} is a concept C , denoted as $C = MSC_{\mathcal{A}}(a)$, such that $\mathcal{A} \models C(a)$, and $\forall D$ such that $\mathcal{A} \models D(a)$, $C \sqsubseteq D$ (where \models is the inference operator).

At once, it should be noted that in the general case of acyclic ABox in the expressive DL \mathcal{MSC} cannot be expressed by the final description of the concept [2], it is possible to obtain only its approximation. So, the existence of the most specific concept for an individual of ABox is not guaranteed, or it is difficult to calculate, and the approximation is limited by some depth of a set. A maximal depth of the approximation, as defined in [20], corresponds to the depth of ABox. In this case, we can define the most specific concept $MSC(\alpha)$ or its approximation $MSC^*(\alpha)$ for any instance α of ABox.

Defining a semantic similarity of concepts

Today, a lot of researches exist that try to transform semantic relations between concepts into some quantitative indicators. It is clear, that the principles of formation of such measures are affected, first of all, by the essence of the compared concepts, and the business problem for the solution of which the similarity functions are chosen or determined. The most of existing studies use a semantic approach in conjunction with a structural one, which compares the descriptions of the concepts under consideration. Certainly, this allows to significantly detail the description, and the influence of syntactic matching can be reduced by using more expressive DLs to represent information and by moving the focus to semantic properties of concepts.

In establishing the degree of semantic correspondence between the concepts of the same ontology, the similarity function, in fact, is a mapping $\mathcal{S}: \mathcal{L}(\mathcal{T}) \times \mathcal{L}(\mathcal{T}) \rightarrow Y$, where \mathcal{T} is TBox of this ontology represented in DL \mathcal{L} , and Y is real value, which quantifies the degree of similarity. In measures that are based on ratio $Y \in [0,1]$ but another measure models also exist.

In general, the task is more complex. If matching concepts from different ontologies with TBoxes $\mathcal{T}1$ and $\mathcal{T}2$ of DLs $\mathcal{L}1$ and $\mathcal{L}2$, respectively, it is needed to build the mapping $\mathcal{S}: \mathcal{L}1(\mathcal{T}1) \times \mathcal{L}2(\mathcal{T}2) \rightarrow Y$.

In any case, the similarity function must have the following properties:

1) let E is a set of items (objects of the same or different ontologies), for which the

value of similarity should be determined, then function \mathcal{S} is defined on the set $E \times E$;

2) the function \mathcal{S} is positively defined, so $\mathcal{S}(C,D) \geq 0$;

3) $\forall C,D: \mathcal{S}(C,D) \leq \mathcal{S}(C,C)$.

Defining the similarity function, it is necessary to understand that concept similarity may be considered both in terms of the degree of their commonality and the degree of their difference, and the similarity function should have a positive correlation with the value of commonality between the concepts and negative correlation with an indicator of the difference between them. It is clear this indicator depends on many factors, namely: the specifics of the content that is studied, the expressiveness and homogeneity of the languages of representation of ontologies, and so on. But the key question in determining a similarity function is “how to calculate the value of commonality (difference) of concepts”, which, in turn, is related to the question “how we collect an investigated information”. It is unlikely that the similarity indicator can be considered as an absolute value, but it should provide the possibility of a reliable ranking of concepts by similarity values. As the main approaches to the defining such function can be distinguished:

1) defining similarity as function of a path-distance between taxons in hierarchy which underlies this ontology [10, 11, 12];

2) evaluating a feature-based semantic similarity [13];

3) defining a value of similarity by information content [14,15];

4) existential similarity of concepts.

The first approach may be applied only based on one ontology, so its usage may be appropriate only if the evaluation is performed on the basis of a single source of information, and the matched information items are concepts of a same ontology or an integrated ontology of information sources. Another approach for calculating semantic similarity uses both general and discrimination features between concepts and/or individuals. Methods of third group are based on Theory of information. They determine measure of similarity between two concepts in the hierarchy from the point of view of the amount of information transmitted directly by the super-concept, which includes the matched concepts. We may name all measures

that are based on features of concepts as the measures of intentional similarity. Under the *existential similarity of concepts* we will understand the degree of their closeness by the sets of instances that they include.

In the case of matching concepts of different, probably heterogeneous, ontologies listed approaches works only when certain conditions and restrictions are carried out. First, formal representation of these ontologies should support inference engines such as subsume. (Note that subsume engine is supported by basic DL as, for example, *ALC*). Second, applying the calculation approaches are based on using a general ontology, and local concepts in different ontologies should inherit the structure of description from their general ontology. In [16] some approaches for matching such concepts from different ontologies by their individuals are proposed. Namely, it is made an assumption that when the restrictions are performed the criteria of matching two concepts may be intersection of sets of their individuals. To match descriptions of the concepts that may be united in the general ontology, three main approaches are applied:

- filtering based on path-distance between concepts in the general ontology;
- defining measures based on matching graph that establish one-to-one correspondence between elements of the concept descriptions;
- defining probabilistic measures that give the correspondence in terms of the joint distribution of concepts.

$$sim(C, D) =_{def} \frac{f(ftrs(C) \cap ftrs(D))}{f(ftrs(C) \cap ftrs(D)) + \alpha f(ftrs(C) \setminus ftrs(D)) + \beta f(ftrs(D) \setminus ftrs(C))}$$

If suppose that similarity function is symmetric then $\alpha = \beta = 0,5$. Assuming that the function f is distributive on intersected sets then $sim(C, D)$ may be transformed as follows:

$$sim(C, D) =_{def} \frac{2f(ftrs(C) \cap ftrs(D))}{f(ftrs(C)) + f(ftrs(D))}$$

In the **semantic-network based models** a reference information is given in the semantic-network form that includes nodes-concepts and, at least, is-a edges (sometimes it contains more complex relations as in

Also, if computation of similarity values is performed for concepts belonging to different ontologies, it is necessary to take into account a difference between formalization levels of specification of these ontologies. Particularly, in [17] a similarity function determines classes of similar entities by matching using synonym sets, semantic neighborhood, and discriminating features that are classified into parts, functions, and attributes. In [9] another approaches are presented. It is aimed at finding common features among concepts or statements.

Listed groups of approaches for similarity computation are based on appropriate models.

The most common evaluation models include:

- feature-based models;
- semantic-network based models;
- models based on information content.

In **feature-based models** concept C is characterized by set of its features, denoted $ftrs(C)$. In [18] two groups of measures for such model are proposed:

1) contrast model where the similarity between two concepts C and D is defined by the linear function

$$contra(C, D) = \theta f(ftrs(C) \cap ftrs(D)) - \alpha f(ftrs(C) \setminus ftrs(D)) - \beta f(ftrs(D) \setminus ftrs(C)),$$

where \setminus is operation of sets difference, α , β and θ non negative constants, and $f(\cdot)$ expresses a number of features in the set;

2) a normalized model of the ratio where similarity is defined as quotient of the sets:

WordNet). It is an example of the case when similarity computation is based on measures of path-distance between concepts in the network. As concepts are in the taxonomy (linked by generic relations) so the similarity value between two concepts is computed by calculating edges on the path from considered concepts to their closer ancestor. If the entities are divided by only some connections then they are rated as similar. The more connections they share, the less similar they are [8, 19, 12, 20]. So, to evaluate similarity of

concepts C and D it is found the most specific is-a ancestor $E = \text{MSA}(C,D)$ of C and D and a similarity measure is computed as the sum of path distances from C to E and from E to D. More advanced estimates may take into account the depth of the concept $\text{MSA}(C, D)$, the density of the edges at the path nodes, and the weight of the edges.

In the **models based on information content** the information $pr(C)$ about the probability that entity is described by the concept is used as well as semantic network. This probability, as usual, is estimated based on an initial particular task.

The value of information content is measured based on the probability $pr(C)$ as $IC(C) =_{def} -\log pr(C)$. In [21] it is proposed the measure of the similarity of the concepts C and D based on probabilistic estimation of their MSA:

$$sim(C, D) =_{def} IC(\text{MSA}(C, D)) =_{def} -\log pr(\text{MSA}(C, D)).$$

In [22] it is proposed the measure of the path distance in the network based on their information content. It takes into account such factors as the depth and density of the edges of the path between the concepts is:

$$dist(C,D) =_{def} IC(C)+IC(D)-2IC(\text{MSA}(C,D))$$

In [18] it is proposed the similarity measure that defined by the ratio:

$$sim(C,D) =_{def} \frac{2IC(\text{MSA}(C,D))}{(IC(C)+IC(D))}$$

$$sim(C, D) =_{def} \frac{2 * f((C, D))}{2 * f(lsc(C, D)) + f(diff(C, D) + f(diff(D, C)))}$$

Note, the function f is a counter of properties, possibly weighted, in these measures.

Now, let consider the model of the semantic network. When the network is an hierarchy and the concept C has is-a ancestor: $U_1, U_2, U_3, \dots, U_{n-1}, U_n$, introduce the concept C^* such that $C := C^* \sqcap U_1 \sqcap U_2 \sqcap U_3 \sqcap \dots \sqcap U_{n-1} \sqcap U_n$. In the result T-Box the defined concept has the same hierarchy as initial nodes in the semantic network. Moreover, if the source network $U_1, U_2, \dots, U_n = \top$ is the path to the root of is-a hierarchy then the normal form of the concept U_1 in DL is $nf(U_1) = U_{(1)}^* \sqcap U_{(2)}^* \sqcap \dots \sqcap U_{(n-1)}^*$. Other words, if the network is a tree then concept's cardinality

Defining similarity values for DL descriptions of concepts

All metrics above are defined for atomic concepts. But these measures may be reformulated for complex DL concepts. Note, we suppose that the concept descriptions are represented in basic DL which support only operation of intersection of concepts. Any description of a complex concept may be normalized, namely, decomposed in such way that it will contain only atomic concepts. Usually, this is done simply by substituting the concept descriptions in the definition instead of non-atomic concepts. Denote as $nf(C)$ the set of atomic concepts which is in the normal form of the concept C. Note that $C \sqsubseteq D$ (where \sqsubseteq - structural subsumption), if $nf(D) \subseteq nf(C)$.

Taking into account given structural description of the concept, the measures above can be reformulated as follows.

For feature-based model, we will consider features of the concept as atomic concepts and a complex concept as conjunction of these atomic concepts. Considering the peculiarities of the intersection and the difference of the sets of atomic properties, the similarity measures, under the conditions of their symmetry, can be determined as follows:

$$\begin{aligned} contra(C, D) =_{def} & f(lcs(C, D)) \\ & - 0,5 * f(diff(C, D)) \\ & - 0,5 * f(diff(D, C)) \end{aligned}$$

is the normal form of the concept C - $|nf(C)|$ and it is equal to the distance of the path from node U_1 to the root. Paths from C and D to the root are intersected in $E = \text{MSA}(C,D)$, that is the same as $\text{LCS}(C,D)$ on the subsumption hierarchy. Then the distance between the concepts C and D may be defined as follows:

$$dist(C,D) =_{def} \frac{|nf(C)| + |nf(D)| - 2 * |nf(\text{LCS}(C,D))|}{|nf(C)| + |nf(D)|}$$

Respectively, for information models:

$$\begin{aligned} dist(C,D) & =_{def} IC(C)+IC(D)-2*IC(lcs(C,D)), \\ sim(C,D) & =_{def} \frac{2*IC(lsc(C,D))}{(IC(C)+IC(D))} \end{aligned}$$

Existential measures of the concept similarity

In the existential approaches a similarity value is calculated by counting joint instances of the concept extensions [26] or by measuring a content variation between concepts [27, 28, 29].

As a rule, an ontology has a structure which is more complex than simple taxonomy. So, similarity measures that are based on distances in the taxonomy or based on usage of MSA can't be applied.

Note that the semantic relation of subsumption is based on canonic interpretation of ABox and assumption of unique namespace (UNA) of DL. It follows that the interpretation of the instances of ABox are themselves, and different individuals, corresponding to different objects of the business-area, have different names in the namespace. So, we will determine the similarity measure based on their extensions in the canonical interpretation of DL [25].

Let \mathcal{L} is a set of concepts in DL \mathcal{ALC} , \mathcal{A} is ABox with the canonical interpretation \mathcal{J} . The semantic similarity of concepts s is a function: $s: \mathcal{L} \times \mathcal{L} \rightarrow [0,1]$, that is defined as:

$$s(C, D) = \frac{|I^{\mathcal{J}}|}{|C^{\mathcal{J}}| + |D^{\mathcal{J}}| - |I^{\mathcal{J}}|} * \max(|I^{\mathcal{J}}| / |C^{\mathcal{J}}|, |I^{\mathcal{J}}| / |D^{\mathcal{J}}|),$$

where $I = C \cap D$ and $(.)^{\mathcal{J}}$ is an extension of the concept in the interpretation \mathcal{J} .

The measure above may be justified as follows. If the concepts C and D are equivalent (both $C \sqsubseteq D$ and $D \sqsubseteq C$ are true) then $s=1$. If the concepts are different at whole and intersection of their extensions is empty, then the similarity value is a minimal, so it is equal 0. In the case of non-empty intersection of the concepts the measure has a value in the rank from 0 to 1. So, this measure expresses a degree of the similarity of the concepts C and D reduced on the value of $\max(|I^{\mathcal{J}}| / |C^{\mathcal{J}}|, |I^{\mathcal{J}}| / |D^{\mathcal{J}}|)$. This value presents a difference of these concepts. It means that the similarity is considered as value weighted respectively the similarity degree (it is not absolute value). This measure corresponds to a rather strict semantic relation between the concepts, which is provided by the subsumption.

Measures of GCS-similarity of the concepts

The measures of GCS-similarity are determined based on the term of GCS-cover. They may be applied in the cases when other measures, namely, ones based on concept extensions intersections, information content or path distance, don't work. The measures based on GCS also use the term of a concept extension but the similarity value is defined as variation of number of instances in the concepts' extensions relatively the number of instances in the extension of their super-concept instead of counting common instances of the concepts. The common super-concept is defined by GCS of the concepts and the measure w.r.t. TBox \mathcal{T} of \mathcal{ALC} is formally determined as follows.

\mathcal{T} is \mathcal{ALC} -TBox. \mathcal{L} is descriptive logic that include \mathcal{ALC} . C and D are concept descriptions in $\mathcal{L}(\mathcal{T})$. Then a semantic similarity measure \mathcal{S} is a function $\mathcal{S}: (\mathcal{T}) \times (\mathcal{T}) \rightarrow [0,1]$ that determined as follows

$$\mathcal{S}(C, D) = \frac{\min(|C^{\mathcal{J}}|, |D^{\mathcal{J}}|)}{|(GCS(C, D))^{\mathcal{J}}|} * \left(1 - \frac{|(GCS(C, D))^{\mathcal{J}}|}{|\Delta^{\mathcal{J}}|} * \left(1 - \frac{\min(|C^{\mathcal{J}}|, |D^{\mathcal{J}}|)}{|(GCS(C, D))^{\mathcal{J}}|} \right) \right),$$

where $(.)^{\mathcal{J}}$ calculates a concept's extension w.r.t. interpretation \mathcal{J} (canonical interpretation [2, 9]).

So, if two concepts are semantically similar they should have good common super-concept that is close to both concepts, namely, it is extension of super-concept which contains a lot of individuals that are common with initial concepts. In such case the value of the function is approaching 1. Vice versa, if the initial concepts are very different, then their GCS and their super-concept contains many instances that do not belong to the source concepts, i.e. the value of the similarity will approach 0. This measure doesn't require the intersection of the concepts and doesn't take into account the path distance between them. Moreover, to

avoid obtaining an incorrect value of similarity in the case when one concept is very similar to the super-concept and very different from another concept, which is compared, the minimal extension of concepts is considered in the measure's definition.

Defining the similarity measures

at knowledge and mixed levels

Recall that similarity metrics of knowledge and mixed levels are measures for determining values of matching individuals and an individual and a concept, respectively. Determining measures involving individuals is based on the term of the Most Specific Concept (MSC). We can compute MSC or its approximation for each instance in ABox. These terms are equivalent in some cases.

Let a and b are two instances of ABox, $A^* = MSC^*(a)$, $B^* = MSC^*(b)$. Then, semantic similarity measures may be applied to the descriptions of the concepts A^* and B^* , and a result value will express the degree of similarity of corresponding individuals:

$$\forall a, b: s(a, b) = s(A^*, B^*) = s(MSC^*(a), MSC^*(b))$$

Likewise, the value of similarity between the descriptions of the concept C and the individual a may be calculated by determining the approximation of MSC of the instances and further applying the similarity measure to the concept C and the approximation MSC^* of the instance a :

$$\forall a, C: s(a, C) = s(A^*, C) = s(MSC^*(a), C)$$

So, both measures are reduced to determining the similarity of the concept descriptions after preliminary approximation of instances. In this case, any of the above models can be used to calculate the value of similarity of concepts.

It should be noted that the complexity of the proposed methods depends on the complexity of the standard methods of inherits in DL.

Applying the similarity measures based on the DL ontology POGeometry (an example)

Consider the application of the measures of similarity of concept descriptions

based on their extensions in canonical interpretation DL on an example of the domain ontology POGeometry.

TBox of the domain ontology POGeometry:

Coordinate, GeometricFigure

Vertex \sqsubseteq has.XCoordinate

Vertex \sqsubseteq has.YCoordinate

XCoordinate \sqsubseteq Coordinate

YCoordinate \sqsubseteq Coordinate

Coordinate \sqsubseteq hasValue.NUMBER

Vector \sqsubseteq 2has.Vertex

Vector \sqsubseteq has.VectorLength

Vector \sqsubseteq has.VectorAngle

VertexLength \sqsubseteq hasType.NUMBER

VertexAngle \sqsubseteq hasType.NUMBER

Height \sqsubseteq hasType.NUMBER

EdgeLenth \sqsubseteq hasType.NUMBER

...

Polygon \equiv GeometricFigure \sqcap =has.Vertex \sqcap

=has.Vector

Circle \sqsubseteq GeometricFigure

Quadrangle \equiv Polygon \sqcap =4has.Vertex \sqcap

=4has.Vector

Triangle \equiv Polygon \sqcap =3has.Vertex =3has.

Vector

Polygon \sqsubseteq has.Vertex

Polygon \sqsubseteq has.Vector

Triangle \sqsubseteq =3has.Height

Square \sqsubseteq hasType.NUMBER

GeometricFigure \sqsubseteq has.Square

Circle \sqsubseteq GeometricFigure

ABox:

Triangle(ABC), Triangle(XYZ),

Triangle(A1B1C1), Triangle(B1C1D1),

Triangle(A1C1D1), Triangle(A1B1D1),

Triangle(X1X2X3), Triangle(X2X3X4),

Triangle(X3X4X5), Triangle(

X4X5X6), ..., Quadrangle(A1B1C1D1),

Polygon(X1X2X3X4X5X6.), Circle(O1),

Circle(O2)

Taking into account the definitions of the concepts Quadrangle and Triangle we may inherit the subsumption of the concepts Triangle \sqsubseteq Polygon and Quadrangle \sqsubseteq Polygon. So, all individuals of the concepts Triangle and Quadrangle are instances of the concept Polygon.

So, $|Polygon^J| = 47$, $|Triangle^J| = 29$, $|Quadrangle^J| = 17$.

Then, the similarity of the concepts Triangle and Polygon may be determined based on sets of their instances as follows:

$$\begin{aligned}
 & \text{Let } I = \text{Triangle} \sqcap \text{Polygon}, \text{ then} \\
 & s(\text{Polygon}, \text{Triangle}) \\
 &= \frac{|I^J|}{|\text{Polygon}^J| + |\text{Triangle}^J| - |I^J|} \\
 & * \max\left(\frac{|I^J|}{|\text{Polygon}^J|}, \frac{|I^J|}{|\text{Triangle}^J|}\right) \\
 &= \frac{29}{47 + 29 - 29} * \max\left(\frac{29}{47}, \frac{29}{29}\right) = \frac{29}{47} \\
 &= 0,62
 \end{aligned}$$

Taking into account that the interpretations of the concepts Triangle and Quadrangle have no intersection

$|I^J| = 0$, where $I = \text{Triangle} \sqcap \text{Quadrangle}$, their values of similarity by instances will also be equal 0. In this case, certainly, the feature-based similarity measures or similarity measures using the least common subsumer are more reliable.

It should be noted that shown example is based on basic DL which use only

the inter-section constructor, and TBox, in fact, is a taxonomy. So, LCS always exists for their concepts, and for any concepts C and D from this Tbox the statement $LCS(C,D) = MSA(C,D)$ is the true. Particularly, $\text{Polygon} = LCS(\text{Triangle}, \text{Quadrangle}) = MSA(\text{Triangle}, \text{Quadrangle})$.

The function of the similarity concepts based on LCS may be defined based on the path distances between concepts or based on intersections of extensions of corresponding concepts (their sets of individuals).

$$\begin{aligned}
 \text{dist}(\text{Triangle}, \text{Quadrangle}) &= \text{def} \\
 & |nf(\text{Quadrangle})| + |nf(\text{Triangle})| - \\
 & 2 * |nf(\text{lcs}(\text{Triangle}, \text{Quadrangle}))| = \\
 & |nf(\text{Quadrangle})| + |nf(\text{Triangle})| - \\
 & 2 * |nf(\text{Polygon})| = 2 + 2 - 2 * 1 = 2
 \end{aligned}$$

Using feature-based model the similarity measure is:

$$s(\text{Triangle}, \text{Quadrangle}) =_{\text{def}} \frac{2f(\text{ftrs}(\text{Triangle}) \cap \text{ftrs}(\text{Quadrangle}))}{f(\text{ftrs}(\text{Triangle})) + f(\text{ftrs}(\text{Quadrangle}))} = \frac{2 * 1}{3 + 3} = \frac{1}{3}$$

Taking into account that, in this case, $GCS = LCS = MSA$ the similarity value is:

$$\begin{aligned}
 & S(\text{Triangle}, \text{Quadrangle}) \\
 &= \frac{\min(|\text{Triangle}^J|, |\text{Quadrangle}^J|)}{|(LCS(\text{Triangle}, \text{Quadrangle}))^J|} \\
 & * \left(1 - \frac{|(LCS(\text{Triangle}, \text{Quadrangle}))^J|}{|\Delta^J|}\right) \\
 & * \left(1 - \frac{\min(|\text{Triangle}^J|, |\text{Quadrangle}^J|)}{|(LCS(\text{Triangle}, \text{Quadrangle}))^J|}\right) \\
 &= \frac{\min(|\text{Triangle}^J|, |\text{Quadrangle}^J|)}{|\text{Polygon}^J|} \\
 & * \left(1 - \frac{|\text{Polygon}^J|}{|\Delta^J|} * \left(1 - \frac{\min(|\text{Triangle}^J|, |\text{Quadrangle}^J|)}{|\text{Polygon}^J|}\right)\right) \\
 &= \frac{17}{47} * \left(1 - \frac{47}{49} * \left(1 - \frac{17}{47}\right)\right) = \frac{17}{47} * \frac{19}{49} \approx 0,14
 \end{aligned}$$

Conclusions

In this paper the analysis of semantic similarity indicators, classified by approaches and estimation models is carried out. Described measures use semantic reasoning such as, for example, instances checking of given ABox (it means calculating the concepts extensions). The internal complexity of expressive DL languages, such as ALC, causes the non-effectiveness of structural approaches to reasoning, so the definition of similarity functions is based on the use of The Set Theory. This allows the use of numerical approaches at the symbolic level of representation of DL.

The estimation models and similarity measures on different estimation levels are analyzed in the article. The main is defining similarity between concepts (models of conceptual level). The tasks of calculating the values of similarity between individuals or between an individual and a concept reduced to finding MSC for individual(s) and estimating similarity of appropriate concepts.

The most of described measures are built based on basic DLs which support only intersection constructor. But described approaches may be applied for any DL that provides basic reasoning services, namely: instances checking and MSC (approximation).

Proposed similarity measures may be useful for resolving a lot of different problems of different types, particularly big data problems such as, for example, information retrieval in the context of terminological systems of knowledge representation, data classification and categorization, etc.

References

1. Fellbaum, C. (Ed.). (1998). *Wordnet: An Electronic Lexical Database*. MA: MIT Press.
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: *The Description Logic Handbook*. Cambridge University Press (2003)
3. Staab, S., Studer, R., eds.: *Handbook on Ontologies*. International Handbooks on Information Systems. Springer (2004)
4. Thompson, K., Langley, P.: Concept formation in structured domains. In Fisher, D., Paz-zani, M., Langley, P., eds.: *Concept Formation: Knowledge and Experience in Unsupervised Learning*. Morgan Kaufmann (1991)
5. Haussler, D.: Learning conjunctive concepts in structural domains. *Machine Learning* (1989) 7–40
6. F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In T. Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 96–101. Morgan Kaufmann, 1999.
7. F. Baader, R. Sertkaya, and Y. Turhan. Computing least common subsumers w.r.t. a background terminology. In V. Haarslev and R. Möller, editors, *Proceedings of Proceedings of the 2004 International Workshop on Description Logics (DL2004)*. CEUR-WS.org, 2004.
8. R. Rada, H. Milli, E. Bicknell, M. Blettner, "Development and Application of a metric on Semantic Nets", *IEEE Trans. on Systems, Man, and Cybernetics*, 19(1): 17-30 (1989)
9. Mantay, T.: *Commonality-based ABox retrieval*. Technical Report FBI-HH-M- 291/2000, Department of Computer Science, University of Hamburg, Germany (2000)
10. Collet, C., Huhns, M.N., Shen, W.M.: Resource integration using a large knowledge base in carnot. *IEEE Computer* 24 (1991) 55– 62
11. Fankhauser, P., Neuhold, E.J.: Knowledge based integration of heterogeneous databases. In Hsiao, D.K., Neuhold, E.J., Sacks-Davis, R., eds.: *Proceedings of the IFIP WG 2.6 Database Semantics Conference on Interoperable Database Systems (DS-5)*. IFIP Transactions, North-Holland (1992)
12. Bright, M.W., Hurson, A.R., Pakzad, S.H.: Automated resolution of semantic heterogeneity in multidatabases. *ACM Transaction on Database Systems* 19 (1994) 212–253
13. Tversky, A.: Features of similarity. *Psychological Review* 84 (1977) 327–352
14. Jang, J., Conrath, D.: Semantic similarity based on corpus statistic and lexical taxonomy. In: *Proceedings of the International Conference on Computational Linguistics*. (1997)
15. 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* 11 (1999) 95–130

16. Weinstein, P., Birmingham, P.: Comparing concepts in differentiated ontologies. In: Proceedings of 12th Workshop on Knowledge Acquisition, Modelling, and Management. (1999)
17. Rodríguez, M.A., Egenhofer, M.J.: Determining semantic similarity among entity classes from different ontologies. *IEEE Transaction on Knowledge and Data Engineering* 15 (2003) 442–456
18. A. Tversky, “Features of Similarity”, *Psychological Review* 84(4): 327-352, 1977.
19. J. Lee, M. Kim, and Y. Lee. Information retrieval based on conceptual distance in is-a hierarchies. *Journal of Documentation*, 2(49):188–207, 1993.
20. D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In *Proceeding of the EON 2006 Workshop*, 2006.
21. P. Resnik, ”Using Information Content to Evaluate Semantic Similarity”, *Proc. IJCAI 1995* : 448-453
22. G. Miller & W.G. Charles, ”Contextual correlates of semantic similarity”, *Language and Cognitive Processes*, 6, 1-28, 1991.
23. W. Cohen, A. Borgida, H. Hirsh: “Computing Least Common Subsumers in Description Logics”, *AAAI 1992*: 754-760
24. R. Kusters & R. Molitor, “Computing Least Common Subsumers in ALEN”, *IJCAI 2001*: 219-224
25. Claudia d’Amato, Steffen Staab, Nicola Fanizzi, F. Esposito: “Efficient Discovery of Services Specified in Description Logics Languages”, *SMRR 2007*
26. C. d’Amato, N. Fanizzi, and F. Esposito. A semantic similarity measure for expressive description logics. In A. Pettorossi, editor, *Proceedings of Convegno Italiano di Logica Computazionale, CILC05, Rome, Italy, 2005*
27. C. d’Amato, N. Fanizzi, and F. Esposito. A dissimilarity measure for ALC concept descriptions. In *Proc. of the 21st Annual ACM Symposium of Applied Computing, SAC2006, 2006*.
28. P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.
29. A. Borgida, T. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In Horrocks, U. Sattler, and F. Wolter, editors, *Proceedings of the 2005 International Workshop on Description Logics (DL2005)*, volume 147 of *CEURWorkshop Proceedings*. CEUR-WS.org, 2005.

Received: 23.04.2021

About the author:

Olga Zakharova,

PhD,

Researcher.

Number of scientific publications in Ukrainian journals – 31. <http://orcid.org/0000-0002-9579-2973>.

Affiliation:

Institute of Software Systems of National Academy of Sciences of Ukraine.

Ac. Glushkov Avenue, 40. Phone.: 526 5139.

E-mail: ozakharova68@gmail.com. Mob.

phone: +38(068)594756

A.Ya. Gladun, K.A. Khala

ONTOLOGY-BASED SEMANTIC SIMILARITY TO METADATA ANALYSIS IN THE INFORMATION SECURITY DOMAIN

It is becoming clear that one of the most important resources to combat cyberattacks is the processing of large amounts of data in the cyber environment. There is also a need to automate the tasks of searching, selecting and interpreting Big Data to solve operational information security problems. For analyzing Big Data metadata, the authors propose pre-processing of metadata at the semantic level. As analysis tools, it is proposed to create a task thesaurus based on the domain ontology, which should provide a terminological basis for the integration of ontologies of different levels. For building task thesaurus, authors proposed to use the standards of open information resources. The development of an ontology hierarchy formalizes the relationships between data elements for machine learning, and development of artificial intelligence algorithms to adapt to changes in the environment, which will increase the efficiency of big data analytics for the cybersecurity domain.

Keywords: big data analytics, information security, cyber security, ontology, thesaurus, unstructured data, metadata, semantic similarity.

Introduction

Maintaining the growth and efficiency of enterprises while protecting confidential information is becoming increasingly difficult due to the ever-increasing threat of cybersecurity. The rise of cyberattacks is of great concern to both businesses and individuals. Also, the amount of information processed around the world has grown significantly over time, prompting cybersecurity to become more sophisticated and to introduce new methods of processing large amounts of data.

The use of big data itself can be incredibly useful, as it can not only help block any potential cyberattacks, but also help analyze vast amounts of data much faster and easier.

Obviously, data corruption prevention is one of the biggest big data challenges in cybersecurity. To make the most of big data, you need to know how to analyze it properly and use it to make the wisest.

Then cybersecurity big data analytics comes forward. It allows security professionals to analyze much more information and data than traditional cybersecurity solutions. Security systems use big data to automate the calculation of operations as correlation rules, which have the ability to dramatically reduce the number of false positives generated by the system.

The rapid growth in the popularity of big data analytics contributes to machine learning and deep learning, which are subsets of artificial intelligence. These teaching methods can process large amounts of data collected by the system and identify patterns that may indicate a cyber-threat. The challenge of safe big data is to analyze and process very large amounts of data in a timely manner to respond more quickly to incidents and obtain meaningful information that can be used by cybersecurity professionals.

The data itself faces big data challenges, which creates difficulties at every step from data collection to visualization and use. Thus, there is a need for a semantic context to access data and use and interpret results. For a semantic context, the same term can be represented differently, and therefore the result will depend on the context itself. However, you can find different concepts that represent the same object, or data that share a definition that is different from another. It is semantic technologies used to eliminate inconsistencies, evaluate and identify new information from existing knowledge bases, so it is advisable to consider different approaches that combine semantic technology with big data.

So Big Data is being used effectively today to make decisions in information se-

curity and cybersecurity systems. Big Data analytics allows you to make more informed decisions, ensure regulated implementation, and make recommendations to improve policies, guidelines, procedures, tools, and other aspects of network processes. The use of semantic modelling methods in Big Data analytics is necessary for the selection and combination of heterogeneous Big Data sources, recognition of patterns of network attacks and other cyber threats, which must occur quickly to implement countermeasures [1].

Metadata role's for interpretation Big Data

Big Data analytics in information security needs to solve the tasks of external units of Big Data. These data are used to predict and stop cyberattacks. Attack prevention and threat intelligence are becoming important for securing information systems and technologies.

In order for a data set to be considered big data, it must have one or more of the following characteristics: volume; speed; diversity; certainty; value [2]. Volume is the volume of data sets, i.e. the amount of data generated; speed (speed of formation and transmission of data) covers the structure, behaviour and permutation of data sets; diversity (type of structured and unstructured data) encompasses the tools and methods used to

process large or complex data sets. Reliability refers to the quality or accuracy of the data, which can cause data processing to eliminate errors and noise. Value is defined as the usefulness of data and it is intuitively related to reliability, because the higher the accuracy of the data, the greater their usefulness.

Metadata (see Fig.) for Big Data are blocks of data, both physically attached to big data and located externally from Big Data. These metadata provide information about the characteristics and structure of the Big Data sets: name; data origin, data source information; source; XML tags indicating the author and date of creation of the document; attributes indicating the size and formatting, control of the total amount; number of data set records; image resolution; brief description of data, etc. [3, 4].

It is important that metadata is machine-readable, as this helps maintain the origin of the data throughout the lifecycle of big data analytics, which helps to establish and maintain the accuracy and quality of the data.

Thus, there is a need for semantic analysis of Big Data metadata based on the development of methods for analyzing natural language (NL) metadata texts using the Big Data ontology, which formalizes the knowledge and features of the domain and allows for semantic processing, if necessary, other elements of big data metadata.

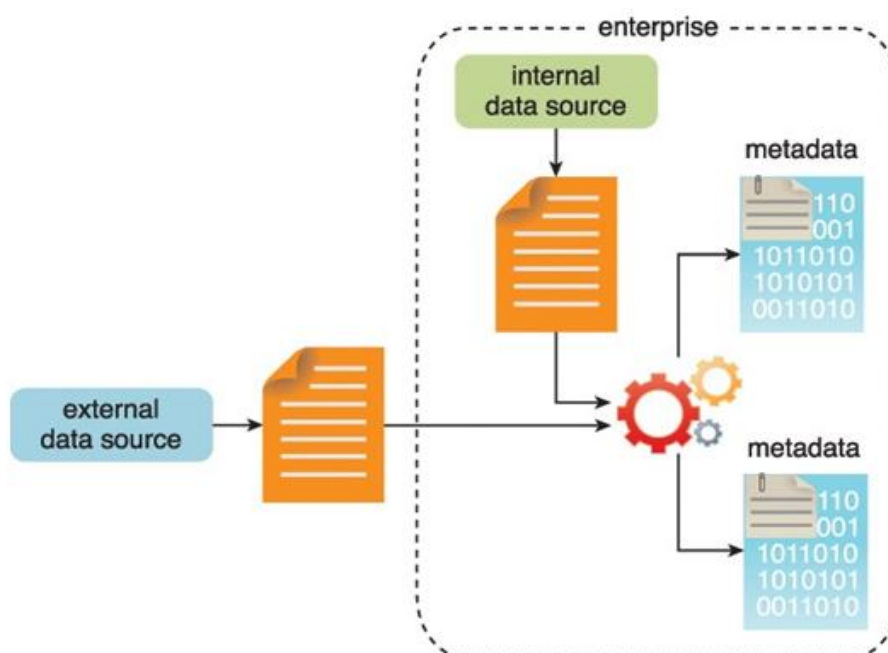


Fig.1. Metadata are adding to data from internal and external sources [1]

Ontological analysis for information security

The basis of the cybersecurity ontology is the need for a common language that includes basic concepts, complex relationships, and basic ideas. The most important feature of the cybersecurity ontology is that it illustrates the relationship between all the elements. By creating a correct and coherent cybersecurity ontology, cybersecurity professionals around the world can communicate effectively and develop a common understanding of important areas in this field.

Because cybersecurity ontologies are unique in that they cover the relationship between each entry in the ontology, this allows cybersecurity professionals to make faster and more accurate decisions. In addition, the ability to see the relationship between incidents, events and concepts provides valuable information.

Cybersecurity ontologies have become increasingly popular in recent years, as such a taxonomy allows cybersecurity professionals in different organizations or even in different countries to communicate faster and more efficiently, as well as to use their resources more efficiently. Also, ontologies can be very useful for describing critical vulnerabilities, risky vulnerabilities, and vulnerabilities that can harm organizations, employees, and regular users who use mobile devices.

Today, there are a large number of ontologies for information security that reflect various individual aspects of this subject area. For example, researchers have developed application ontologies to identify and classify network attacks: ontology for distinguishing network security status [5]; ontology of intrusion detection [6]; ontology for automated classification of network attacks [7]; ontology for predicting potential network attacks [8].

Other ontologies can provide an adaptive vocabulary that can improve behavioural analysis and help stop the spread of threats. Terms for such IS ontologies can be obtained from open sources, such as a dictionary of IS terms [9] and the standards of this subject area.

These ontologies describe the main artefacts of a cyber-attack to support the overall presentation of collected data and reuse, namely:

- the attacker's network environment, including its IP address, network size, range and name;
- the attacker's hosting environment, including information on the hosting operating system and its vulnerabilities, detected open ports, the level of the blacklist of the host in question based on its IP address, the number of virtual hosts;
- the type of organization where the attack will take place, as well as information about the location of the host with coordinates;
- type of attack based on its classification according to existing cyberattack dictionaries;
- date, day and time of the attack, taking into account the time zone of the attacker.

Clarifications or sub-concepts regarding countermeasures, assets, threats and vulnerabilities consist of a specific technical vocabulary. The dictionary was compiled from literature and security taxonomies. Ontology is implemented in OWL, where concepts are implemented as classes, relationships are implemented as properties, and axioms are implemented with constraints. In Fig. a fragment of such an ontology of upper level of cybersecurity is given.

To select and interpret the external blocks of Big Data, an ontology of the problem to be solved in the field of cybersecurity is used. The cybersecurity information system contains a hierarchical structure of interconnected ontologies: domain ontology, Big Data ontology, and task ontology. To select Big Data blocks, the task ontology can be replaced by a task thesaurus, which can be built by a Big Data ontology, you need to select a set of classes and a set of instances of classes. It is also advisable to highlight the relationship between attribute instances and their values. The following formal model was used to describe the ontologies of big data:

$$O_{BD} = \langle C, R, I, D_t, A \rangle \quad (1)$$

which contains the following elements:

$C = C_C \cup C_{In}$ – a set of ontology concepts, where C_C a set of classes, C_{In} a set of class instances;

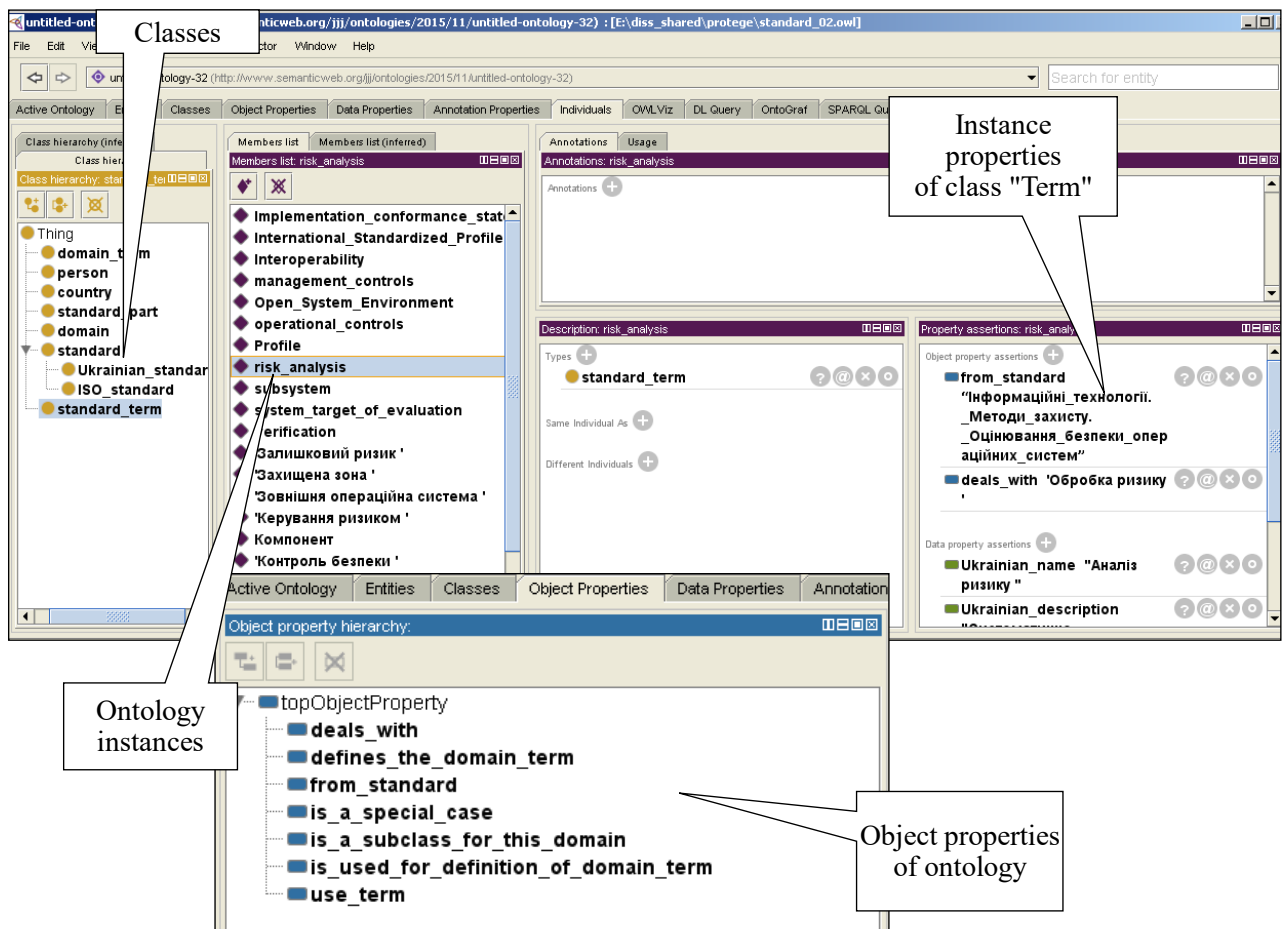


Fig. 2. The fragment of ontology of the cybersecurity [10]

$R = cr_{er} \cup \{or_i\} \cup or_{ie} \cup \{dr_j\} \cup dr_{er}$ – the set of relations between the elements of the ontology, where cr_{er} – hierarchical relations between ontology classes; $\{or_i\}$ – a set of object properties that establish relationships between instances of classes; or_{er} – hierarchical relations between ontology classes; $\{dr_j\}$ – a set of data properties that establish the relationship between instances of classes and values from D_i ; dr_{er} – hierarchical relations between the properties of these instances of ontology classes;

$I = \{I_C \cup I_P\}$ – a set of characteristics that can be used for logical inference over the ontology;

D_i – a set of data types for dr_j ontology class values; A – a set of rules.

Task thesaurus is a special case of the subject area ontology, which contains only ontological terms, but does not describe the semantics of the relationships between them in order to analyze NL texts. It can be automatically generated from the ontology of the subject area and the de-

scription of the problem in NL [11]. In the task thesaurus for concepts and relationships, a weight is introduced that indicates the degree of significance of a concept or relationship that improves the quality of model processing. The formal model of the thesaurus has the form:

$$T = \langle C_t, R_t, Inf \rangle \quad (2)$$

where $C_t \subseteq C$ – final set of terms; and $R_t \subseteq R$ – the final set of relationships between these terms, Inf – additional information about timing (e.g. weight).

The task thesaurus has a simpler structure because it does not cover ontological relations and for each concept has additional information as a weight $w_i \in W, i = 1, n$. Then the formal model task thesaurus is defined as a set of ordered pairs $T_{task} = \langle (c_i \in C_t, w_i \in W), \emptyset, Inf \rangle$ with more information in Inf regarding the source of the ontology. The algorithm of thesaurus generation for InRs has the following main stages:

Formation $Docs = \{doc_i\}, i = \overline{1, n}$, initial set $Docs$ from text documents doc_i related to InRs, where each document doc_i from the set $Docs$ has a weight factor W , which allows you to determine the importance of document elements for the InRs thesaurus.

Formation of the dictionary InRs $D(doc_i)$ for everyone doc_i , which contains all the words found in the document. Then the dictionary D_{Docs} formed as a sum $D(doc_i)$:

$$D_{Docs} = \bigcup_{i=1}^n D(doc_i).$$

Generation of InRs thesauri T_{res} , as projections of a set of ontological concepts C on the plural D_{Docs} . $T_{res} \subseteq C$. This processing step is aimed at removing terms from other non-user domains and stop words. The main problem at this stage is the semantic relationship of fragments of NL with T_{res} with the concepts of the set C domain ontology O_{BD} . It can be solved by linguistic methods that use lexical knowledge bases for each NL that go beyond this work. Each word from the thesaurus must be associated with one of the ontological terms. In the case of a lack of relations, the word is considered as a stop word or an element of notation, in which case it must be rejected.

A semantic bunch $R_{t_j}, j = \overline{1, n}$ is a group of thesaurus InRs associated with a single ontological term, used to train the semantics of documents written in different languages, and treated as $\forall p \in T_{res} \in R_{t_j}$, where $R_{t_j} = \{p \in D_{Docs} : Term(p) = c_j \in C\}$.

In the case where the domain ontology O_{BD} is not defined, we assume that the domain has no restrictions, and therefore does not remove any elements from the dictionary InRs: $T_{res} = D_{Docs}$.

Task thesaurus can also be generated based on InRs thesauri using set operations like sum, intersection and complement sets. Thus, a thesaurus of a particular domain can be formed as the sum of thesauri InRs related to that domain. The weight of a term for a given amount operation is defined as the sum of its weight in each InRs.

Methods for assessing semantic similarity for generating thesauri based on ontology

Semantic similarity is a field of research that is actively developing, which is

based on an attempt to calculate the relationship between words, concepts, sentences, paragraphs and documents. The similarity between two words is a measure of the probability of their meaning, calculated on the basis of the properties of concepts and their relationships in the ontology. Semantic similarity plays a fundamental role in information management, especially for unstructured data that in addition comes from a variety of flexible sources.

Semantic similarity is used to encompass similarity measures that use an ontology structure or external sources of knowledge to determine similarities between entities within one ontology or between two different ontologies. Potential applications of these measures are knowledge identification and decision support systems that use an ontology.

Semantic similarity concepts are a subset of the domain concepts that can be joined by some relations or properties. If domain is modeled by ontology then Semantic similarity concepts is a subset of the domain ontology concepts. There are several ways to build semantically similar concepts, which can be used separately or together. The user can define Semantic similarity concepts directly (manually – by choosing from the set of ontology concepts or automatically – by any mechanism of comparison of ontology with description of user current interests that uses linguistic or statistical properties of this description).

Semantic similarity concepts can join concepts linked with initial set of concepts by some subset of the ontological relations (directly or through other concepts of the ontology). Each semantically similar concept has a weight (positive or negative) which determines the degree of semantic similarity of the concept with the initial set of concepts.

A lot of different approaches used now to quantifying the semantic distance between concepts are based on ontologies that contain these concepts and define their relations and properties.

Factors related to the hierarchy of ontologies can affect the measurement of semantic distance: path length, depth and local density. Similarity measures and taxonomy are interconnected by taxonomic connec-

tions, i.e. the position of concepts in the taxonomy, the number of hierarchical connections and the information content of concepts are considered.

Approaches to calculating semantic similarity can be classified into the following main categories:

- by structure - approaches based on the structure or calculation of edges, semantic similarity based on taxonomic relations of the ontology hierarchy (is-a, part-of). They calculate the length of the path connecting the terms and the position of terms in the taxonomy. Thus, the more similar the two concepts, the more connections between the concepts and the more closely they are related [12] [13].

the shortest path is a simple, powerful measurement designed first and foremost to work with hierarchies. Where Max is the maximum path length between C_1 and C_2 in the taxonomy, and SP is the short path connecting C_1 with C_2 :

$$Sim(C_1, C_2) = 2 * Max(C_1, C_2) - SP \quad (3)$$

Hirst and St-Onge - measure HaS [14] calculates the relationship between concepts, using the distance between the nodes of concepts, the number of changes in the direction of the path connecting the two concepts, and the acceptability of the path. Where SP is the short path connecting C_1 to C_2 , d is the number of direction changes, C and k are constants. Thus, the longer the path and the more direction changes, the smaller the Sim :

$$Sim_{HaS}(C_1, C_2) = C - SP - k * d \quad (4)$$

Wu and Palmer - WaP measure [15] calculates the similarity, taking into account the depth of the two concepts in the taxonomies of WordNet, as well as the depth of LCS (Least Common Subsumer (LCS), the formula:

$$Sim_{WaP}(C_1, C_2) = 2 \times \frac{depth(LCS(C_1, C_2))}{depth(C_1) + depth(C_2)} \quad (5)$$

- by terms of information content - approaches based on the content of information use, the information content of concepts to measure the semantic similarity between

two concepts. The value of the information content of the concept is calculated based on the frequency of the term in this collection of documents.

Lin – Lin et al. [16] [17] proposed a measure based on an ontology bounded by hierarchical connections and corpus. This similarity takes into account the information between two concepts, such as Reznik [18], but the difference between them is in the definition:

$$Sim_{Lin}(C_1, C_2) = \frac{2 * \ln(p_{mis}(C_1, C_2))}{\ln(p(C_1)) + \ln(p(C_2))} \quad (6)$$

- by characteristics - characteristics-based approaches assume that each term is described by a set of terms that indicate its properties or characteristics. The degree of similarity between two terms is determined according to their properties or according to their relationship with other similar terms in the hierarchical structure. Tversky [19] takes into account the characteristics of terms to calculate the similarity between different concepts, ignoring the position and information content of terms in the taxonomy. Each term should be described by a set of words indicating its characteristics.

$$Sim_{TvsK}(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \alpha |C_1 - C_2| + (\alpha - 1) |C_2 - C_1|} \quad (7)$$

where C_1 and C_2 represent the corresponding sets of descriptions of the two terms. $\alpha \in E[0,1]$ is the relative importance of unusual characteristics. The value of α increases with commonality and decreases with the difference between the two concepts. The definition of α is based on the observation that similarity is not necessarily a symmetric relationship.

Many measures take into account only the path length between concepts. The basic idea of such estimates is that the similarity of the two concepts is a function of the path length that connects concepts (by taxonomic relation “is-a”) and their positions in the taxonomy. The same approach can be applied to arbitrary domain ontology where path between concepts can consist of all ontological relations.

Semantic similarity estimation parameters from various approaches (for example,

from (3)-(7)) can be used for generation of task thesaurus. We can consider such thesaurus as a set of concepts that have semantic distance from some initial set of concepts greater than some constant ones. In these estimations we can use different coefficients for universal and domain-specific relations R of domain ontology O_{BD} .

Conclusions

Prospects for automating the creation of thesauri based on ontologies depend on the availability of appropriate domain ontologies and well-structured, reliable InRs that characterize the needs and interests of users in information. Therefore, we can find InRs where such parameters are clearly defined and can be processed without additional pre-processing. Semantic Wiki, where the relationship between concepts and their characteristics is determined by semantic properties, meet the following conditions.

Big data is the best way to develop when it comes to cybersecurity, as identifying threats at the earliest opportunity becomes easier. Big data undoubtedly has advantages for any business that requires regular processing of large amounts of data. But despite this, the increasingly sophisticated methods used by cybercriminals are becoming increasingly difficult to combat. In large organizations with hundreds of employees, the system collects and analyzes huge amounts of data. Security professionals can use this information to predict trends and improve cybersecurity. With this in mind, it is safe to say that optimal approaches to cybersecurity should be used.

The semantic similarity was reviewed and its important role in the task of automating the creation of thesauri based on ontology was emphasized.

References

1. Erl T., Khattak W., and Buhler P.: Big Data Fundamentals: Concepts, Drivers & Techniques. Prentice Hall, ServiceTech press, 2016.
2. P. Buneman, S. Davidson, M. Fernandez, D. Suciu: Adding structure to unstructured data, In 6th International Conference on Database Theory, pp. 336-350. Delphi, Greece, 1997.
3. Smith K., Seligman L., Rosenthal A.: Big Metadata: The Need for Principled Metadata Management in Big Data Ecosystems. In Proceedings of the Company DanaC@SIGMOD, p. 46-55. Snowbird, UT, USA 2014.
4. Dey A., Chinchwadkar G., Fekete A., Ramachandran K.: Metadata-as-a-Service. In Proceedings of the 31st IEEE International Conference on Data Engineering Workshops, p.6-9. IEEE, Seoul, South Korea, 2015.
5. Salahi A., Ansarinia M.: Predicting Network Attacks Using Ontology-Driven Inference. In IJICTR, IGI Global, vol. 4, no. 2; pp. 27-35, 2012.
6. Bhandari P., Guiral M.S.: Ontology Based Approach for Perception of Network Security State. In Proc.of Recent Advances in Engineering and Computational Sciences, Chandigarh, pp.1-6, 2014.
7. Oltramari A., Cranor L.F., Walls R.J.: Building an Ontology of Cyber Security. In Proc. 9th Inter. Conf. on Semantic Technologies for Intelligence, Defense, and Security, Fairfax, pp. 54-61, 2014.
8. Wang J.A. and Guo M.: OVM. An Ontology for Vulnerability Management. In Proc. 5th Annu. Conf on Cyber Security and Information Intelligence Research, Knoxville, pp. 1-4, 2009.
9. Gladun A.Y., Puchkov O.O, Subach I.Yu., and Khala K.O.: English-Ukrainian dictionary of terms on information technology and cybersecurity. Kiev, Ukraine: NTUU KPI named by Igor Sikorsky, 2018.
10. Protégé 5.0. [Online]. Available: <https://protege.stanford.edu/>. Accessed on: Nov 24, 2020.
11. Gladun A., Rogushina J.: Use of Semantic Web Technologies and Multilinguistic Thesauri for Knowledge-Based Access to Biomedical Resources. International Journal of Intelligent Systems and Applications, №1, pp.11-20, 2012.
12. Rada R., Mili H., Bicknell E.: Development and application of a metric on semantic nets. In Proceedings of the IEEE transactions on systems, man, and cybernetics, p. 17-30, 1989.
13. Richardson R., Smeaton A., Murphy J.: Using WordNet as a knowledge base for measuring

- semantic similarity between words. Technical Report Working Paper CA-1294, School of Computer Applications, Dublin City University, 1994.
14. Hirst G., St-Onge D.: Lexical chains as representations of context for the detection and correction of malapropisms. In Proceedings of the WordNet: An electronic lexical database, vol. 305, p. 305–332, 1998.
 15. Wu Z., Palmer M.: Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, p. 133–138, 1994.
 16. Lin D.: An information-theoretic definition of similarity. In ICML, vol. 98, p. 296–304, 1998.
 17. Lin D.: Principle-based parsing without over-generation. In Proceedings of the 31st annual meeting on Association for Computational Linguistics, p. 112–120, 1993.
 18. Resnik P.: Semantic similarity in a taxonomy. An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)*, vol. 11, p. 95–130, 1999.

Received: 07.05.2021

About authors:

Gladun Anatoliy Yasonovych,
Candidate of Technical Sciences, Senior Research Fellow.

Number of scientific publications in Ukrainian publications - 188.

Number of scientific publications in foreign publications - 75.

<http://orcid.org/0000-0002-4133-8169>,

Khala Kateryna Oleksandrivna,
Researcher.

Number of scientific publications in Ukrainian publications - 31.

Number of scientific publications in foreign publications - 8.

<http://orcid.org/0000-0002-9477-970X>.

Affiliation:

International Research and Training Center of Information Technologies and Systems of National Academy of Sciences of Ukraine and Ministry of Education and Science of Ukraine.

03187, Kyiv,

Academician Glushkov Avenue, 40.

Tel: 044 502 6366.

E-mail: glanat@yahoo.com,

cecerongreat@ukr.net

R.D. Grygoryan, O.I.Yurchak, A.G. Degoda, T.V. Lyudovyk

SPECIALIZED SOFTWARE FOR SIMULATING THE MULTIPLE CONTROL AND MODULATIONS OF HUMAN HEMODYNAMICS

Special PC-based software simulating quantitative models of mechanisms that provide the overall control of human circulation is created. The autonomous software essentially expands the range of tasks concerning the modeling of cardiovascular physiology, in particular, of mechanisms controlling cardiac function, vascular hemodynamics, and total blood volume under unstable internal/external physiochemical environments. The models are verified on data representing hemodynamic responses to certain physical tests. The user-friendly interface provides all stages of preparation and analysis of computer simulation. The PC-based simulator can also be used for educational purposes.

Key words: physiology, cardiovascular system, acute and long-term control, model, simulator.

Introduction

Modern technologies providing the quantitative mathematical (computer) modeling of human physiological systems and the creation of specialized simulators (SS) [1-3] meet three independent problems. The first one is how to choose the right physiological concept to be modeled. The second one is how to apply adequate mathematics. The last problem concerns programming technologies (PT). Especially, a user-friendly interface (UI) development finally plays the most decisive role in the “fate” (usability) of SS.

So, before starting the code creation, a big hidden pre-work has been done. The chosen biological concept is the main stage result. Usually, programmers, involved in complex projects concerning biology are not aware of serious problems arising on this stage of the work. The matter is that in contrast with the commonly modeled objects (e.g. in industrial technologies), the model of every physiological system reflects both the level of science and the fact of simultaneous existence of competitive conceptual views of the same biological object or process. So, the team leader should have a high level of professional knowledge and proper biological background for choosing the most matching physiological concept.

The cardiovascular system (CVS) is one of the most often modeled human physiological systems [1-6,]. Purposes of the modeling determine both the complexity of CVS model and the modeling method. The lumped-parametric approach is the mostly

used method for CVS’s modeling [1-3]. If the number of vascular compartments in the model is properly large, this method provides the accuracy of regional circulation simulation. However, the larger this number is the smaller the numerical simulation step is and the larger calculations are. Therefore, models created for numerical simulations on PC have to match the capabilities of current PCs. On the other hand, every model has to be in some manner identified and validated for control situations. The detailed presentation of the vascular net meets with the problem that data required for the model validation are often absent. Both these opposite demands result in a workable compromise model. Another big problem of CVS’s modeling concerns the endogenic mechanisms. The matter is that not all the concerned mechanisms are circulation controllers. Some of the feedback mechanisms really control CVS’s parameters while others only modulate them. In order to investigate the physiological roles of mechanisms directly or indirectly modulating the current states of the heart and vasculature, the modeler needs a proper physiological concept concerning goals of so-called control mechanisms. The biggest problem is that there is yet no commonly accepted concept. Therefore, the concept accepted by the modeler usually reflects his subjective motives.

This publication aims to illustrate both internal connections of the triad of problems and the ways of these problems

solution during the special software development. Commonly accepted physiological concepts concerning arterial pressure neural-hormonal control are modified by new ideas proposed in [9,10].

A short description of the basic model

SS is based on a complex quantitative mathematical model which presents the human CVS as an open system interacting with a certain number of associated physiological systems (APS). Within the framework of traditional physiology some of these APS are known as circulation controllers. They could influence the total blood volume dynamics and current values of CVS's parameters.

The core model and models of certain APS are described in [5-9] while models of additional APS are described in this paper.

It is worth to stress here that the core model of hemodynamics contains of two sub-models separately describing pump functions of the right and left ventricles respectively. The third sub-model describes hemodynamics in a vascular net of 21 lumped-parametric arterial and venous compartments. Each vascular section possesses its own constant initial characteristics (rigidity, unstressed volume, and resistance between neighboring lumped-parametric vessel sections), [5]. In addition, the core model assumes the total blood volume is constant. But in the complex model, the total blood volume is under influences of certain endogenic mechanisms. Namely, each of the so-called control mechanisms is able to alter the value of at least one of the cardiovascular parameters included in the core model.

Both ventricles are modeled similarly as a flow generator: it connects the mean output flow of ventricles with their constant parameters and variable mean input pressure [5]. The transforming function of the ventricle takes into account its structurally determined constants, as well as its inotropic state. The stable heart rate on its autonomous level transforms ventricles' output to a heart output in ml/sec. So, the heart model does not imitate cardiac pulsatile behavior but is a model relative to mean values of pressures and flows.

Figure 1 represents structure of the basic model. There are eight mechanisms each of

them specifically influences (controls) parameters of a core model representing CVS.

In our complex model, main simulations do cover long time periods (tens of minutes, hours, and days), thus the pulsatile model of a heart pump function (HPF) [4] is substituted by the model describing HPF as a generator of mean blood flow.

In fact, the HPF is substituted by two similar models describing quasi-static relationships of ventricular input – output variables depending on much more inertial immanent characteristics of right or left ventricle. Input variables are central venous ($P_v(t)$), and lung vein ($P_{lv}(t)$) pressures. Ventricles' end-diastolic capacity ($C_r(t), C_l(t)$), output valve resistances ($R_{ovr}(t), R_{ovl}(t)$), and inotropic coefficients ($k_r(t), k_l(t)$) are connected with their input pressures ($P_v(t), P_{lv}(t)$) and output flows ($Q_r(t), Q_l(t)$) by identical functional relationships of:

$$Q_r(t) = f_r(P_v(t), C_r(t), R_{ovr}(t), k_r(t)),$$

$$Q_l(t) = f_l(P_{lv}(t), C_l(t), R_{ovl}(t), k_l(t)).$$

The principle is that the core model of CVS has six independent channels flows through which can alter the total blood volume.

In this version of the complex model, only for some of these channels the resistance is internally connected with activities of associated physiological control mechanisms (PCM). In the remaining hemodynamic channels, values of hydrodynamic resistances can be directly set through UI.

Models of seven independently acting PCM have been created:

- Mechanoreceptor reflexes initiated by receptors located in aortic arc, carotid sinuses, lung artery, and brain arteries of the circle of Willis;

- Peripheral chemoreceptor reflexes activated under alterations of arterial blood PaCO₂, PaO₂, and pH;

- Central and local renin-angiotensin-aldosterone mechanism: it is a negative feedback mechanism activated under low local flows in kidneys or other organs. Vascular tones are the main effectors.

- Antidiuretic hormone mechanism: this model reciprocally connects the increased concentrations of aldosterone with decrease of the rate of diuresis.

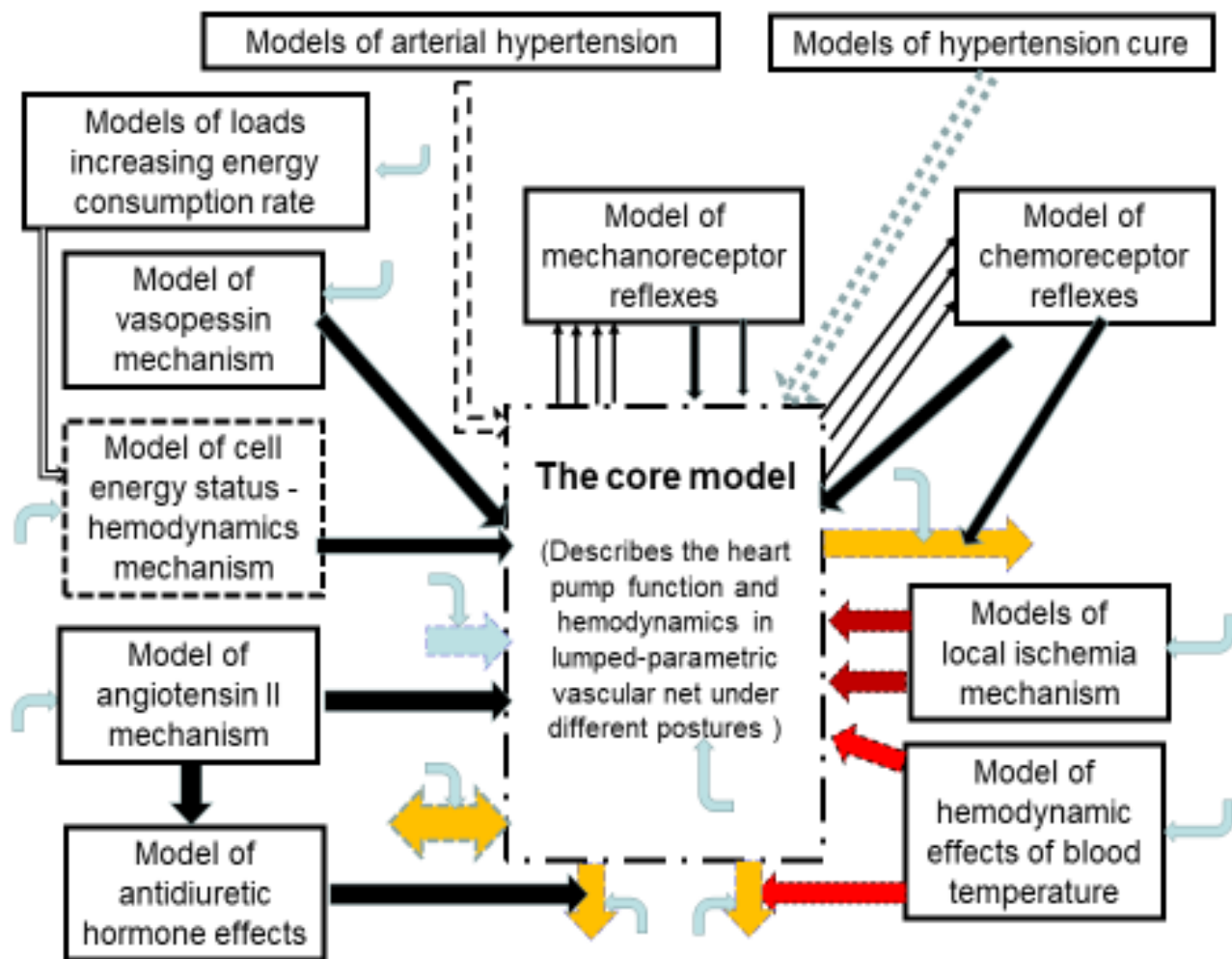


Fig.1. Structure of complex model necessary and sufficient for simulation of mechanisms controlling or modulating human hemodynamics under external / internal influences.

- Mechanisms activated under energy deficiency (EDM).
- Regional ischemia.
- Models of blood temperature influence on hemodynamics.

Special attention should be drawn to the model for the first time describing EDM. In human cells, as in every unicellular aerobic organism, there are two ways for a synthesis of energy (molecules of ATP). One way providing less than 5% of ATP molecules is their anaerobe synthesis using glucose (glycolysis). The byproduct of this synthesis is pyruvate which enters into mitochondria and its oxygenation provides about 95% of ATP. So, in ATP production oxygen, along with carbohydrates, plays the main role. Under low or moderate deficiency of ATP, oxygen-associated regulator effects can be modeled through chemoreceptor reflexes [6]. Their effects on lung ventilation and hemodynam-

ics are well-known. However, under acute severe energy deficiency, the chemoreceptor increases of arterial pressure and blood flows are not sufficient for overcoming the inhibited metabolism in big groups of cells. There are additional enhancers of ATP synthesis.

The mechanism of AMP-activated protein kinase is a way for compensating the glucose lack [9]. But more important is the glucagon-glucose mechanism activating liver glycogen back-transformation to blood glucose. In parallel with this, under glucose lack a peptide (glucagon), entering into blood, activates both the chronotropic and the inotropic states of the heart [9]. So, EDM has an additional opportunity to modulate the circulation. The last effects have been modeled for the first time as follows.

$$T_G \frac{dG(t)}{dt} = G(t) + I(t) - \varpi_1 \cdot W(t)$$

$$T_g \frac{dg(t)}{dt} = \begin{cases} \varpi_2 \cdot (G(t) - G_c) - g(t) - g_u, & G(t) \geq G_c \\ \varpi_3 \cdot (G_c - G(t)) - g(t), & G_c \geq G(t) \end{cases}$$

In these equations, $G(t)$ and G_c represent current and critical blood glucose concentrations, $W(t)$ is the total load, $g(t)$ is the concentration of blood glucagon, ϖ_1, ϖ_2 , and ϖ_3 are approximation constants, T_G and T_g characterize inertia of glucose-glucagon transformation mechanism.

$$\Delta k_g(t) = \varpi_4 \cdot (q_c(t) - q_c(0));$$

$$k(t) = k(0) + \Delta k_g(t);$$

$$\Delta F_g(t) = \varpi_5 \cdot q(t);$$

$$R_c(t) = R_c(0) \cdot (1 - \varpi_6 \cdot q_c(0) / q_c(t))$$

Here again ϖ_4, ϖ_5 and ϖ_6 are approximation constants, $\Delta k_g(t)$ is the glucagon-caused increase of the heart inotropic state $k(t)$ from its initial value of $k(0)$, $q_c(0)$ $q_c(t)$ represent initial (normal) and current values of coronary blood flows, $R_c(0)$ $R_c(t)$ -appropriate resistances of coronary arteries. $\Delta F_g(t)$ is the glucagon-influenced increase of the heart rate.

Inotropic states of left ($k_l(t)$) and right ($k_r(t)$) ventricles assumed to be the same: $k_l(t) = k_r(t) = k(t)$

Each regulator has its part in summary shifts of $F(t)$, $k_r(t)$ and $k_l(t)$, as well as in shifts of compartmental values of $D_m(t)$ and $U_m(t)$, therefore the complex model is capable to simulate the main endogenic modulations of both central and regional circulation.

$$F(t) = F_a + \sum_j \Delta F_j(t);$$


$$D_m(t) = D_m(0) + \sum_m \Delta D_m(t);$$

$$U_m(t) = U_m(0) - \sum_m \Delta U_m(t);$$

The core model was created in assumption that CVS's characteristics including the total blood volume $V_T(t)$ are constant. Above it was shown that both heart and vascular characteristics are under regulators influences. It is known that under long-time observations, the value of $V_T(t)$ is also altered. Alterations depend on changes of the balance between liquid inflows into CVS and outflows from CVS. So, CVS is an open system interacting with multiple organs (kidneys, skin, lungs), intercellular space, as well as with the

digestive system. Most of these participants are under specific regulators that have not been yet definitely described in the complex model. At the same time, empty arrows in Fig.1 depict the fact that the UI provides the user with direct alterations of certain arbitrarily chosen liquid flows (Δq_n). Thanks to such manipulations, the physiologist will be able to watch hemodynamic effects induced by each such alteration.

$$\frac{dV_T(t)}{dt} = \sum_n \Delta q_n(t)$$

In Fig.1, arrows like  indicate that the corresponding model also has direct inputs provided by UI. Combinations of multiple arrows illustrate that values of associate input or output flow can be directly modulated by means of UI. These temporarily implemented additional options have been used to provide first (rough) assessments concerning the likely contribution of real mechanisms modulating these flows in the intact organism.

Simulation algorithms

A single simulation algorithm (SA) depends on: 1) actual configuration of physiological models (ACPM); and 2) actual group of input loads (AGIL). This can be illustrated by means of Fig.2 which represents the general view on SA.

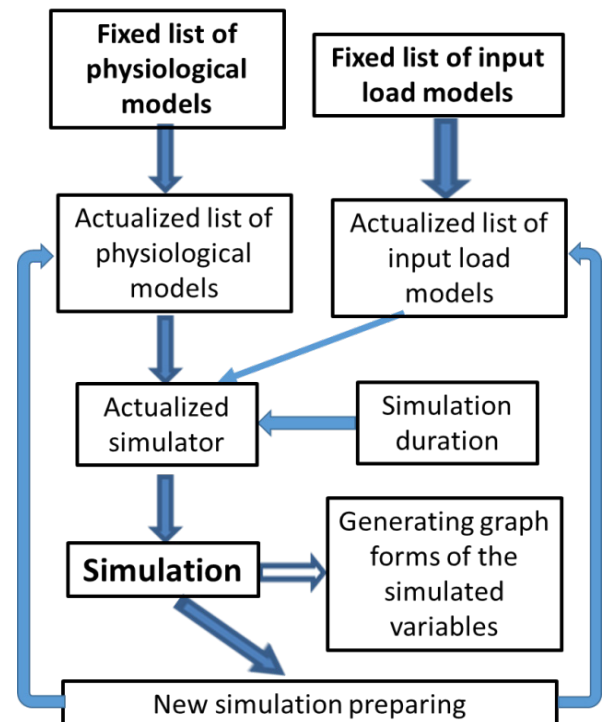


Fig.2. Simulation algorithms.

According to this algorithm, two independent procedures have to be performed before the simulator is ready to execute calculations. As a result of the first procedure the user gets the actualized ACPM. The second procedure generates AIGL. Additionally, the user should set the simulation duration. Changing at least one value in characteristics of ACPM and/or AGIL, the user can start next simulation.

Potentially, our simulator consists of 12 independently functioning physiological models and 10 models each representing one dynamic input load. So, the number of actualized ACPM and AGIL is too large. In fact, no empirical physiologist has ever observed hemodynamic effects of entire scenarios provided by our simulator. The user will be able to run and analyze the entire spectrum of simulations, he / she will be provided by an effective user interface.

Input loads

Our models and the entire SS imitate dynamic physiological responses of a healthy person to dynamic input loads. Namely, the response depends on the absolute level and shape of the applied load. Theoretically, it is possible to create a simulator providing the construction of every arbitrary load profile. Currently, our SS provides only fixed input loads each having its specific shape. At the same time, the shape is set based on the common trapezoidal shape formally describing the input load $W(t)$ as:

$$W(t) = \begin{cases} 0, & t < T_b; t > T_e \\ v_u \cdot (t - T_{p1}); & T_b \leq t \leq T_{p1} \\ W_{\max}; & T_{p1} \leq t \leq T_{p2} \\ W_{\max} - v_d \cdot (t - T_{p2}); & T_{p2} \leq t \leq T_e \end{cases}$$

where

$$T_{p1} = T_b + W_{\max} / v_u;$$

$$T_e = T_{p2} + W_{\max} / v_d.$$

Here T_b is time to start the loading, T_{p1} - time when the maximum load W_{\max} is reached; v_u is the load increasing velocity, v_d is the load decreasing velocity, T_{p2} is the end time of the load plateau, T_e is the exposure end time.

All other loads (changes of total blood volume, blood temperature, occlusion of local artery) have similar shape or its reduced versions.

Controlled linear alterations of total blood volume (V), namely ($\pm \Delta V$), are provided according to formulae:

$$\Delta V(t) = \begin{cases} V_{ab}(t) \pm v_{al}; & T_{b\Delta V} < t < T_{b\Delta V} \pm \Delta V / v_{al} \\ V_{ab}(t); & t \leq T_{b\Delta V}; t \geq T_{b\Delta V} \pm \Delta V \end{cases}$$

where $V_{ab}(t)$ is abdominal vein volume, $T_{b\Delta V}$ is the start time for the altering of total blood volume with the velocity v_{al} .

Blood temperature ($T^o(t)$) alterations ($\pm \Delta T^o$) alter almost linearly the heart rate $F(t)$ and regional vascular diameters. These effects have been modeled by us. In order to offer the user an access to these mechanisms, additional formulas describing activation (deactivation) of these mechanisms are needed. In our current SS, the incorporated formulas provide setting of numerical values of normal blood temperature (T_N^o) and stable velocity of temperature's elevation ($+v_T$) until the maximal (T_{\max}^o) level is reached:

$$T^o(t) = \begin{cases} T_{\max}^o; & T^o(t) \geq T_{\max}^o \\ T_N^o + v_T \cdot t; & t_{bT} < t < t_{eT} \end{cases}$$

$$t_{eT} = (T_{\max}^o - T_N^o) / v_T$$

By analogy, under temperature lowering with stable velocity of ($-v_T$), and maximal (T_{\max}^o) or minimal (T_{\min}^o) levels:

$$T^o(t) = \begin{cases} T_{\min}^o; & T^o(t) \leq T_{\min}^o \\ T_N^o - v_T \cdot t; & t_{bT} < t < t_{eT} \end{cases}$$

$$t_{eT} = (T_N^o - T_{\min}^o) / v_T$$

The tilt test is an exclusive dynamic load connected only with the angle ($A(t)$) that a body forms with the horizontal axis. Simulations of this test are possible due to the fact that in the basic model of hemodynamics, one of determinants of blood flows is the gravitational factor:

$$G_i(t) = \gamma \cdot H_i \cdot \sin(A(t)).$$

Here γ is a constant transforming measures in cm to mm Hg, index "i" concerns each vascular compartment located at a distance of H_i from the zero level (feet).

To imitate gradual ischemia of regional (kidney, coronary, or brain) artery, two parameters are used. One determines the magnitude of local resistance increase, the other one determines its speed. Formally, the ischemia simulation is modeled as:

$$R_1(t) = \begin{cases} R_1 0 \cdot \left(\frac{V_i 0}{V_i(t)} \right)^2, & t < T_{bl} \\ R_1 0 \cdot (1 + \Delta_i(t)) \cdot \left(\frac{V_i 0}{V_i(t)} \right)^2, & T_{bl} < t < T_{el} \\ R_1(T_{el}), & t > T_{el} \end{cases}$$

where $\Delta_i(t) = \Delta_i^{\max} \cdot (t - T_{bl})$,

Constants Δ_i^{\max} , T_{bl} and T_{el} will be set through UI.

Requirements to user interface

According to the stated goal, our SS is a software-modeling tool (SMT). So, SMT does give the physiologist-researcher the ability to simulate almost the entire spectrum of events and situations that have ever been proposed as cardiovascular functional tests. The most known functional tests are:

- 1) Exogenous dynamic alterations of total blood volume;
- 2) Postural tests for different tilting angles;
- 3) Dynamic physical aerobic loads of given profiles;
- 4) Alterations of blood temperature;
- 5) Heart myocardium ischemia;
- 6) Brain ischemia;
- 7) Kidneys ischemia;
- 8) Shutdown of any number of control mechanisms (including the extreme case of the uncontrolled CVS).

All items on this list excluding postural tests can be combined. So, specialized UI does easily provide the physiologist with a computer experiment (simulation) very similar to experiments provided on a natural organism. Each experiment has to be specially marked for further independent analysis. Therefore, special simulation passport is necessary. The passport contains information about the characteristics of the experiment, namely, the model configuration, parameters of tests, experiment's duration, human body position.

Certainly, numerical characteristics of models do not cover the entire diapason of the regulator mechanism or possible values of test parameters. Our SMT will adequately simulate physiological responses only within certain boundaries that were verified in special investigations. Most problems, asso-

ciated with the tuning and the verification of models' constants, were mainly solved by means of special software and interface described in [11]. That software provided accesses to about 450 parameters of models. But in the current software, only the most informative physiological characteristics are explicated in graph forms. Namely, the list of these characteristics includes those variables that physiologists usually try to observe and analyze in their traditional experiments. Seven groups collecting these characteristics are designed.

The first group consists of graphs representing dynamics of mean arterial pressure, systolic and diastolic arterial pressures in the aortic arch, the mean arterial pressures in the area of carotid sinus, in brain, in kidneys, in lungs, mean venous pressures in lungs veins and in central vein. Besides, the heart rate is also included in this group.

The second group is formed of graphs showing dynamics of flows. Here are collected outputs of right and left ventricles, summary flow directed to the head and its separation to flows in hands and brain, flows into abdominal organs, kidneys, legs.

The third group represents graphs of six sectional blood volumes. Body sections are associated with cavities and lungs. Such information is important for assessing the power of the regulators in different postures of a person.

The fourth group consists of graphs representing variables related to the chemoreceptor reflex. The group connects three input variables (pH, PaCO₂, and PaO₂) with afferent impulse patterns into the brain structures that, via modulating of efferent sympathetic and parasympathetic impulse patterns, alter the lung ventilation, blood concentration of hemoglobin, as well as the state of CVS. The matter is that pH, PaCO₂, and PaO₂ depend on the total rate of energy consumption in cells.

The fifth group represents graphs of baroreceptor activities in four arterial zones: aortic arch, carotid sinus, Willis's circle, and lungs artery.

The sixth group consists of graphs representing blood concentrations of glucose, of glucagon, of angiotensin-II, and of the antidiuretic hormone.

The seventh group collects of graphs representing characteristics of right and left ventricles.

General view of the main window of SS is shown in Fig.3. Namely, this window contains main commands, necessary for both preparing and executing a simulation. In addition, the window also provides the user with capabilities to look simulation results.

Simulations preparing and execution

Every simulation is an independent computer experiment with a previously collected configuration of models. Operations needed to prepare a computer simulation, as well as its executing and results analyzing, are listed in the window located on the left side of the UI, shown in Fig.3. Information concerning details of every chosen string is indicated in the right side of the UI window.

Model configuring is a multi-step operation aimed to create the desired combination of activated regulator mechanisms, tests to be applied, and simulation duration. Additional

opportunities for models activation or deactivation are provided through the windows shown in right sector of the main window. Some of these windows are pop-up windows. An example of the pop-up window designed to actualize parameters of baroreceptor reflexes is shown in Fig.4.

Simulation (when activated) will last until the exposure time is over. All simulation results are saved in the operative memory thus this parameter of PC is critical for determining the maximal simulation duration.

Our simulator supports the creation of multiple biological model versions each of which is capable of providing hemodynamics under a single or more chosen input loads. In fact, these manipulations imitate empirical methods of certain control mechanisms deactivation (activation).

Special window providing tests choice is shown in Fig.5.

Below is the window for preparing simulations under a special test of aerobic load. Pay attention that the values of $W(t)$ are given relatively to the zero level which con-

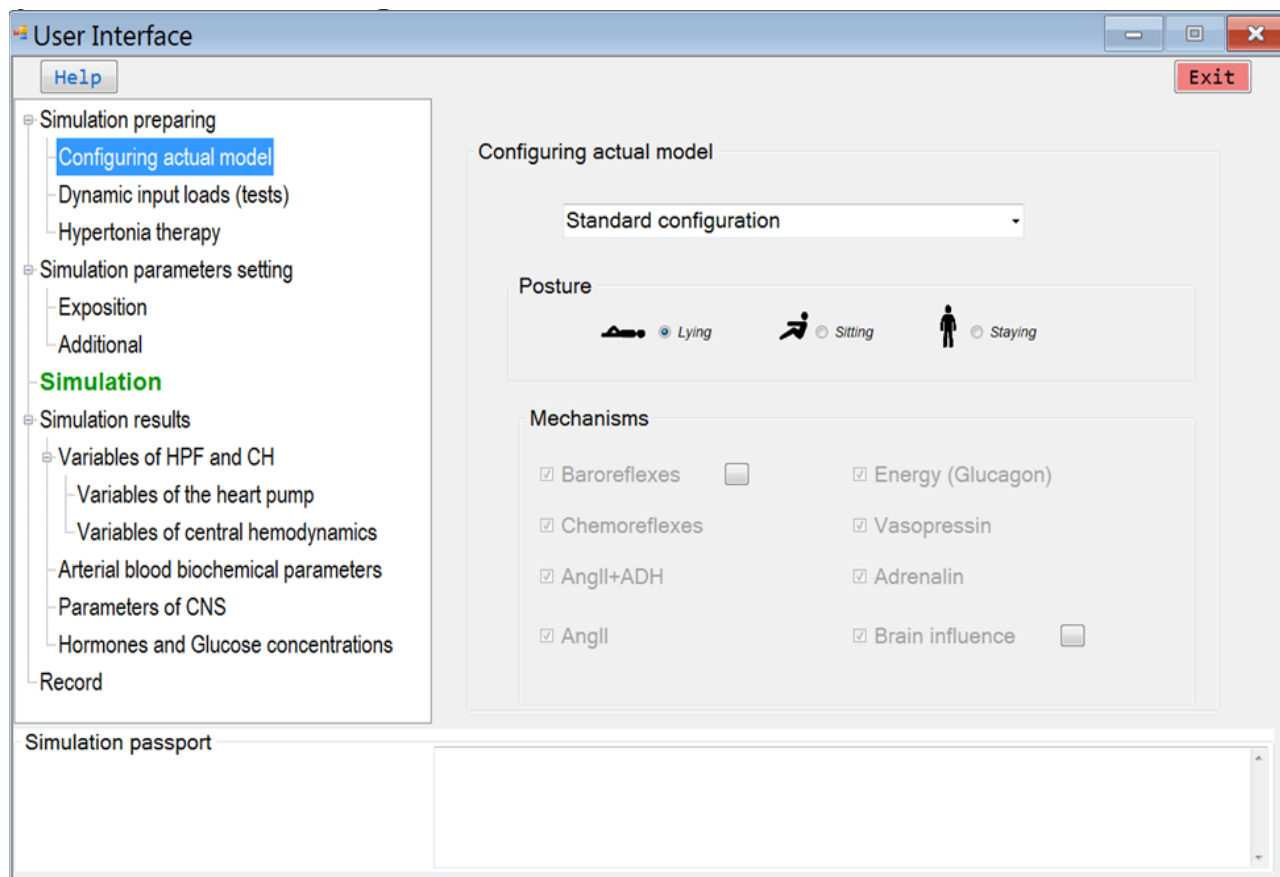


Fig.3. User interface in case of regulators' standard configuration.

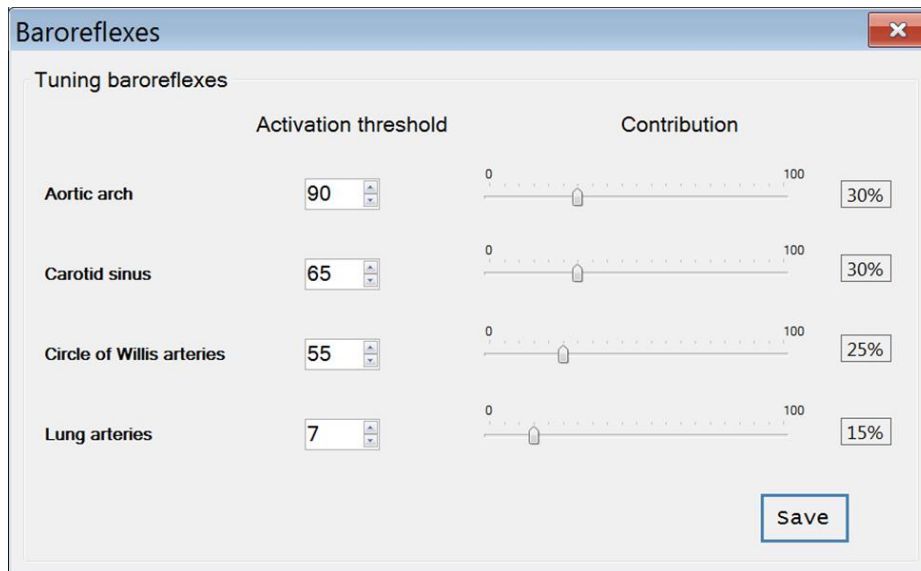


Fig.4. User interface special window for setting activation thresholds of each reflex, as well as its relative contribution to summary baroreceptor reflex.

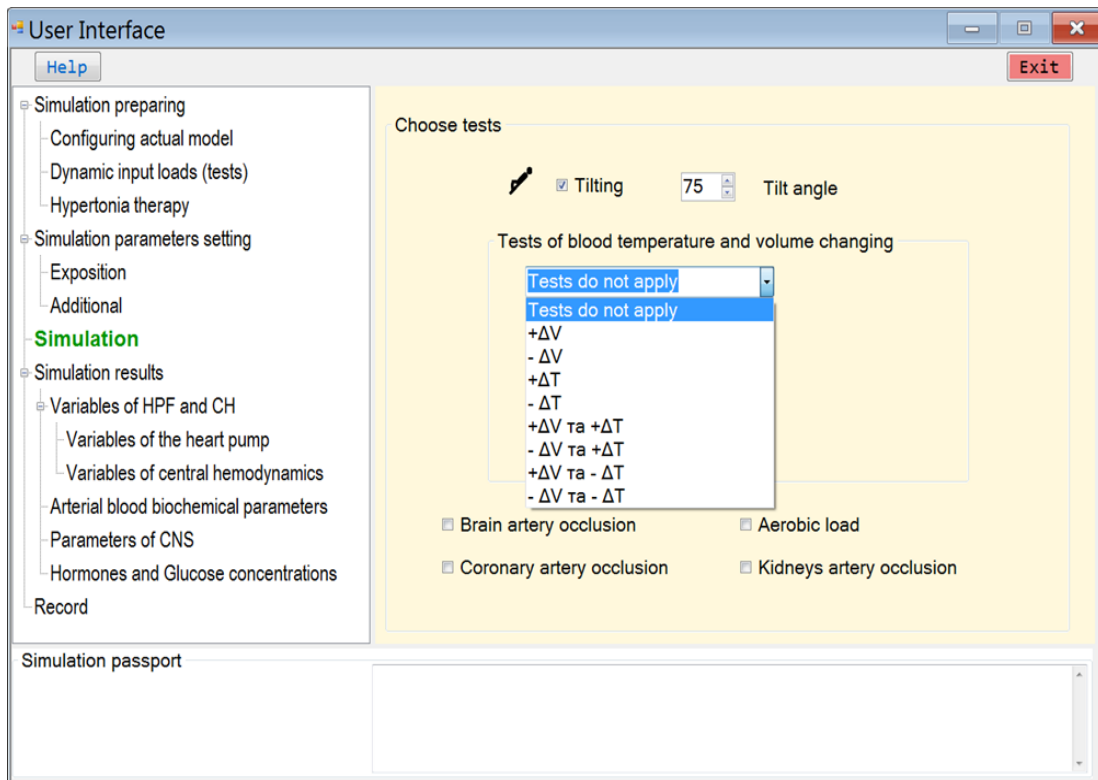


Fig.5. User interface window in case of tests performing.

ventionally represents organism's summary energy expenditures needed for providing the rest physiology in human body horizontal (clinostatic) position. The user can arbitrarily set the load starting time (TW_{begin}), the maximal excess load (in % to the rest level energy expenditures), the load's increasing speed (vW_{up}). These three parameters determine the time moment (TW_{Plato1}) when the load

reaches its W_{max} . Parameter of T_{stab} , characterizing stable load, is also set by the user, so, the time momentum TW_{Plato2} for starting load decreasing with a user-defined speed of vW_{down} will determine the time momentum TW_{end} for finishing the loading. For $t > TW_{end}$ values of $W(t)=0$. Despite this organism's responses will last until hemodynamic parameters return to their initial rest values.

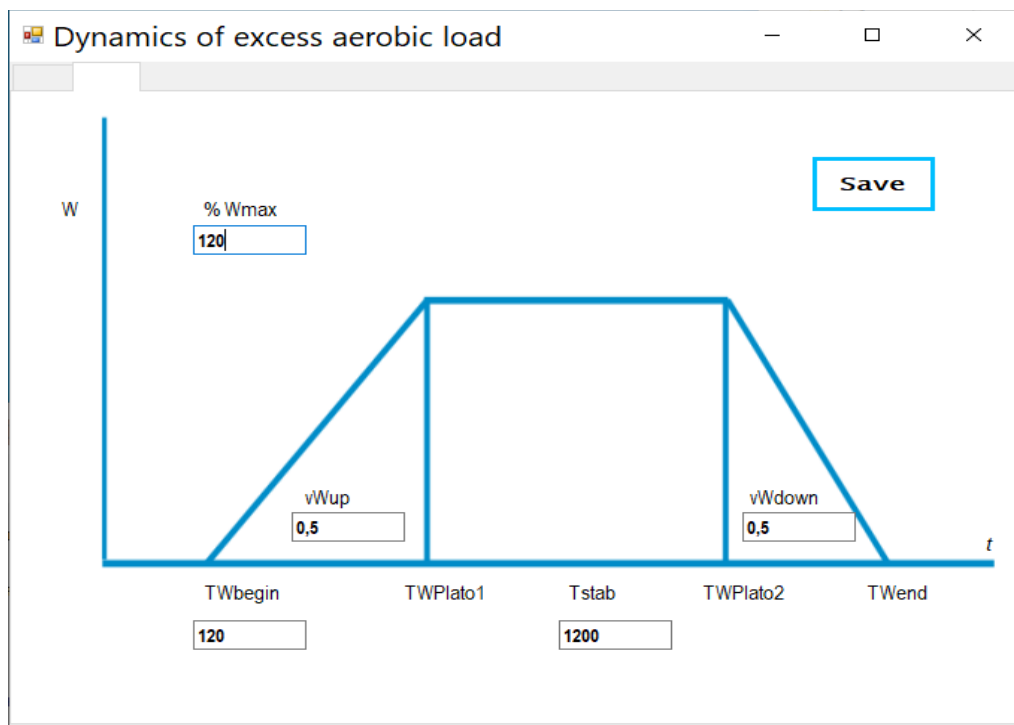


Fig.6. User interface window for setting current values of parameters of input load $W(t)$. The latter is given in a percentage of aerobic W_{max} . Time in sec.

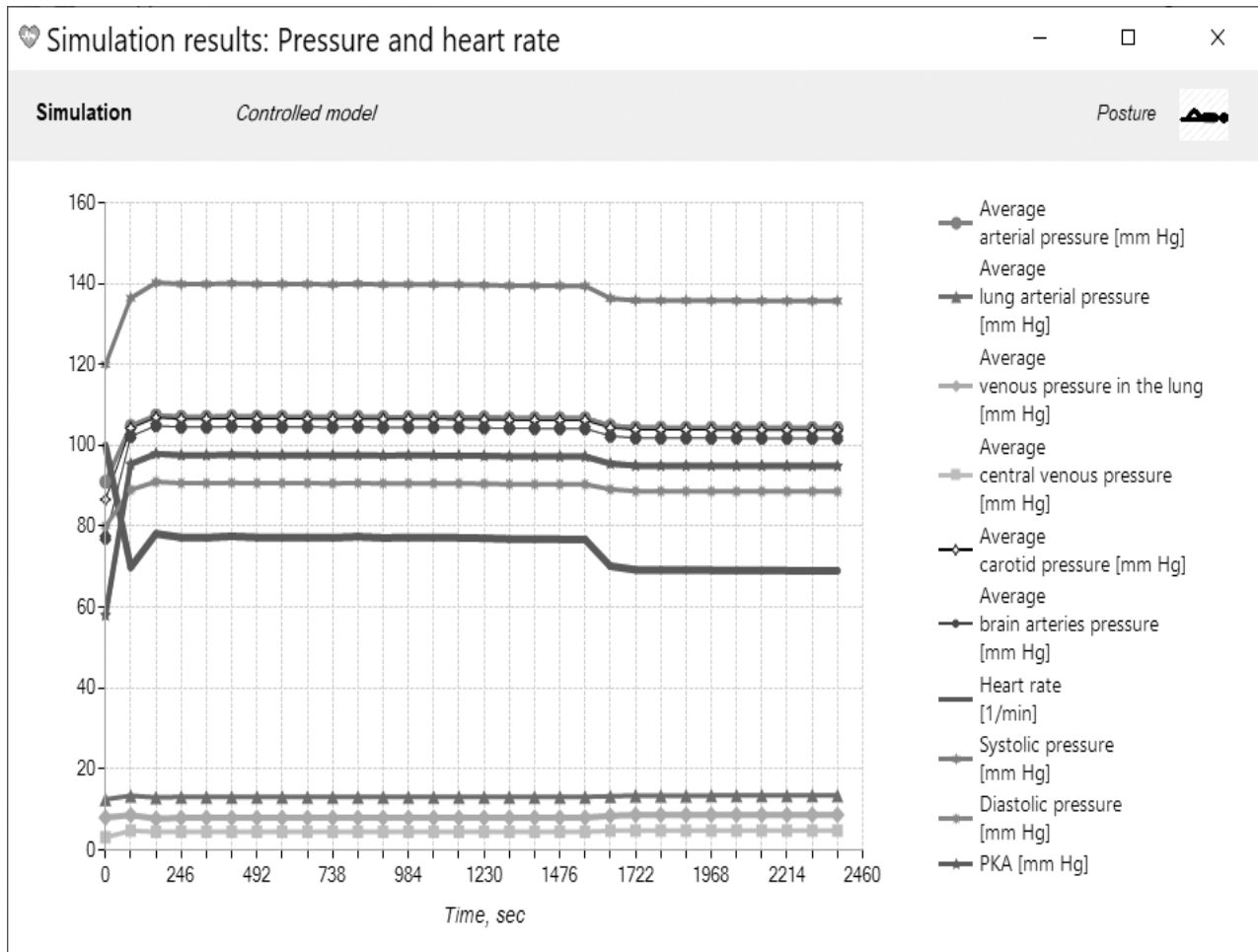


Fig.7. Main variables of central hemodynamics under aerobic load test.

The loading profile can be saved for repetitive simulations with new actual models.

In this paper, we do not present graph results of each of the seven simulation collections. Their interface design looks similar to next pictures illustrating graphs constructed for the case presented in Fig.6 under regulators' standard configuration and parameters of baroreceptor reflexes illustrated in Fig. 4.

It should be noted that in our current simulator, a simulation starts from data that have not been exactly tuned for the balanced hemodynamics. The exact tuning will require additional special algorithms associated with human posture. A random data set includes a certain phase of the initial transitory process that leads to balanced hemodynamics. Simulator's previous exploration has shown that the initial transitory process is almost completed at the 20-th second. So, it is much easier to use a common calculation algorithm taking into account that the simulation results are informative just for $t > 20$ sec. Namely, they present dynamic effects of regulators activation against

hemodynamic imbalances initiated by the dynamics of $W(t)$. Fig.8 below shows organism's responses to this specific load.

As curves in Fig.8 show, under 100% elevation of $W(t)$ with an increasing speeds of $0,5 W_{max}/sec$ and decreasing speeds of $0,5 W_{max}/sec$, chemical indicators of arterial blood have significant shifts. Shapes of these shifts are similar to the shape of $W(t)$ (see also Fig.6). Under the used elevation of $W(t)$, decreases of pO_2 , increases of pCO_2 are large enough to activate compensatory mechanisms of peripheral chemoreceptor reflexes. Proper increasing of lung ventilation and of the number of red blood cells are only particular compensatory effects of this reflex. Additional hemodynamic effects of chemoreceptor reflexes are shown in Fig. 7. Just $W(t)$ returns to its zero value, all compensatory shifts are slowly returning to their initial values.

The list of human physiological characteristics used in SS is larger than those shown in Fig.7 and Fig.8. These figures only illustrate our approach to visualizing dynamic

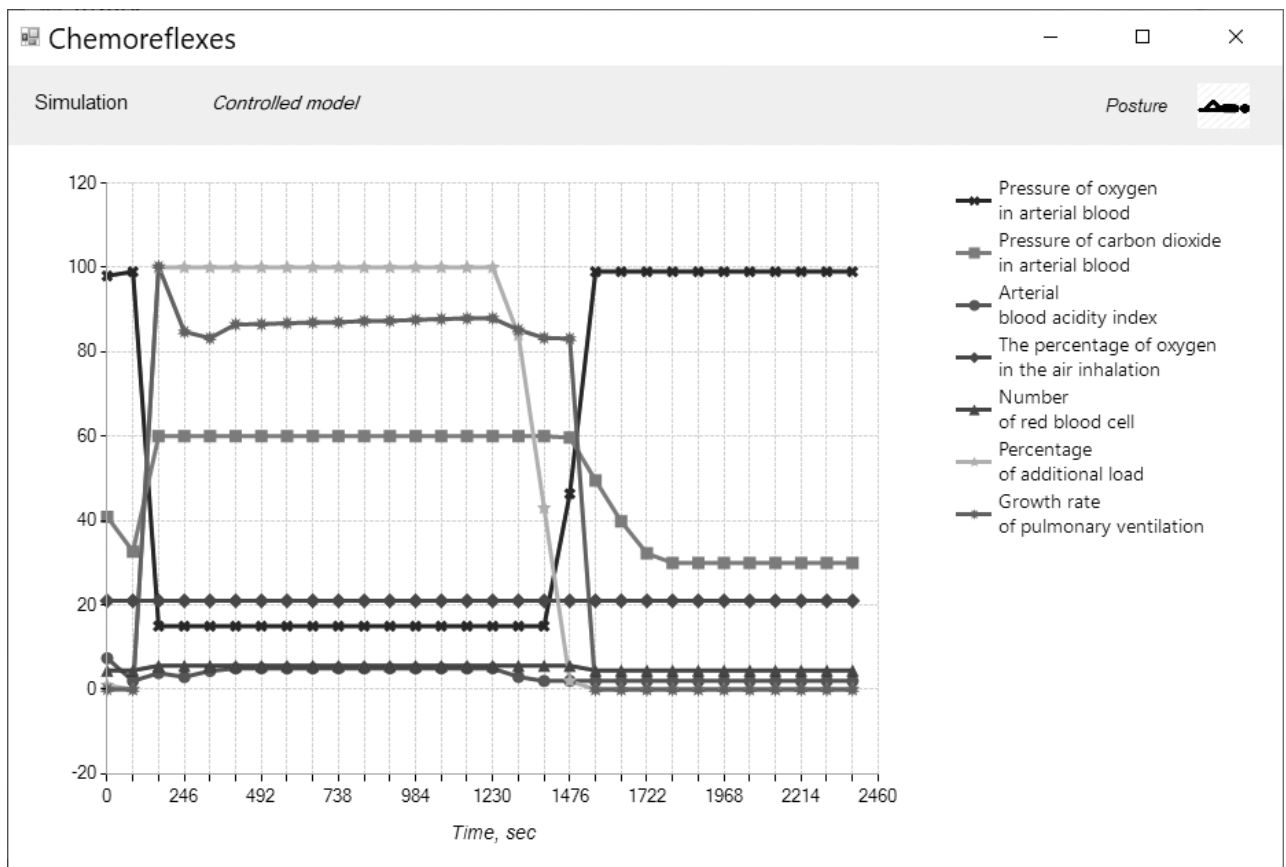


Fig.8. Illustrations of simulations for a group of variables characterizing chemoreceptor reflex. Here the aerobic load is twice as high as it is in human rest condition in clinostatics.

physiological characteristics potentially necessary to be included by physiologists in their systematic analysis of complex relationships determining cardiovascular and associated systems' integrative responses to internal / external physical-chemical loads.

Discussion

The simulator described in the paper is autonomous software designed for IBM compatible computers. At the same time, it is the first result of the long-term fundamental investigations aimed to understand human integrative physiology by means of novel conceptual and methodological renovations. In particular, renovations concern traditional empiric research that has to be expanded with theoretical computer-based research provided by adequate quantitative simulators of physiological mechanisms and events [6-11].

As it was mentioned above, the complex model includes 11 physiological regulators each modifying certain parameters of the core model. The last one describes self-regulatory properties of a closed loop (see Fig.1). So, the complex model includes 12 regulator-associated partial models. Even in the case when the test and its parameters are chosen, the user can provide experiments for $2^{12}=5096$ combinations of activated or deactivated regulators. This number is too huge: in fact no experimenter has ever explored the full range of regulators combinations. Adding to this number also the versions that appear due to combining certain tests and modifying their parameters, one can see that the number of potential simulation scenarios becomes so big that can hardly be tested by creators of SS. At the same time, it is worth to note that the main constants of our models were tuned using specially developed assistant software [11]. These reasons give us a basis to consider our SS the first working version capable of essentially expanding the physiological research concerning mechanisms altering human hemodynamics. Further improvement of this SS needs proper feedbacks from physiologists exploring our SS. We are sure that the exploration will accumulate additional data requiring our more thorough analysis.

Conclusion

For the first time special software (SS) capable of simulating alterations of human hemodynamics via automatic or arbitrary activations of main endogenous physiological mechanisms, is developed. SS is based on quantitative mathematical models representing CVS as an open system interacting with multiple associated organs and systems. Models have been tested and validated on the knowledge basis concerning physiological norm. Additionally, main hypotheses of arterial hypertension etiology can be modeled. SS provides physiologists with a novel research technology essentially widening and deepening the fundamental knowledge concerning human circulation. SS is also a good modern PC-based tool for simultaneously visualization of CVS's dynamic characteristics under the chosen list of input violations. The latter aspect will promote medical students to better understand non-obvious integrative human physiology and special pathologies. SS is also a good computer program to be used in educational purposes for illustrating main physiological and certain pathological regularities to medical students. We plan to expand the models and the software in order to simulate much more realistic scenarios of both normal and pathological human physiology.

References

1. 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.
2. Fresiello L., Ferrari G., Di Molfetta A., Zieliński K., Tzallas A., Jacobs S., et al. A cardiovascular simulator tailored for training and clinical uses. *J Biomed Inform* 2015,57. P. 100–112.
3. Grygoryan R.D. Problem-oriented computer simulators for solving of theoretical and applied tasks of human physiology. *Problems of programming*. 2017, №3, P. 102-111.
4. Grygoryan R.D., Lissov P.N. A software-simulator of human cardiovascular system based on its mathematical model. *Problems of program-*

- ming. 2004, №4. C.100-111 (Rus).
5. 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. (Rus.) doi.org/10.15407/pp2019.01.090.
 6. Grygoryan R.D., Degoda A.G., Dzhurinsky Y.A. A simulator of mechanisms of long-term control of human hemodynamics. Problems of programming, 2019;4:111-120.(Rus). doi.org/10.15407/pp2019.04.111.
 7. Grygoryan R.D., Aksenova T.V., Degoda A.G. A computer simulator of mechanisms providing energy balance in human cells. Cybernetics and computing technologies. 2017, №2 (188), P.65-73. (Rus).
 8. 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).
 9. Grygoryan R.D. The optimal circulation: cells’ contribution to arterial pressure. 2017, Nova Science, N.Y., 298 p.
 10. Grygoryan R.D. The unknown aspects of arterial pressure. Znanstvena misel journal, 2019, 33:19-23.
 11. Grygoryan R.D., Yurchak O.I., Degoda A.G., Lyudovyk T.V. A software technology providing tuning procedures of a quantitative model of human hemodynamics. Problems of programming, 2020;4:03-13. (Ukr). <https://doi.org/10.15407/pp2020.04.003>.

Received: 06.05.2021

About authors:

Grygoryan Rafik

Department chief, PhD, D-r in biology
 Publications number in Ukraine journals -147
 Publications number in English journals -46.
 Hirsch index – 10
<http://orcid.org/0000-0001-8762-733X>.

Yurchak Oksana,

Leading software engineer.
 Publications number in Ukraine journals – 14.
 Publications number in English journals - 0.
 Hirsch index –0.
<https://orcid.org/0000-0003-3941-1555>.

Degoda Anna,

Senior scientist, PhD.
 Publications number in Ukraine journals – 15.
 Publications number in English journals -1.
 Hirsch index – 3.
<http://orcid.org/0000-0001-6364-5568>.

Lyudovyk Tetiana,

Senior scientist, PhD.
 Publications number in Ukraine journals – 30.
 Publications number in English journals -17.
 Hirsch index – 5.
<https://orcid.org/0000-0003-0209-2001>.

Affiliation:

Institute of software systems of Ukraine National
 Academy of Sciences
 03187, Kyïv,
 Acad. Glushkov avenue, 40,
 Phone.: 526 5169.
 E-mail:
rgrygoryan@gmail.com, daravatan@gmail.com,
anna@silverlinecrm.com, tetyana.lyudovyk@gmail.com

Dmytro V. Rahozin, Anatoliy Yu. Doroshenko

EXTENDED PERFORMANCE ACCOUNTING USING VALGRIND TOOL

Modern workloads, parallel or sequential, usually suffer from insufficient memory and computing performance. Common trends to improve workload performance include the utilizations of complex functional units or coprocessors, which are able not only to provide accelerated computations but also independently fetch data from memory generating complex address patterns, with or without support of control flow operations. Such coprocessors usually are not adopted by optimizing compilers and should be utilized by special application interfaces by hand. On the other hand, memory bottlenecks may be avoided with proper use of processor prefetch capabilities which load necessary data ahead of actual utilization time, and the prefetch is also adopted only for simple cases making programmers to do it usually by hand. As workloads are fast migrating to embedded applications a problem raises how to utilize all hardware capabilities for speeding up workload at moderate efforts. This requires precise analysis of memory access patterns at program run time and marking hot spots where the vast amount of memory accesses is issued. Precise memory access model can be analyzed via simulators, for example Valgrind, which is capable to run really big workload, for example neural network inference in reasonable time. But simulators and hardware performance analyzers fail to separate the full amount of memory references and cache misses per particular modules as it requires the analysis of program call graph. We are extending Valgrind tool cache simulator, which allows to account memory accesses per software modules and render realistic distribution of hot spot in a program. Additionally the analysis of address sequences in the simulator allows to recover array access patterns and propose effective prefetching schemes. Motivating samples are provided to illustrate the use of Valgrind tool.

Keywords: workload, performance analysis, coprocessors, prefetch, computer system simulator.

Introduction

The ultimate goal of computer hardware development activities is the improvement of application performance some way. During early days of microprocessor development, the basic hardware features were adopted – such as hardware pipeline, on-board cache memory and SIMD instructions, so one microprocessor instruction transforms operands during minimal number of clock cycles. After that the more advanced hardware methods were adopted – for example, the extraction of instruction-level parallelism and introduction of complex instructions, for example RSA crypto support. Excessive amount of research efforts was spent to gather the most often used computation patterns, and so extra hundreds of complex instructions were added. These instructions apply complex computation pipelines over small amount of data, and usually are used only with help of sophisticated optimizing compiler or direct assembly language instructions. The next level of performance improvement is the use of specialized coprocessors, which not only apply the com-

plex computation pipeline, but also provide sophisticated address generation so that the coprocessor is able to access big memory areas. The cornerstone problem of this so called “next level” is the coprocessor complexity and the absence of good optimizing compilers which can transform original program code and map it onto the hardware coprocessor. So, mapping of the original algorithm to specialized hardware coprocessor is usually done by hand and rises software development costs. There are many examples of different kinds of mappings: starting at simpler RSA crypto-algorithm accelerators in various platforms – x86/ARM/PowerPC, where the coder just takes a code example from an application note; up to programming graphics card using shader concept, using a complex compiler to generate parallel code for GPU.

This so called “next level” of specialized hardware applications requires not only analysis of computational patterns over some scalar data, but also requires the analysis of data flow and transformations in nested

loops. The goal of the analysis is not only code optimization, but gathering requirements for useful coprocessor employment to accelerate complex computing patterns. Let's define the *complex computing pattern* as a part of computing algorithm which includes at least one single or nested loop and needs a complex memory access pattern (much more complex than just SIMD data path).

The goal of this paper is to make a step forward in the topic of software performance analysis and optimization for solving the following cases: 1) semi-automatic extraction of the information for possible optimization of the complex computing patterns with the help of co-processors with programmable memory access; 2) analysis of mapping of complex computing patterns for co-processors with programmable memory access. We consider off-the-shelf software and possible optimization cases for it and we propose techniques for this software optimization using performance analysis tools. We consider the extensions for Valgrind software (especially its subtool Cachegrind) which enable additional analysis of complex computing patterns.

1. Application performance analysis

So, complex computational patterns we are looking for usually reflect commonly used computation procedures, for example convolutions, matrix multiplication loops, loops similar to high-level BLAS kernels, neural network computational kernels, various DSP kernels. For many cases we already have appropriate tools, there iterative optimization techniques are employed for the existing code [1] and this greatly improves developer experience and reduces efforts necessary to optimize software. Another way which allows to simplify development and decrease efforts is the use of formal methods, were the development system already operates with parallel algorithms [2]. Although many complex systems can be modeled using high-level formal models, usually the model formalization is the second or third step of technology adaptation. Let us consider the modern topic of convolutional neural networks with popular implementations from Nvidia [3] and Berkeley [4]. Both implemen-

tations, despite a neural network can be well described as the formal models, are brilliant high-level frameworks for neural networks implementation, but still are hardly portable to any architecture except initial targets - Nvidia video cards and x86-compatible multicore processors. If a different hardware is required to run the neural network back-end and this hardware includes special coprocessor, we need to analyze the initial programs for "hot spots" – kernels or loop nest where the program spent the most of time. As software became more and more complex we need to have appropriate tools to analyze the code and there is no a trend to use highly formal techniques to define neural networks due to high complexity of infrastructure for formal models for parallel applications. Instead, we see the use of simpler tools such as Darknet [5] for implementing complex pipelines such as Yolo-v4 [6]. The analysis of such complex applications [7] made us to start looking for effective tools for analyzing the programs which can be optimized on modern multiprocessors minimizing human effort need to be spent on this analysis.

Today in practice neural network algorithms can be efficiently optimized with the use of a specialized coprocessor (such as Qualcomm's Neural Processing Engine [8]) and this is the common trend in system-on-chip design. The number of workloads, which performance can be improved with various coprocessors, quickly increases, so the interest of employing a coprocessor in application constantly increases. The impressive application – AI Benchmark (ai-benchmark.com), which uses QNPE SDK [8] for neural network processing – enables optimized neural network processing for system-on-chips widely used in off-the-shelf smartphones.

But this is the only side of hardware development. The fact is that the performance of some applications is bounded by execution speed (e.g. cryptographic hashes computations), another by memory performance (matrix multiplications, neural network convolutions), another by both memory performance and execution speed. Modern software is extremely complex and can be analyzed and optimized mostly in parts, but the common rule is that 90% of execu-

tion time is spent in 10% of code, for some workloads this ratio is 99%/1%. Even basic analysis shows that the most of modern applications are memory bounded, but the more detailed analysis is able to detect the most time consuming “hot spots” in application, as is successfully done using e.g. Intel Vtune performance analyser [9], or another similar software. Anyway, Intel Vtune and another analysis software are based on statistical approach and shows only “hot spot” with quite accurate number of memory traffic per executed instructions. But the high-level information about address sequences for memory-related instructions is not gathered during this analysis, but this missed information is the key for understanding the algorithm behavior and possible algorithm mapping on the coprocessor.

As current optimizing compilers are not able to map nested loops and complex addressing patterns on complex hardware, the co-processor should be utilized by hand using the SDK such as described in [8]. In order to simplify the handmade optimizations, performance analysis software should analyze addressing patterns, addressing patterns spatial locality and issued operations. This type of reporting is used in two ways: 1) reporting potential computation patterns, which can be optimized by hands; 2) reporting “hot spots” which are potentially optimizable if a corresponding coprocessor is included into system-on-chip. Another valuable point is the definition of prefetch scheme, which may be used for cache utilization optimization. Data access analysis is able to recover at least simple addressing patterns, which may be used for hand-made prefetch optimization.

An important trend is that commonly used hardware employs more and more microprocessor kernels (usually ARM in latest system-on-chips as ARM hardware kernel are small and energy efficient), and these kernels share or sit on common cache memory. Each kernel has enough computation power – with up to 3.5GHz clock frequency, with limited instruction level parallelism extraction (as in Cortex-A57/A75/A77) – and is able to process complex modern algorithms even without coprocessors. Note, that copro-

cessors also use the common cache memory for operations, so executing a computational thread on coprocessor hardware just heavily increases load on cache memory bus. This can be illustrated by the old fact related to employment of hyperthreading technology, when a processor core is able to execute instructions of two threads simultaneously. Running parallel threads which does not operate on same data effectively halves the cache size, which reduces performance despite employing two threads. The same works e.g. for typical conjugate gradient solver – it scales well only for several threads. Cache behavior analysis helps to determine cache bottlenecks and check if the bottleneck can be avoided, and additionally cache prefetch scheme can be defined for a pattern.

2. Efficient cache memory use and prefetch techniques

Memory bounded applications (such as mentioned above conjugate gradient solver) performance may be improved by proper prefetching scheme. The cache memory never works foreheads, so the cost of L1 cache misses remains extremely high. For high processor frequencies (~3 GHz) one L2 hit (i.e. L1 miss) costs up to dozen of clock cycles, one L3 hit (i.e. L2 miss) costs up to 70 cycles for cache interconnection type typical for Intel multicore processors. Read from DRAM costs up to 300 cycles. If compared to usual floating-point operations time, which equals to 1-2 clock cycles, memory access in case of L1 cache miss looks extremely high.

Farther in the paper while considering processor operations speed, we omit time spent for computational operations. It was mid-1980s when the off-the-shelf processor executed numerical operations taking multiple clock cycles. Starting at 32-bit processors in 1990s we never expect that basic floating-point operations (addition, multiplication) takes more than 1 clock cycle. On the other hand, if in early-1980s DRAM memory access appeared without additional wait cycles even at ancient ISA bus, each time after processor clock speed was significantly increased, the memory access time raised and the complexity of memory hierarchy in-

creased up to be enormously complex. Sure, that the main focus was moved from “how to compute fast” to “how to feed a processor with data fast”.

A thoughtful reader may note that some memory-hungry operations may be introduced into memory controller, especially so called “reduction” operations, when some scalar operation is executed over large data array, for example convolution operation. Talking more precisely, a coprocessor which is able to speed up various BLAS kernels may improve significantly the performance of operations in many current workloads, but BLAS kernels work fast only if cache memory is used efficiently, that’s why modern BLAS libraries always make a tuning for BLAS library before compiling it for particular machine for processor type and its cache memory configuration. This also applies for graphics processors, as BLAS package is compiled separately for each GPU architecture and modification type to use GPU registers in efficient way. So prefetch here is able to set the cache memory into desired configuration to avoid L1 cache misses. Modern prefetch instructions are able to move data between cache layers in order to hide even L2/L3 latencies. Anyway, this does not look a silver bullet – multithreaded applications perform side-effects on cache contents and Cachegrind [10] is a good tool to check efficiency of cache utilization in application regardless if application works on CPU or on GPU, as the memory traffic requirements are practically the same.

Several dozens of prefetch application schemes are considered in [11], but the fully automatic prefetch is extremely limited by adopting only simpler addressing schemes and compiler ability to determine loop constructions. Hardware prefetch schemes are also considered in [11], still they work for simpler addressing schemes.

Current prefetch hardware includes not only prefetch instructions, but also cache lines locking instructions, data invalidating instructions, changing priority of data invalidation/eviction in set and other service commands. The most complex case includes one or more separate prefetch coprocessors, which are able to prefetch arrays of large data

blocks with stride synchronously with main thread(s) or prefetch linked lists of large data blocks. Here the prefetching coprocessor needs to be programmed by a full-blown control code and looks to be quite complex hardware.

Although current optimizing compilers use powerful algorithms, complex prefetch schemes are too hard for them – if a simple loop can be analyzed usually successfully, the loop nest is harder to analyze. The analysis of memory accesses sequence (address patterns) is the valuable way to check possible “hot spots” potentially optimizable by prefetch patterns.

3. Valgrind performance analysis tool

Valgrind tool [10] basically emulates a microprocessor instruction set, memory state of Intel x86, ARM and several other processors. Valgrind is able to emulate basic operation system libraries and run a program on the top of emulated system. The program is not modified any way (except special cases extending the program to control Valgrind behavior in run-time). Several useful tools are based on the simulator: a usual debugger *gdb*, a memory error detector (it checks for incorrect heap memory use), a cache memory and branch predictor simulator (a.k.a. Cachegrind), a call-graph generator, a concurrent thread error checker and even more profiling tools. Our target sub-tool is Cachegrind, the cache memory simulator and profiler. The sub-tool simulates two-level cache memory with a bunch of programmable parameters, and the most important parameter is the size of cache memory levels storage. Basically, the cache simulator provides tracking of: 1) traffic from register file to cache memory, 1st level, this is the application data throughput; 2) data traffic to 2nd level cache as the result of cache misses for 1st level cache memory for both reads and writes; 3) cache misses for 2nd cache level which is the final traffic to memory. The ratio (1) to (3) is the cache memory utilization efficiency, which depends on application type, compiler optimizations and applied threading model. With help of changeable cache

parameters Cachegrind analyzes cache performance deeply even for multithreaded programs. The simplest analysis case is graph of memory traffic (3) vs cache size, which allows to define the most comfortable cache size for application.

Before considering why we are limited in Cachegrind functionality we need to introduce types of workloads we are going to analyse with Cachegrind tool.

Microprocessor performance is still a keystone for enabling computing technologies for mass market, and more and more computing performance nowadays is reached by introducing more and more parallelism to newer hardware. Well-known Moore law says that the number of gates (transistors) on (silicon) die doubles every 18 months, but these extra transistors do not help to compute faster, so they need to be used in computational units which are able to perform in parallel and the computing task should have enough level of coarse-grained parallelism. The ultimate case is a video card equipped with a massive parallel processor, which initially was used to compute pixel color inside a rendered triangle, which is a highly parallel task performed on independent data. But the performance bottleneck for any video card is the memory channel, so the huge number of architectural solutions in video cards solves the problem of hiding memory latency. For less specialized computing software the matrix processing operations still are championing the race for computing resources. Modern neural network processing is based on large matrix multiplications, and despite the matrix multiplication optimizations are well studied, they are combined with another memory operations. This combination is not studied well due to its novelty, so it is our main case for performance

analysis as 1) neural networks are emerging topic for mass market and challenging topic for computer hardware; 2) huge industrial demand for object and environment recognition, which is potentially solved by neural networks technology.

Recently we conducted a research [7] where we simulated neural network runs in order to project this workload performance for an embedded platform. We proved that Valgrind/Cachegrind is able to simulate huge modern neural networks runs with getting full performance results for cache memory from Cachegrind. Let us consider a result got from Cachegrind run simulating Yolo-v4 run on Intel x86 platform, details can be found in [7]. Note, that original program – Darknet infrastructure [5] was not changed anyway.

Fig 1. shows the “hot spot” in Darknet Yolo-v4 run, which is *gemm_nn* function, providing general matrix multiplication. The function generates 83.5% misses in 1st level cache memory and 98.5% misses in 2nd level cache memory and due to analysis, it is the main source of memory traffic in the application. This information looks very short in terms of the performance analysis of parallel or coprocessor-supported execution of the sample. Looking at fig. 1 we consider the matrix multiplication information, which can benefit from blocked matrix multiplication, but no one can easily determine if some particular threading affinity configuration makes the program faster due to sharing cache data among threads or makes the program slower due to extra data spills from cache. Cachegrind simulation potentially can provide such type of analysis, but current output at fig. 1 limits a Cachegrind user in ability to analyze the software run – practically all the memory resource is spent in one *gemm_nn* function.

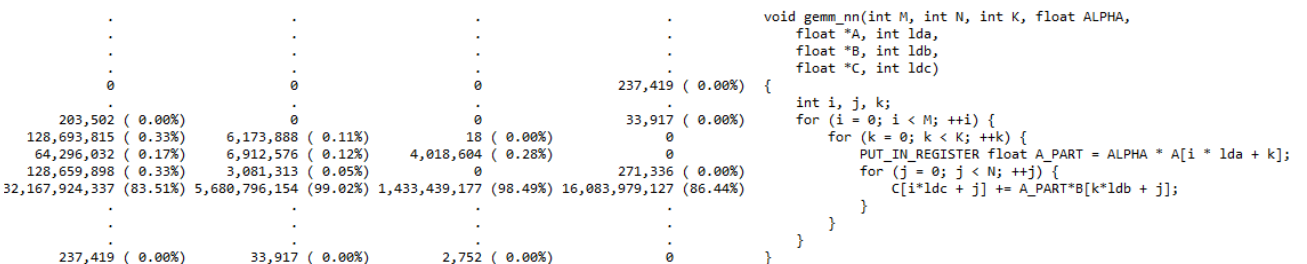


Fig. 1. Excerpt from Yolo-v4 run performance result: D1mr, D1mw, D2mr, D2mw values

4. Performance analysis shortcomings

Let us consider fig. 2. It includes three functional elements (FEs), #1, #2, #3, which represent three separate neural network layers, each calls several kernels from library.

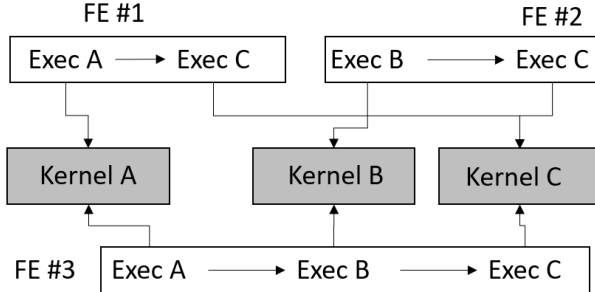


Fig. 2. A sample call graph for an excerpt from neural network.

So, FE #1 call kernels *A* and *C*, FE #2 calls *B* and *C*, and FE #3 calls all *A*, *B* and *C*. So, kernel *A* collects cache events from FE #1, #3; *B* from #2, #3 and *C* from all FEs. Therefore the performance gathering via Cachegrind does not reflect real distribution of spent resources per original FEs. Fig. 3 shows the sequence of basic neural network layers, forming Yolo-v4 pipeline. In practice, Valgrind mixes performance information from all the layers at fig. 3 into one function *gemm_nn*, generating the report at fig. 1, where *gemm_nn* gathers practically all memory operations into one function. This prevents proper analysis, as there is no information for each separate network layer, which calls *gemm_nn* function. Similar considerations work for sharing cache contents between several execution threads, here we have the case that the cache memory is simulated correctly, but it is unknown how each execution thread influences the cache content and how to compare one-threaded and multithreaded program execution in terms of cache behavior and cal-

culate the [non]efficiency of cache use if threading model and number of threads are changed in run-time.

The only way to overcome this limitation is to instrument (add the functionality for controlling performance analysis) the code, so that the performance accounting for all FEs is separated. “From-the-box” Cachegrind is not controllable somehow, but Valgrind has extensible API to control the behavior of other Valgrind tools.

5. Extending Valgrind tool

Valgrind includes a mechanism which allows user to control Valgrind-based execution of a program. The user is able to place “specific client requests” into program to control several Valgrind components, for example for Callgrind and add new client requests.

To control the tool user should use *callgrind.h* file from Valgrind distribution and use predefined macros, for example *CALLGRIND_START_INSTRUMENTATION*. This “C-style” macro definition and any other client request macros are directly translated into a specific processor instruction for target platform (x86, PowerPC), which is “void” i.e. does not change micro-processor state, but allows to pass arguments to Valgrind kernel. Client request parameters are passed into Cachegrind in similar way to standard function argument list. In run-time Valgrind core intercepts the compiled binary instruction and passes control to appropriate Cachegrind handler, which provides necessary functionality to analyze or change Cachegrind internal state.

Cachegrind handles cache memory state in separate data structures, including cache memory contents, memory tags state, eviction candidates information and statistics for cache misses/loads/traffic to

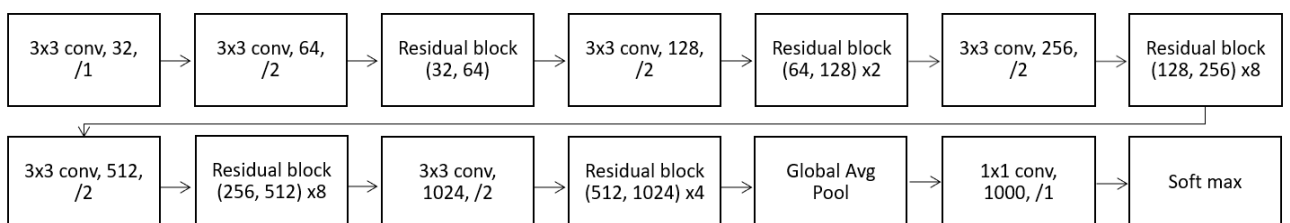


Fig 3. Computational structure of Yolo-v4 blocks.

memory. In order to improve cache simulation, we provide several copies of cache statistics – we call it “context” - and each context copy is filled with statistics separately by Cachegrind simulator. The number of contexts is set by user during Valgrind compilation. Basically, we add two new client requests – *CG_PUSH_CONTEXT* and *CG_POP_CONTEXT*. “CG” stands for “Cachegrind”. A new client request *CG_PUSH_CONTEXT* switches current context to another enumerated one so that the further statistics about cache misses and data traffic is added to another context. Default context number is 0, it is used from the start of cache simulation. After the context switch the previous context number is saved in the stack of context numbers, so that the following request *CG_POP_CONTEXT* is able to restore the previous context. Appropriate client requests for changing cache context are inserted into the source code so that the selected hot spots and kernels are separated in different cache contexts. Another improvement is adding an “old context” bit into cache content descriptors per each cache memory line. This bit is set in the case of cache context switch for all data stored in cache and is cleared in the case if a cache “hit” into the cache data was detected or in the case of data eviction from cache. The accounting for the number of bit clears of the “old context” bit in the case of a cache hit gives us the amount of data which was reused across different hot spots/kernels in the program code.

Some useful inter-thread data usage statistics may use the similar principle – track the thread identifiers (ids) and variable operations (read/write) to analyze variable sharing efficiency. Valgrind already have the tool analyzing the multithreaded program for dangerous data races, but we leave this research for near future.

We decided to research in steering for generated addresses – handling a pool of last used addresses, keeping a record of possible increments for each address and establishing a sequence of accessed memory cell for addressing schemes in a loop allows a user to analyze an address patterns

and form a report notifying about structural accesses i.e. data accesses with some specific address change pattern over one variable, structure, array, array of structures. The similar functionality is embedded into x86 processors: hardware-based automatic data prefetch. For example, the loop:

```
For(int i=0; i<N; i++)
{ c[i] = A*a[i] + B*b[i] + P; }
```

has the following structural accesses (we use the pattern *variable_name[start_index:step:end_index]*: *read a[0:1:N]*, *read b[0:1:N]*, *set c[0:1:N]*. This analysis helps in cases of mapping cache accesses and real arrays and allows to determine how much time the array was accessed and reused in cache. The address steering information is dumped at the end of simulation and does not require additional user control.

6. Use of extended Valgrind functionality

Let us return back to figure 2 and consider block FE #1, #2, #3, where the kernels *A*, *B* and *C* are called in some sequence. Default cache simulation shows, that the number of cache misses is distributed as follows:

Kernels	A	B	C
	22%	45%	25%

The same distribution for FEs is:

FE	#1	#2	#3
	0.7%	1%	0.5%

These tables show just that all memory traffic is utilized in library kernels *A*, *B* and *C* but says nothing about actual memory traffic distribution across FEs, which is necessary to get real distribution of memory accesses per FE. Note that the use of hardware counter in e.g. Intel Vtune shows the same picture for this case. To improve the analysis let us use cache context 1, 2, 3 for FEs #1, #2, #3 and add corresponding Cachegrind controls *CG_PUSH/POP CONTEXT* into the source code of FEs. Passing the updated program into the simulator shows the next picture:

Kernel	A	B	C
FE #1	3%	6%	3%
FE #2	11%	21%	14%
FE #3	8%	18%	8%

Now we clearly see how much memory traffic is utilized in each FE. The difference

between the default performance data and improved data does not require any comments.

Conclusions

The article considers the use and extension of a microprocessor and system simulator Valgrind for performance accounting and performance analysis of big modern workloads. As a conclusion, we accent several points which are helpful for studying of efficiency of modern workloads:

1. Even big modern workloads such as object detection neural networks are able to be analyzed by system simulator using off-the-shelf computers in short time. Also, memory behavior simulation is the main hardware subsystem we need to analyze in order to understand the workload performance bottlenecks.

2. Valgrind tool may be easily extended for research purpose to control or change the simulation process behavior via client requests.

3. Our extensions for Cachegrind control allows to analyze big pipelines in parts and determine bottlenecks in memory subsystem (cache memory and memory bus).

4. We have checked methods for analyzing data address streams for recovering prefetch pattern for nested loops and found that basic memory addressing schemes are recovered good enough to provide data for necessary cache traffic. This works extremely good jointly with (3).

5. Data stream recovery allows to separate automatically cache misses while loading several data arrays (streams) and compare cache performance for each array (and its prefetch method) for various data layouts and prefetch methods.

6. Cachegrind allows to simulate various cache behavior, so we can change e.g. cache data eviction policies in order to check if this can improve performance while executing some parts of workloads.

This kind of performance statistics gathering and grouping allows the qualified software engineer to find potentially optimizable part of code much faster and enable various preprocessors for a workload or apply profitable prefetch schemes. Simulator analysis here works more efficiently than di-

rect workload runs as the simulations allows to gather and keep information which is lost while fast direct software runs.

These results give us some prospects for future work. Valgrind supports multithreaded execution and memory races analysis for several threads. One of interesting ways to use Valgrind is to run different combinations and affinity of software threads in a multithreaded workload, checking performance effects on shared cache data, and this topic is the research target for near future.

References

1. A. Doroshenko, O. Beketov. Large-Scale Loops Parallelization for GPU Accelerators. //In Proc. of the 15th Int. Conf. on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Vol I. Kherson, Ukraine, June 12-15, 2019. CEUR-WS, vol. 2387 (2019).-P.82-89. <http://ceur-ws.org/Vol-2387/>
2. A. Doroshenko, O. Yatsenko. Formal and Adaptive Methods for Automation of Parallel Programs Construction: Emerging Research and Opportunities. IGI Global, Hershey, Pennsylvania, USA. 2021, 279 p. DOI: 10.4018/978-1-5225-9384-3
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./pp. 265--283)
4. Y. Jia and Evan Shelhamer and J. Donahue and S. Karayev and J. Long and Ross B. Girshick et al. Caffe: Convolutional Architecture for Fast Feature Embedding. // In Proc. of the 22nd ACM international conference on Multimedia, 2014
5. J.Redmon. (2013) [Online]. Darknet: Open Source Neural Networks in C. – Available from <https://pjreddie.com/darknet/>
6. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv: 2004.10934.
7. D. Ragozin, A. Doroshenko. Memory Subsystems Performance Analysis for CNN Workloads. // In Proc. of AUTOMATION 2020: 26-th Scientific conf. in memory of

- L. Pontryagin, N. Krasovsky and B. Pshenichny, 2020, Kyiv, Ukraine. P. 12-122.
8. Ignatov A. et al. (2019) AI Benchmark: Running Deep Neural Networks on Android Smartphones. In: Leal-Taixé L., Roth S. (eds) Computer Vision – ECCV 2018 Workshops. ECCV 2018. Lecture Notes in Computer Science, vol 11133. Springer, Cham. https://doi.org/10.1007/978-3-030-11021-5_19
 9. J. Rainders and J. Jeffers. High Performance Parallelism Pearls. Morgan-Kaufmann, 2015. 502 p., <https://doi.org/10.1016/C2014-0-01797-2>
 10. J. Weidendorfer. Sequential Performance Analysis with Callgrind and KCachegrind. // In Proc. of the 2nd International Workshop on Parallel Tools for High Performance Computing, July 2008, HLRS, Stuttgart, pp. 93-113
 11. Mittal, Sparsh. (2016). A Survey of Recent Prefetching Techniques for Processor Caches. ACM Computing Surveys. 49. 10.1145/2907071.
 12. Kim, Yoongu; Yang, Weikun; Mutlu, Onur (2016): Ramulator: A Fast and Extensible DRAM Simulator. Carnegie Mellon University. Journal contribution. <https://doi.org/10.1184/R1/6469208.v1>

Received: 17.05.2021

About authors:

Dmytro V. Rahozin, candidate of tech. sciences (PhD)

More than 10 publication in Ukrainian and foreign journals.

<https://orcid.org/0000-0002-8445-9921>

Anatoliy Doroshenko, Doctor of Sciences in Physics and Mathematics, Professor, Head of the Department of Computing Theory, Institute of Software System of the National Academy of Sciences of Ukraine, Professor of Department of Automation and Control in Technical Systems

Igor Sikorsky Kyiv Polytechnic Institute.

Number of scientific publications in

Ukrainian publications - more than 180.

Number of scientific publications in foreign publications - more than 70.

Hirsh index - 6.

<http://orcid.org/0000-0002-8435-1451>

Affiliations:

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

M. Kosovets, L. Tovstenko

SPECIFIC FEATURES OF THE USE OF ARTIFICIAL INTELLIGENCE IN THE DEVELOPMENT OF THE ARCHITECTURE OF INTELLIGENT FAULT-TOLERANT RADAR SYSTEMS

The problem of developing the architecture of modern cognitive radar systems in the form of a set of heterogeneous neuromultimicroprocessor modules using artificial intelligence technologies, taking into account the requirements for the purpose, the influence of external and internal factors, is considered. The concept of a resource in general and an abstract reliability resource in particular and its role in the design of a neuromultimicroprocessor with fault tolerance properties are introduced. The change in the ratio of performance and reliability of a neural network is shown, which is rebuilt in the process of solving a problem in real time with a lack of reliability resources at the system level by means of the operating system which dynamically changing the architectural appearance of the system with structural redundancy, using fault-tolerant technologies and dependable computations.

Keywords: neuromultimicroprocessor, probability of trouble-free operation, initialization, resource, interface, modularity, supervisor, multiprogramming, reconfiguration system, access method.

Introduction

The growth of the use of radar technologies in various sectors of the economy: medicine, military equipment, security, agriculture, geology, IoT and others became possible due to the miniaturization of the element base, the development of artificial intelligence technologies, cloud computing [1,2]. Cognitiveness of modern radars plays a key role, and there is no alternative to this path [3]. The main requirements for radars are to minimize the negative impact on human health, the surrounding electronic devices, their invisibility, to obtain information about environmental pollution, 3D images of the location scene, information about the health of the environment, the presence of viruses and bacteria [4-9]. These requirements complicate both the transmitting part of the radar and the receiving part, which is associated with the problem of extracting signals from the noise. The development of technology explains the failed attempts to build ground penetrating radar (GRP), mine detection radars (Mine Radar), Portable Smart through Wall 3D Imager Radar (PSTW), marine radars with a low probability of interception of the near and far zone ("Low Probability of Intercept" (LPI)) and ways overcoming them. This is the intel-

lectualization of radars: the use of neural networks and their deep learning [10-20].

The deployment of radar systems has a multi-level system with continuous coverage and close connection between the levels. The primary link forms a network of radars, telecommunications environment, the second link - the deployment of military, and security radars [21-24], the third link - IoT radars, radars built into gadgets located in cars, pollution checkpoints, and virus detection and others. The construction of radars of different links has significant differences. The focus of the first link is on a reliable neural network and its connections to cloud computing and providing fast access. The second link is characterized by the use of a neural network to organize a neural computer and the use of neural network computing through deep learning. The third link is the most widely used, characterized by small size, often placed in gadgets, which teach artificial neural network, in contrast to the first two links. In first two links the neural network is designed and reproduced in hard ware by large teams of developers, creating and providing resources to the third link. Collection, processing, storage and reproduction of

information are possible from other sensors and information artificially entered for processing and storage.

Existing architectural models are not able to adequately display applied radar information. Unfortunately, artificial intelligence technologies do not yet have sufficient development, developed neural network components, experience of deep learning of the neural network, developments in cognitive algorithms, issues of achieving fault tolerance, providing dependability of computations in real time. Therefore, today we use a compromise option that combines signal processing and in-depth learning, neural networks with multiprocessing [25-28]. This will help to use all the developments so far, to make new developments that will be relevant in full intellectualization.

The use of neural network systems for processing radar information has many advantages. It is possible to gradually build up the neural network and train the modified network. Also, in the neural network, we lay the possibilities of increasing fault tolerance by reconfiguring the system in the process of solving the problem. Fault tolerance of a radar system with minimal time redundancy and deadlock termination is especially important in real-time systems [29-33]. An analysis of the problems associated with the processing of radar information shows that many tasks of processing radar information are solved using a neural network, the architecture of which does not correspond to the class of tasks. Thus, the tasks of processing multidimensional fields are solved using built-in micro-computers with a streaming accelerator, the architecture of which reflects the problems of organizing computations, ranked by dependability levels.

We pay special attention to the infrastructure of neural network design tools. Unfortunately, there are no neural network design tools on the market, except for debugging tools for individual components. The manufacture of a neural network on a crystal is spreading. Architectural models reflect the functionality of hardware and software components using high-level abstraction in the form of data streams and abstractly represent implementation technology over time. Architectural models contain arbitration schemes, can be parameterized and typed. Thus, the configuration

parameters of the architectural model make it possible to determine the relationship between the implementation of functions by hardware and software. System-level design using architectural modeling simplifies the design specification, makes a smooth transition from functional requirements to formal requirements, since it separates the problems of developing functional requirements and design specifications, since there are usually no means to quantify specifications [34-38].

The problem of reliability of real-time radar intelligent systems

The problem of reliability plays an important role in the design of systems in networks for collecting, processing and transmitting information. By reliability we mean the probability that the system will perform a given function in a given period of time under specified environmental conditions. The analysis of the systems has shown that reliability at a basic level is a fundamental parameter. Since the systems are distributed in space, failure leads to the collapse of the entire system. Solution paths are changing all the time, especially now when neural networks are simultaneously used to ensure reliability and to solve an applied problem. On the one hand, they simplify the solution of the problem of reliability, survivability of systems, and on the other hand, they complicate the hardware and software. The article discusses design methods for fault-tolerant neuromultimicro-processor real-time systems. These methods include software and circuit methods for detecting failures, which must ensure the regular operation of each processor module, data exchange between subsystems and the reliability of information before using it.

The problem of reliability is always considered at the design stage. Traditionally, failure prevention has been achieved in various ways, for example: by creating ultra-reliable multiprocessor components; improving maintenance through the development of effective troubleshooting methods; improving the procedure for controlling the technological process of manufacturing, testing and certification of finished products; implementation of hardware redundancy; the creation of technology for designing systems that have the properties

of resistance to failures in conditions when defects inevitably exist and manifest themselves in the form of failures and random failures. By fault tolerance, we mean such a property of the architecture of a digital system that allows a logical machine to continue working even when a variety of component failures occur in a real system that is its carrier [39]. The main task of the fault tolerance solution is to restore the computational process from the point of failure. To do this, it is necessary to detect and isolate the failure. To restore operability, knowledge of the state vector at the current time is required. Recovery techniques depend on the ability to isolate a detected failure in the system at the lowest possible level of system abstraction. The recovery mechanism proposed in this article not only ensures modularity and simplicity of the system, but also enables quick recovery and accurate prediction of the task completion time.

Initially, fault-tolerant technologies were developed for on-board electronic equipment, for which a number of parameters are critical: weight, dimensions, power consumption, system unification, time and money spent on designing a new system, the complexity of modernization procedures, and high reliability requirements. A typical system based on a real-time multiprocessor, insensitive to failures, should at least recognize 98% of all possible errors and identify at least 95%. To do this, we use built-in control systems, tests of acceptability and meaningfulness, and the implementation of reliability procedures.

After an error is detected, the faulty components are localized and excluded from the computational process. The system is reconfigured, the task is redistributed between free processors, which are initialized and included in the computational process from the point of failure. A restore point is defined by an application program that stores information up to the restore point. If an error is detected in the subsystem, recovery is possible through restart, which is impractical, since the computational process starts from the initial value, and if the computational processes are interconnected, then it becomes difficult to isolate the refusal from affecting other parallel processes. Therefore, when developing programs, parallel processes must be carefully structured so,

that restore points in interacting processes are mutually consistent.

A replay of one process can propagate to other processes and events, which is called replay propagation. Sometimes there is an avalanche of repetitions. In such cases, the process goes back a few steps. In this case, redundancy in the recovery process and loss of productivity are inevitable. We have considered the issues related to the resource management of a fault-tolerant real-time neuromultimicroprocessor, now we will try to cover in more details the issue of the impact of resource management on the fault-tolerant system.

In the design of radar systems, solutions are still being sought to improve reliability through redundancy but their architecture is outdated. When building special-purpose systems, especially airborne, radar, telecommunications, there is no alternative to fault-tolerant architectures. The challenge of building fault-tolerant systems lies in the complete revision of ingrained design principles and ideology. The value of reliability is calculated and laid down at the system level and depends not only on hardware and software resources, but to a greater extent on their interaction, resource management. As a result, reliability acts as an abstract resource of the system and varies depending on the task being performed.

Fault tolerance is provided by hardware, software or hardware-software redundancy. Failure of an individual processor module manifests itself in a limited loop. Fault tolerance during operation is determined by error detection, reconfiguration of system components and restoration of error-free operation of the neuromultimicroprocessor.

The greatest recovery efficiency in case of failures of a neuromultimicro-processor element is achieved at the hardware level. To restore processor operation means to restore the correct state of the processors. The software implementation of the control process based on breakpoints is ineffective and is determined by the application problem and the requirements for the reliability of the computing system. When a malfunction is detected, diagnostic tests isolate the malfunction and rule out the defective processor element.

A neural network for solving loosely coupled processes is based on the principles

of functional separation and has a structure consisting of multi-microprocessor sets, a communication network and a system module. Each processor element is able to independently solve the assigned tasks using the internal structure and exchanging messages over the communication network. The scenario for diagnostics, recovery and degradation is embedded in the system module. With this architecture, the multiprocessor has its own local and system resources available to all processor elements.

The base module has a bus architecture neuromultimicroprocessor and includes serial, parallel and local bus exchange, private resources and resources that are shared by processor subsystems. Such architecture of the basic module allows processor expansion with homogeneous and heterogeneous modules, having previously agreed on the bus exchange protocol. It is important to allocate system resources related to share multi-microprocessor resources and individual local resources. This allows you to organize the reliability, integration, performance, efficiency of the base module at the level of system resources. Local resources provide the functioning of individual consumer tasks: exchange with cloud resources, provision of high-performance computing, and receiving data from multidimensional information sensors. Local resources are isolated from errors that appear in other parts of the system, which increases the reliability of the entire system.

We introduce a message space for the synchronous exchange of information in blocks at maximum speed without using the processor resource. Basic modules are identified by the central service module, initializing all modules in the system by geographic principle. With the geographic distribution of the base modules, information is exchanged over a radio channel. The coding of the exchange channel, protection, exchange rate depends on the application. We use the local serial bus in the base module to control and diagnose system resources.

The capabilities of the multi-microprocessor are flexibly rebuilt. The function of the interface module is responsible for the transmission and reception of expected and unexpected messages, access to the components of the system module: I / O registers

and system memory. Dividing messages into expected and unexpected will optimize the transmission of short and long messages. The system module assigns arbitration, generates a system reset, provides data protection in the event of a power failure, and resumes starting when it occurs.

The work of a neuromultimicro-processor begins with initialization, transferring system components from an undefined state to a known one. The initialization process includes resetting, initializing individual modules, initializing the entire system, and booting. Initially, the processor boards have the same priority, during initialization, you can assign the modules by priority and assign the master, that is, assign the highest priority.

If the task is divided into N processes, then, if available, we assign N processor elements. Saving the state of a task involves saving the state of these processor elements. The device for storing the state of the task represents the stack memory, that is, the current state is the last write to the memory and is taken out first.

Suppose that the task is distributed among N processors ($i = 1, 2 \dots n$). Saving the state of the task means saving the state of the processors. Repeating the process is equivalent to restoring the states of the corresponding processors. Due to the ambiguity of the interaction of processes and the asynchronous nature of saving states, restarting other processes or multi-step recovery may be required. To solve the problem of reconfiguration, network management and other system issues, a system monitor and a switch controller are included in the neuromultimicroprocessor. The controller analyzes replays and multi-step recovery. Performance Monitor receives a command to execute processes and allocates processors and system memory for it. Physically, the system monitor is located in the system unit or a separate processor is allocated and is endowed with the functions of a monitor and a communication network controller.

When an error occurs, System Monitor indicates that the operation has been restarted, and if it repeats, all processor modules are suspended. It detects the failed processors and resumes solving the problem using the processors in which there were no errors. In the

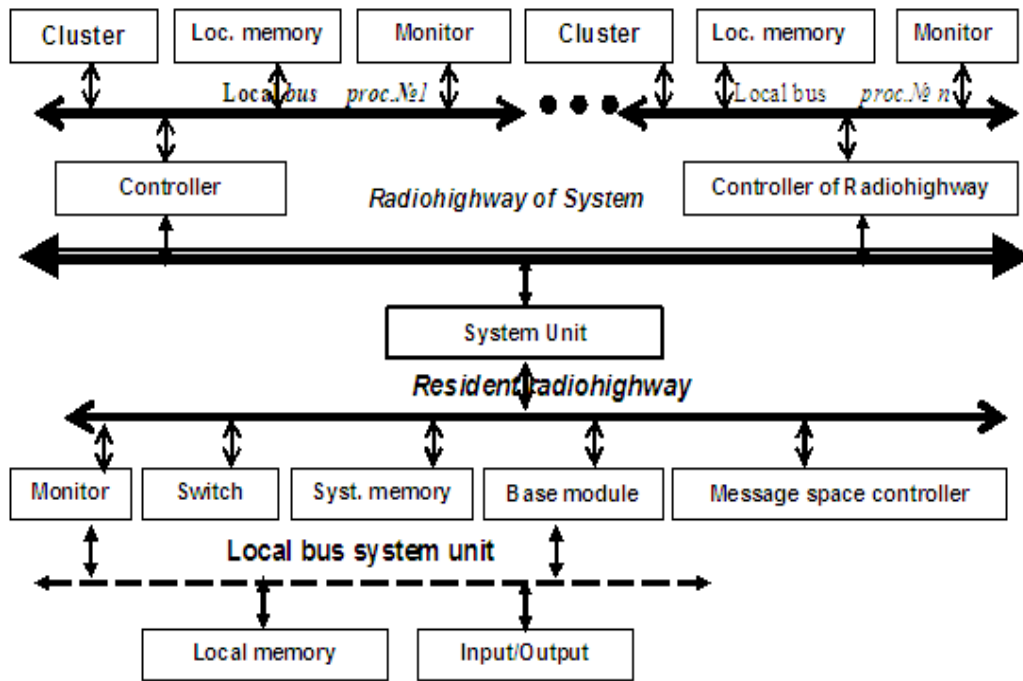


Fig.1. The architecture of an intelligent m-ensemble fault-tolerant neuromultimicroprocessor with a high-speed radio communication channel and a built-in recovery mechanism

presence of free processors, the task is redistributed taking into account them; in the absence of an adequate replacement, processes are distributed taking into account the available ones. When the number of processors decreases below the critical value, the system degrades, that is, processes are redistributed among the available processors, in which case the system performance decreases without deterioration in quality.

Let the failure of one processor module entail restarting a part of the computational process in all processor modules interacting with the data. We will assume that the restart process P_i entails a restart process P_j , that is, there is a propagation of restarts. Let's us denote the n-interval P_i as $T_i(n)$, and the initial time when P_i it retained its value as $t_i(n)$. Then, if a computation error occurs at the beginning of the time interval, and at the moment $t_i(n)$ the state is saved, interaction between processes is possible. In this case, the first process is restarted, and the interacting processes are suspended until the first is restored.

Consider a probabilistic model for estimating the restart propagation of a part of the computational process, with multi-step recovery. If an error occurs in calculations at a point in time t_s during the interval $T_i(k)$, a check is made for the possibility of recovery in one step.

This will enable us to evaluate the effectiveness of this built-in failover mechanism. We assume that the processor module has $(N + 1)ms$ cells for storing valid flags, and the task is distributed among M processor modules.

Let's calculate the range of multi-step recovery. It can be argued on the basis of the above stated that there was at least one call from module i to module j during the state saving interval. Formally, you can write it like this:

$$f_{ij} = f_{ji} = g_{ij} + g_{ji} - g_{ij}g_{ji}$$

where f_{ijn} : is the average probability of the propagation of restarts from the processor module i to the processor module j due to n-step recovery in the module i ;

$$g_{ij} = \frac{1}{N_1} \sum_{k=1}^{N_1} (1 - e^{-u_{jk}T_{ss}})$$

represents the average probability of access from module i to module j in one interval.

N_t : The total number of states of the processor unit saved until the completion of the task, provided that there were no failures.

$$N_t = \lfloor T_{ef} / (T_{ss} - T_{su}) \rfloor.$$

where: $T_i(k)$: Duration of the k -ro interval of the i processor module. Let us assume that the signal of persistence and self-preservation are equal in time. That is $T_{ss} = T_i(k) = \text{const}$.

T_{ef} : The total running time of the task, provided that there are no errors. The uptime is not included, for example, test sequences, recovery unit generation, etc.

T_{su} : Recovery unit generation time.

Because the total number of calls between modules i is j equal to

$$a_{ij} \left[T_{ef} / \sum_{m=1}^N a_{im} e_{im} \right] \text{ and} \\ \sum_{k=1}^N u_{ijk} (T_{ss} - T_{su}),$$

where: u_{ijk} The average intensity of the processor module i to the module j during k - ro the module interval i .

Suppose that the messages obey Poisson's law of distribution of the probability of a sequence of appeals. Given that the number of states of the processor module is large, it u_{ijk} can be considered constant during the k - ro interval, we obtain the following relation:

The maximum value of the intensity of memory u_{ijk} accesses must be less than or equal to the inverse value e_j , that is:

$$\frac{1}{e_{ij}} \geq (u_{ijk})_{\max} \geq u_{ijk} \geq 0$$

where E : Matrix $[e_{ij}]$, $i, j = 1, 2, \dots, M$, and $e_{i,j}$ represents the average execution time of calls from module i to module j .

Function f_{ij} - a monotonically increasing function from a bounded concave function of an argument g_{ij} has the maximum value when $u_{ij1} = u_{ij2} = \dots = u_{ijN_t}$ and the minimum value $[f_{ij}^N]$, when there are h intervals.

$$h = \frac{e_{ij} T_{ef} a_{ij}}{\left[(T_{ss} - T_{su}) \sum_{m=1}^M a_{im} e_{im} \right]}$$

where $u_{ijk} = 1/e_{ij}$, when $(N_t - h - 1)$ intervals and $u_{ijk} = 0$ when one interval,

$$u_{ijk} = \left\{ \frac{T_{ef} a_{ij}}{(T_{ss} - T_{su}) \sum_{m=1}^M a_{im} e_{im}} \right\} - \frac{h}{e_{ij}}$$

The value f_{ij} can be considered as the probability of a direct connection between

nodes i and j . To determine the probability of restarting r_{ij} : in the processor module j , we use calculations from the theory of network reliability:

$$r_{ij} = \bigcup_q (D_{ij,q})$$

where $D_{ij,q}$ represent the probability of what is q - way out of the node i to node j and \bigcup - probabilistic operation unification. Let us introduce an additional proposition that the occurrence of a fault in the static sense is uniformly distributed over the entire set of modules. Then the range of one-step recovery is determined as follows:

$$C(1) = (1/M) \sum_{i=1}^M \prod_{j=1}^M \left[1 - r_{ij} \left(1 - \sum_{k=1}^M b_{jk} \right) \right]$$

Calculations show that it is enough to have a small number of cells for registering states to achieve a satisfactory recovery result through restarting the task processes. Reducing access to resources leads to an increase in the memory value of the valid flags.

Architecture of a fault-tolerant neuromultimicrocomputer with a high-speed radio exchange channel for processing multidimensional radar signals

The optimal choice of architecture is ensured by its maximum approximation to the class of problems to be solved. Digital radar signal processing refers to the processing of signals that can be represented as a sequence of multidimensional arrays of numbers, such as sampling signals continuously varying over time from multiple sensors. Since field processing tasks are distinguished by a large amount of information that needs to be processed in real time, it is advisable to develop neural network ensembles in the form of a set of functional modules aimed at solving them.

Ensembles perform computations on data and interact with other ensembles to generate computations on distributed data. A multimicroprocessor ensemble organization defines an adaptive organization and distribution of functions for control, computations, data transfer and restructuring of a neural network in the process of a reliable solution of a given

task and ensuring the necessary fault tolerance. Information exchange between ensembles is provided by a wireless communication channel. We introduce the central processing unit as a system unit for testing, diagnosing and ensuring the initial startup of the neuromicrocomputer.

When building systems with an inter-ensemble exchange radio channel, we will use the system backbone for high-performance systems with advanced functionality associated with the presence of additional address spaces - message space and interconnection space. Using the architecture of a neural network with a radio channel and removing system bus operations from the central processor makes its processor independent.

The traditional way of communicating and transferring has been to use a shared memory space when using two or a maximum of three ensembles on the backbone, but it is not efficient for a larger number. As the number of neural network components grows, the time required to access the data increases unacceptably. More efficient is the exchange of data through the message space (see Fig. 1). Just as mail decouples sender and receiver, so processors are decoupled from the task of passing messages. To transfer a message, the sending ensembles prepare the message in a local buffer and indicate to the communication processor the transfer address. If the recipient is ready to accept the message, he gives his consent to the transfer of data. The coprocessors then perform the actual transfer and inform their processing units that the transfer is complete. This is the ability to implement a standard network protocol for data transmission and work within one neurosystem to various operating systems.

A neurocomputer can use a virtual interrupt scheme: an interrupt is carried out not by physical interrupt signals, but by transmitting a special interrupt message. A virtual interrupt is a message that contains the interrupt source address, destination, and qualified information.

The interconnection space allows for easy system configuration, simplifies unit testing and system reconfiguration. When a faulty module is found, the latter can be programmatically removed from the structure of the neuromultimicroprocessor and replaced with

another one in hot standby. Diagnostic software is located in each module. Each module can perform a built-in self-test. This operation can be carried out over the network from a remote terminal.

The transmission of information in the message space is transmitted continuously and the channel is not blocked. Thus, real-time mode is provided, information "freezing" during the exchange of ensembles modules is excluded. Messages are transmitted in quanta - data packets by means of a communication processor through the message space used to identify, configure and test the board. Each of the neurocomputer processors runs under its own operating system. To solve the problem of testing, initialization and initial loading of the system, neurocomputer architecture with a system bus has been developed, the physical transmission medium of which is a radio channel. The order of starting, testing and loading the system has been determined. Thus, openness, flexibility, and deep systemic development allow using the radio channel as a system bus for building highly reliable fault-tolerant neural network systems of high performance.

Neural network architecture is optimal for solving loosely coupled problems with natural parallelization. The tightly coupled central service module uses the radio channel at the bus-resident level, significantly increasing the bus bandwidth and ensemble performance as a whole and implements the following functions: system initialization at power-on, power supply control and switching to a backup source, timeout control. It can isolate failed modules, allowing other modules to continue to function normally. The functions of the central service module can be taken over by any other module in the event of its failure, without impairing the reliability of the neurocomputer.

The architecture of the real-time neurocomputer ensemble belongs to the systems of the MIMD type [40] with distributed memory and consists of a plurality of processors that autonomously execute various instructions on different data, i.e. are asynchronous systems with decentralized control.

The Central Processing Unit (CPU) architecture provides parallelism at the level of individual instructions, the level of loops and

iterations, the level of subroutines, the level of job steps, and the level of independent jobs and programs. Independent processor nodes provide parallelism at the level of individual instructions, but the efficiency of parallelization at the levels of loops and subroutines will already depend on the speed of the communication structure connecting the processor nodes. As for the levels of steps of a task and independent tasks and programs, they are usually associated with the multitasking mode of the system and when mapping tasks to individual processors or processor ensembles should not impose special requirements on the speed of data exchange between processors.

In a split-job system, supervisory functions are performed by each processor in accordance with its own needs and the requirements of the programs executed by that processor. Since the supervisor modules are executed by multiple processors, we provide for their re-entry or load copies of them into each processor. The number of conflicts associated with locking system tables is small, since each processor can have its own set. At the same time, the number of common control tables will not be large.

Systems with separate execution of tasks in each processor impose certain limiting requirements on the type of initial information, because systems work efficiently only when the tasks solved by individual processors of the system are well balanced, that is, they use the equipment approximately equally effectively. From the point of view of reliability, all processors in the system are a bottleneck, because failure of any processor means the loss of its program and violation of all program exchanges in which this processor participates. Restoring the system to work requires long-term external intervention. Extending the system without changing programs is impossible.

Systems with symmetric, or homogeneous, processing in all processors are most fully implemented when using a set of functionally homogeneous processor units. Each of the processors can equally effectively perform supervisory functions that "flow" from one processor to another and perform those supervisory functions that are inextricably linked to the problem being solved, and those functions that are necessary for a new task, in the case

when the current one is interrupted or completed completely. However, any processor can perform all or most of the system-wide functions. Due to the fact that the processors are homogeneous and can be used in the same way, any task during its execution can be processed by different processor units of the system. We use different sets of processors for its successful implementation. System-wide control is continuously redistributed between processors: at a time, only one processor can be the control one; a certain priority can be set for the processors, firstly, to resolve conflicts and, secondly, to rank control functions.

The neurocomputer does not impose strict requirements on the nature of the input information. When processor modules fail, performance gradually decreases (system degradation). The expansion of the system is possible without any functional limitations.

The peculiarity of the architecture of the neurocomputer lies in the combination of ensembles by means of a high-speed system highway with a physical transmission medium over a radio channel. A fast and efficient messaging service is implemented using a specialized operating system with a distributed kernel. The program being executed is represented as a set of simultaneously running processes that exchange data and synchronize their work by sending messages [41]. In other words, the program is viewed as a network of processes. The network consists of logical nodes, each of which contains a subset of processes that, from the point of view of the programmer, should run together in one physical node. The radio channel is divided between network subscribers in time, providing multiple accesses to the channel. There are no conflicts through the use of a code modulation radio channel. You can neglect the transit time of the signal through the radio channel. The concept of a network category as a short-range or long-range network loses its meaning. As the main characteristics when assessing the quality, we use the average message delay and the channel capacity (or the channel utilization factor, which is defined as the part of the channel capacity attributable to conflict-free transmission). A common way to match network traffic to incoming user requests is through flow control procedures.

Approbation of the technology for processing radar information was carried out in the SPE “Quantor” laboratory using an ensemble in the form of a multi-microprocessor for collecting radar information from multidimensional sensors in the THz range (75-115 GHz). Spatial configuration is provided by base stations in the 40 GHz range. The research was carried out for the study of hidden prohibited items, the study of the structure of the coatings of special devices, receiving through wall 3D-Image.

The possibility of 3D-radar calibration is studied in the exploring of material properties to the example of Plexiglas, depending on the distance between the sample and the antenna using an absorber. The results of preliminary studies indicate the possibility of measuring the thickness of the material.

In the implementation of 3D scanning small objects is used FMCW radar at operating frequency 100 GHz and bandwidth about 40 GHz of terahertz frequency range, Fig. 2.

We have developed algorithms and have obtained the required accuracy - less than 3 mm. In reality, we can accurately assess the environment model to take it into account in processing. For it we previously will try to calibrate the radar.

On the calibration, a small metal plate and several measurement cycles for averaging the noise were used. It is shown that the accuracy of measurements is influenced by the width of the radiation pattern, the number of

measurement cycles at one point, the accuracy of positioning and moving the head during the measurements, and the time interval between the calibrations.

As a result of the measurement cycle, a frequency dependence of the attenuation in the microwave channel $D(f) = U_{ref}(f)/U_{inc}(f)$ was obtained.

Unknown parameters of the dielectric structure are determined by procedure of global minimization of discrepancy between the measured attenuation in channel $D(f)$ and one calculated theoretically $D_{th}(f, p)$

$$F(\mathbf{p}) = \sum_f |D(f) - D_{th}(f, \mathbf{p})|^2$$

Here $D_{th}(f, p)$ is defined according to the formula

$$D_{th} = \left| k_0 + k_1 \frac{V - V_c}{(1 - k_3 V)(1 - k_3 V_c) - k_2 V V_c} \right|^2$$

and $k_0(f), \dots, k_3(f)$ are complex coefficients, which are determined experimentally using reference samples and describe properties of the microwave channel; f is the frequency of sounding waves; $V_c(f)$ is the complex reflection coefficient (CRC) of the reference arm 3; $V(f, p)$ is a theoretically calculated CRC of the dielectric structure, which depends on a vector of the structure parameters p (thickness of layers and electrical parameters of materials).

We consider that in free space extends a plane electromagnetic wave and normally in-

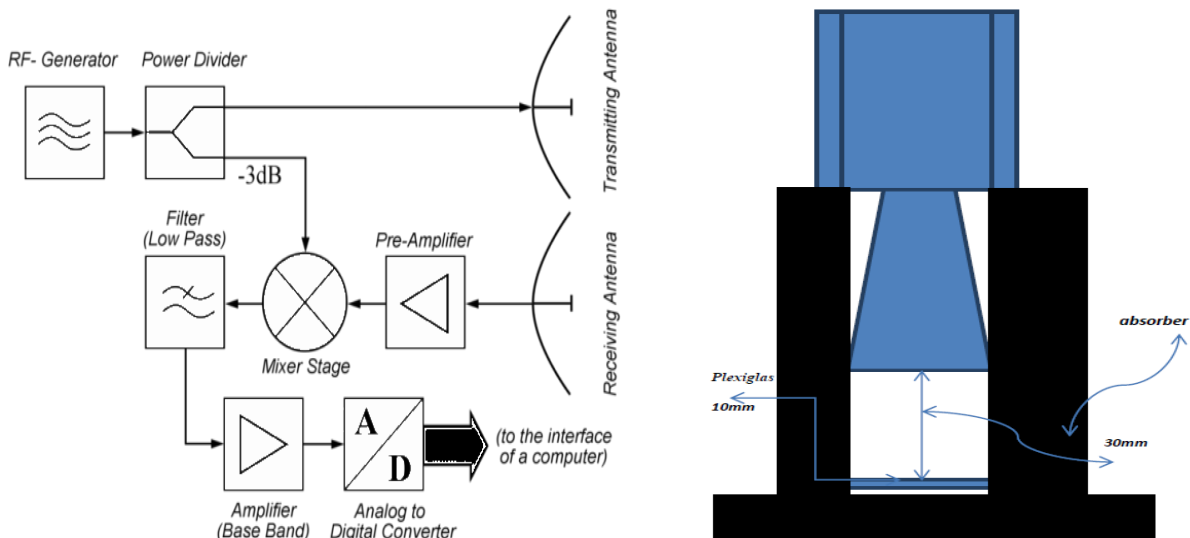


Fig.2. System for testing 3D FMCW Terahertz Radar

cident on the infinite ($M-1$)-layer medium with flat boundaries. The CRC $V(f, p)$ is related of the CRC of the structure in free space $V_s(f, p)$ through the scattering matrix of the antenna S , which is determined experimentally:

$$V = S_{11} + \frac{S_{21}V_s}{1 - S_{22}V_s}$$

The CRC of the structure in free space depends on the thickness and electrophysical parameters of structure layers:

$V_s = V_s(f, h_1 \dots h_m, \epsilon_1 \dots \epsilon_m, \text{tg}\delta_1 \dots \text{tg}\delta_m)$, where $h_m, \epsilon_m, \text{tg}\delta_m$ is thickness, permittivity and loss tangent of m -th layer. The CRC of the plane wave from dielectric plane-layered medium $V_s(f, p)$ is determined by the known formulas:

$$V_s = \frac{W_0 - Y_1}{W_0 + Y_1}$$

We see an Average Basic Function (BF) of 40 response signal from 6x6mm metal at 40 different distance — estimation of Non Removable response from constructive elements (horn and others), and Average BF of 40 response signal from Absorber Only without metal plane. [2] We can see a small difference between Absorber Only Average BF and Calibration (by metal) Average BF (Fig.3).

Step x 104 Average BF and BF Series after 0 compensation. SPC “Quantor”, Ukraine

We have developed algorithms and have obtained the required accuracy - less than

3 mm. But in reality, we cannot accurately assess the environment model to take it into account in processing. For it we will test the radar system, having previously calibrated it.

Conclusions

This article discusses the architecture of intelligent fault-tolerant radar systems based on a neurocomputer. Fault tolerance is provided by varying the ratio of performance and reliability with a shortage of reliability resources of a real-time neural network. We briefly got acquainted with resource management, special attention was paid to the impact of fault tolerance on the reliability resource and the ability to manage it when solving an applied problem. The discussion covered the issues of building a hypothetical model of a real-time multiprocessor with a resource of performance and reliability, as well as their relationship. The prototype of the original neurocomputer operating system was the real-time operating system “RMX-86”. The architecture of this real-time multiprocessor system is determined by the applied task of processing radar information. The elements of fault tolerance and survivability were introduced into the system, initial and diagnostic test support during the execution of an applied task, control of the system backbone, and majorization.

When forming an article on neural network systems for processing radar infor-

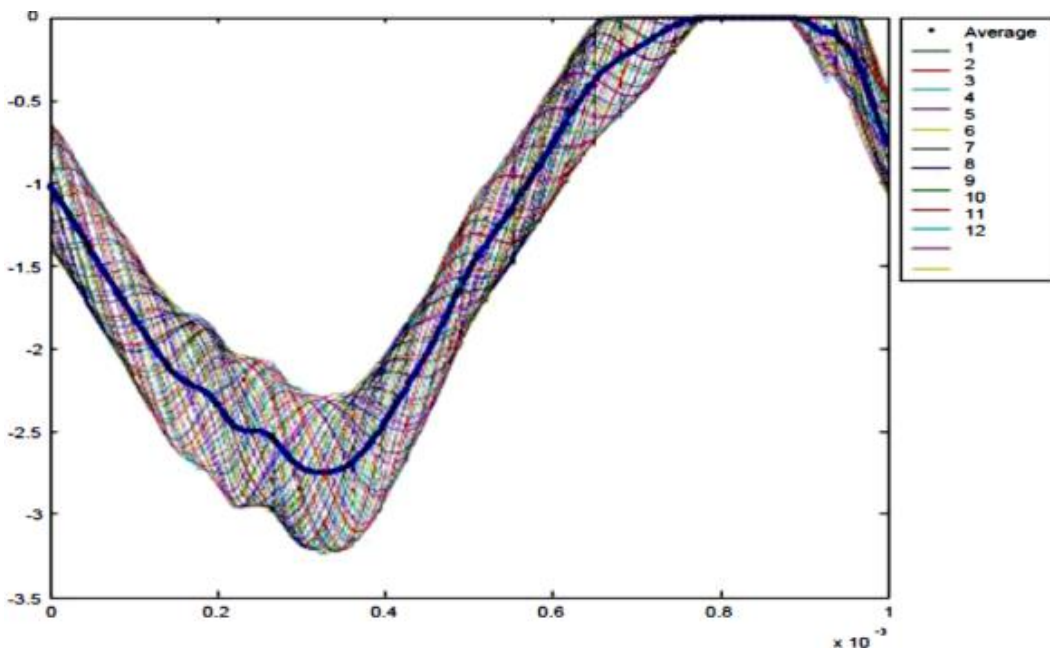


Fig.3 Comparing and Average BF and Series of BF

mation, the number one task was to highlight the principle of organizing the computational process. Unfortunately, the volume of the article did not allow highlighting the issues about the basis of the organization of macro-pipeline calculations of a fault-tolerant neurocomputer, the description of abstract system resources and the management of the reliability resource of a fault-tolerant real-time neurocomputer, the organization of the computational process for solving an applied problem of multidimensional radar information. Also, the issues of building a modeling complex for solving radar problems on neural network structures with deep learning are not covered. In the future, special attention will be paid to the tools for debugging a neurocomputer, since the debugging tools of a multiprocessor are difficult to design, and debugging a neurocomputer requires the design of original tools that are not supplied in a finished form by manufacturers of conventional processor components.

Collection, processing, technical implementation features, testing, diagnostics, calibration of systems based on neural networks will be covered later. They do not affect the general understanding of the organization of computations using neural networks, although the difference is significant in comparison with conventional calculators. In our last works, the elements of construction of THz radars, LPI of marine radars, construction of trained neural radar networks are touched upon. [42] Modeling of processes in a neural network showed the nonlinear nature of the system's behavior from the influence of external and internal disturbances of various powers, information in a neural network is often heuristic. The reliability of computations is ensured by the fault tolerance of neural networks and depends on the reliability resource.

Biomimetic methods are gaining popularity in the construction of radar systems and one of its important characteristics. That is, the external environment affects the operation of the radar system and, in turn, the radar system generates a sounding signal, takes into account the indicators of the environment. Most clearly it sees when using radar in an IoT environment. The second

important biomimetic indicator is “reticular function”. And if early cognition was realized through adaptability, albeit to an incomplete extent, then the reticular function was realized through the fault tolerance of multiprocessors, but in practice it was not implemented at all, due to the high cost of design. For the first time, we applied reliability improvement by fault-tolerant methods in control systems, collection, processing and display of information on board Ukrainian aircraft AN of the Antonov Design Bureau, where the author was the Chief Designer of the fault-tolerant multiprocessor system. When using neural network technologies, this function is modified and becomes more understandable, providing “homeostasis” of the neural network system for collecting and processing multidimensional information. An example of the use of cognition and a complete disregard for reticularity was demonstrated by the latest publications on research on radar technologies within the framework of NATO, where issues related to the fight against failures, solutions to the problem of fault tolerance, survivability, and dependability of computations were practically ignored. Neural network architecture assumes the solution of both problems as interconnected.

References

1. P. Barros, G.I. Parisi, C. Weber, S. Wermter, Emotion-modulated attention improves expression recognition: a deep learning model, *Neurocomputing* 253 (2017) 104–114. Learning Multimodal Data
2. Applications & Challenges of Deep Learning in the field of bioinformatics *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 15, No. 7, July 2017
3. 2 [10] X. Chen et al., “Multi-view 3D object detection network for autonomous driving,” in *Proc. CVPR*, 2017, pp. 6526–6534.
4. S. Ren et al., “Object detection networks on convolutional feature maps,” *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 39, no. 7, pp. 1476–1481, Jul. 2017.
5. Steen, K., Therkildsen, O., Green, O., Karstoft, H.: Detection of bird nests during mechanical weeding by incremental background

- modelling and visual saliency. *Sensors* 15(3), 5096–5111 (2015)
6. Wu, X., Yuan, P., Peng, Q., Ngo, C., He, J.: Detection of bird nests in overhead catenaries system images for high-speed rail. *Pattern Recogn.* 51, 242–254 (2016)
 7. Wang, Christopher Rasmussen (B), and Chunbo Song. Fast, Deep Detection and Tracking of Birds and Nests Qiaosong Department of Computer and Information Sciences, University of Delaware, Newark, DE, USA cer@cis.udel.edu Springer International Publishing AG 2016 G. Bebis et al. (Eds.): ISVC 2016, Part I, LNCS 10072, pp. 146–155, 2016. DOI: 10.1007/978-3-319-50835-1_14
 8. Author J K. Classification of human cancer diseases by gene expression profiles. *App Soft Comput.* 2017; 124:134.
 9. Bard E, Hu W. Identification of a 12 gene signature for lung cancer prognosis through machine learning. *J of cancer.* 2011; 148-156.
 10. R. J. Cintra et al., “Low-complexity approximate convolutional neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 5981–5992, 2018.
 11. D. Tomè et al., “Deep convolutional neural networks for pedestrian detection,” *Signal Process., Image Commun.*, vol. 47, pp. 482–489, Sep. 2016.
 12. J. Ngiam et al., “Multimodal deep learning,” in *Proc. ICML*, 2011, pp. 689–696.
 13. Z. Luo et al., “Non-local deep features for salient object detection,” in *Proc. CVPR*, 2017, pp. 6593–6601.
 14. Seonwoo Min, Byunghan Lee, Sungroh Yoon, Deep Learning in Bioinformatics| Briefings in Bioinformatics, Briefings in Bioinformatics, 2017; 18(5), , 851–869
 15. J. Zhang, W. Li, P.O. Ogunbona, P. Wang, C. Tang, Rgb-d-based action recognition datasets: a survey, *Pattern Recogn.* 60 (2016) 86–105, doi: 10.1016/j.patcog.2016.05.019.
 16. Haohan Wang, Bhiksha Raj, and Eric P. Xing. On the origin of deep learning. *CoRR*, abs/1702.07800, 2017a.
 17. Joel Moniz and Christopher J. Pal. Convolutional residual memory networks. *CoRR*, abs/1606.05262, 2016.
 18. Chen CL, Mahjoubfar A, Tai L-C et al. Deep Learning in Label-free Cell Classification. *Scientific reports* 2016.
 19. S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, 1945 in: *NIPS 2017*, 2017.
 20. A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, L. V. Gool, Weakly supervised cascaded convolutional networks, in: *2017 IEEE Conference 1238 on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5131-1239 5139. doi:10.1109/CVPR.2017.545.
 21. Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, A. Torralba, MIT Saliency Benchmark, 2015, (<http://saliency.mit.edu>).
 22. Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *CoRR*, abs/1712.03480v1, 2017. URL <https://arxiv.org/abs/1712.03480v1>.
 23. Chunxiao Jiang, Haijun Zhang, Yong Ren, Zhu Han, Kwang-Cheng Chen, and Lajos Hanzo. Machine learning paradigms for nextgeneration wireless networks. *IEEE Wireless Communications*, 24(2):98–105, 2017.
 24. Xenofon Foukas, Georgios Patounas, Ahmed Elmokashfi, and Mahesh K Marina. Network slicing in 5G: Survey and challenges. *IEEE Communications Magazine*, 55(5):94–100, 2017.
 25. U. Iqbal, A. Milan, J. Gall, PoseTrack: joint multi-person pose estimation and tracking, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
 26. D. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, *Proceedings of the 3rd International Conference on Learning Representations*, 2015.
 27. J. Zhang, W. Li, P.O. Ogunbona, P. Wang, C. Tang, RGB-based action recognition datasets: a survey, *Pattern Recogn.* 60 (2016) 86–105, doi: 10.1016/j.patcog.2016.05.019.
 28. Ivan Garvanov, Lyubka Doukova, Vladimir Kyovtorov, Christo Kabakchiev. COMPARATIVE ANALYSIS OF CFAR STRUCTURES FOR GPS SIGNALS IN CONDITIONS OF INTENSIVE URBAN PULSE INTERFERENCE Institute of Information Technologies Bulgarian Academy of Sciences.
 29. Z. Cao et al., “Realtime multi-person 2D pose estimation using part affinity fields,” in *Proc. CVPR*, 2017, pp. 1302–1310.
 30. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In: 2016

- IEEE International Conference on Image Processing (ICIP). IEEE; 2016. pp. 3464-3468
31. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: 2017
 32. IEEE International Conference on Image Processing (ICIP). IEEE; 2017. pp. 3645-3649
 33. Sercan "Omer Arik, Mike Chrzanowski, Adam Coates, Greg Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech. CoRR, abs/1702.07825, 2017.
 34. S. H. Khan et al., "Cost-sensitive learning of deep feature representations from imbalanced data," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3573–3587, Aug. 2018.
 35. Khan G, Ghani MU, Siddiqi A, Seo S, Baik SW, Mehmood I, et al. Egocentric visual scene description based on human-object interaction and deep spatial relations among objects. Multimedia Tools and Applications. Vol. 77. Springer; 2018. pp. 1-22
 36. Thenmozhi K, Shanthi S. Optimized Data Retrieval in Big Data Environment using PPFC Approach AJRSSH. 2017; 683-690.
 37. C. Feng, M. Cui, B.-M. Hodge, J. Zhang, A data-driven multi-model 1914 methodology with deep feature selection for short-term wind forecasting, 1915 Applied Energy 190 (2017) 1245{1257.
 38. S. Bouktif, A. Fiaz, A. Ouni, M. Serhani, Optimal deep learning lstm 1855 model for electric load forecasting using feature selection and genetic all 856 algorithm: Comparison with machine learning approaches, Energies 11 (7) 1857 (2018) 1636.
 39. Avizhenis A. Fault tolerance is a property that ensures the continuous performance of digital systems. / IEEE - 1978. – Vol.66 – №10 - p.5-15.
 40. Flynn M.J., Some computer organizations and their effectiveness // IEEE Trans. Comput. - 1972. - Vol. C-21, No. 9. - pp. 948-960.
 41. Palagin A.V. About computers with virtual architecture// CSM. –1999.-№3.-p.33-43.
 42. *Kosovets M., Tovstenko L.* The practical aspect of using the artificial intellectual technology for building a multidimensional function CFAR for smart-handled LPI Radar.

Magazine "Programming Problems". ISSN 1727-4907. 2020. №2-3. Special issue. Proceedings of the thirteenth international scientific-practical conference on PROGRAMMING. UkrPROG'2020. September 16-17, 2020 Kyiv, Ukraine. Pages 202 - 212. <http://www.pp.isofts.kiev.ua>.

Received: 20.05.2021

About the authors:

Mykola Kosovets

Leading Constructor,

Number of scientific publications in Ukrainian publications -53

Number of scientific publications in foreign publications -14

Index Hirsha -2

<https://orcid.org/0000-0001-8443-7805>

Scopus Author ID: 5644007500

Lilia Tovstenko

Leading Software Engineer

Number of scientific publications in Ukrainian publications -21

Number of scientific publications in foreign lands -6

Index Hirsha -2

<https://orcid.org/0000-0002-3348-6065>

Scopus Author ID: 56439972800

Affiliations:

SPE "Quantor"

03057, c. Kyiv-57, str. E.Potye, 8-A

Ph.: (380)66-2554143

E-mail: quantor.nik@gmail.com

Institute of Cybernetics of Glushkov

National Academy of the National Academy Sciences Ukraine

03187, Kyiv-187, Academician Glushkov Avenue, 40.

Ph.: (380)67-7774010

E-mail: 115lili@incyb.kiev.ua

V. L. Shevchenko, Y. S. Lazorenko, O. M. Borovska

INTONATION EXPRESSIVENESS OF THE TEXT AT PROGRAM SOUNDING

As the amount of media content increases, there is a need for its automated sounding with the built-in means. The factors influencing the intonation were analyzed, the dependences of sound characteristics in accordance with the intonations were mathematically described. In the course of the work, the numerical analysis of sentences was improved using the moving average for smoothing audio, approximation lines for generalization of emotions as functions, and Fourier transform for volume control. The obtained dependences allow to synthesize intonations according to the punctuation, emotionally colored vocabulary and psycho-emotional mood of the speaker. Software for emotional sounding of texts was developed, which provides the perception of audio information easier and more comfortable based on the use of built-in processors of mobile devices.

Key words: text analysis, sound characteristics, intonation expressiveness.

Introduction

Most of the information comes to us in a graphical and audio representation. Therefore, the automation of sounding texts is an urgent problem. At the same time it is necessary to bring the intonation of the voice synthesized by the computer as close as possible to the human one. Usually monotonously read text is processed manually by a person.

The development of automated means of emotional sounding of the text is somewhat constrained due to the complexity of the task and, consequently, uncertainty about the success of its solution based on available mobile devices with built-in processors. At the same time, to solve similar problems in related fields, such as recognizing dangerous situations based on smartphone sensors, examples of successful solutions using machine learning exist [1].

In our case, to introduce emotion (intonation), in addition to machine learning methods, it would be desirable to identify and use mathematical patterns of texts. Therefore, the topic of the study of the formation of intonation for different emotions is relevant.

Analysis of the State of Research Issues

Theoretical research of intonation from the philological point of view was actively investigated by Bagmut A.Y. [2], [3]. Her works provide a very detailed analysis of intonations according to the syntactic and semantic features of sentences – some monographs contain an analysis of only a narrative sentence with equal intona-

tion. But at the same time, the intonations given to speech depending on the psycho-emotional state of the speaker, i.e. the emotions themselves from the read text, have been little studied.

Minnigalimov R.T. [4] mathematically described the patterns of changes in the frequencies of the fundamental tone for narrative affirmative and negative, as well as interrogative sentences. However, his practical experiments sometimes contradict each other. The author explains this by the fact that different speakers have different reading styles, so it is necessary to increase the statistical base of speakers for further practical development.

American experts from AT&T Laboratories worked on the synthesis of voices for sound. They created a program that imitates the human voice after processing 10-40 hours of real recording. However, as noted by the developers themselves, the program is not yet able to fully reproduce the voices of real people, and the sound of synthesized recordings is quite technical and does not take into account the emotions of the speaker when sounding [5].

Last year, the British team Sonantic tried to add emotion to the computer voice. The project uses artificial intelligence, which analyzes large amounts of human records [6]. The synthesized voice is indeed similar to a natural sound, but so far the development is focused only on the negative emotion of despair and crying.

The main problem is that in practice everyone reads the same text in their own way, keeping only the basic intonation or mood. The aim of the

work is to identify techniques and characteristics of voice change to increase the expressiveness of speech and formalize these features.

One of the available approaches to sounding texts is monotonous reading with constant pitch, volume, etc. And without pauses. This method of transmitting audio information requires constant focus and independent logical division of the listened text into syntactically integral fragments in content.

Another improved approach is to read in an even voice, but with punctuation pauses. So, after a comma there is a minimum pause, and, for example, after a dash or a point – more. This method facilitates the perception of the text due to the fact that syntactically whole units of speech (whole sentences or their parts) are perceived separately due to pauses [7]. The disadvantage is the lack of intonation difference between the fragments of the text. This approach is used in the built-in libraries of the Python programming language, in applications for viewing text files with the sound function, browsers, etc.

One of the best existing approaches is machine-based sounding. This method generates a number of sound effects for a certain set of emotions. However, the “assignment” of such an emotion to a particular sentence is done manually, i.e. the linguistic features of the text itself are not taken into account. This solution is used, for example, in the British startup Sonantic [6].

The contradiction between these approaches and practical needs is that the principles of intonation in the sound of texts are formulated rather vaguely and are based on human “sense of language”, which is often explained by the skills of expressive reading in philological sources. However, if a person is able to unambiguously determine the mood of a sentence, then, probably, there are certain patterns in how it does it and that includes such a “sense of language.” From this we can conclude that the found dependences can be generalized and formalized, to bring them closer to the concept of rule. This can be a good simplification and basis for the algorithms used in machine learning.

Relationship between Intonation and Punctuation

The main lexical means of denoting emotions in the text is punctuation [8]. It gives

instructions on how the sentence should begin and end, what interaction with the listener is envisaged, what feelings should be evoked in him, how long and frequent pauses should be endured. Emotions formed due to intonation do not depend on the speaker, his manner of narration and feelings, style of text and content, target audience, etc. That is, they will be dim, but always the same. If a comma and a dash indicate only the need for a pause and its length, then the key role in determining the intonation is played by punctuation at the end of the sentence.

According to the emotional color the sentence could be exclamatory and non-exclamatory. So, they have or do not have an exclamation mark at the end. They differ in how important the emphasis on their content is and determine how strongly the information will impress the listener [8].

Invocative sentences are pronounced in a calm voice, without extreme increase or decrease in pitch and volume, not oversaturated with accents on words and their meaning.

Exclamations are pronounced more sublimely, loudly, in a higher tone, the intonation may be less smooth, with tears and bright accents on keywords. Schematically, the differences are presented in fig. 1 – for volume and fig. 2 – for height.

In order to express a sentence, there are narrative, interrogative and motivating ones. Narrators report an event, fact, or phenomenon. At the end of such sentences a full stop is placed, sometimes – three full stops to indicate incompleteness of thought. If there is a full stop at the end, the intonation throughout the sentence remains equal, and at the end it drops, the voice subsides. The volume is kept evenly at the same level, decreasing fairly quickly at the end of the sentence. This expresses the completeness of thought and confidence in what is being said.

If the sentence ends with period, the decline of intonation is smoother, moderately fading, the voice subsides more, but gradually. The opinion is not complete, there is room for the listeners’ own thoughts and their personal assessment of what has been said. Often it is to enhance this effect at the end of a sentence with ellipsis is a longer pause [8]. The exclamation mark in such a sentence often expresses anxiety associated with feelings of fear.

Since in most cases with increasing volume, the pitch of the sound also increases, because the accent and expression are provided by both means, in the future they are accepted as one set of sound characteristics. However, in this case the approximation rejects the jump of volume important for the exclamatory sentence.

Motivational sentences also have a full stop at the end. They differ from narrative content: express a request or demand, and - from purpose: motivate to action. Intonation have a strong emphasis on keywords [7], which, in fact, determine the motivation for action (ask, tell, bring, etc.). Usually such words appear at the beginning of a sentence, inversion is rare and is rather an atypical phenomenon for motivational sentences. Therefore, both the increase in voice and the increase in volume are characteristic of the beginning of a sentence.

Interrogative sentences have a pronounced logical emphasis on the most significant word and the strengthening of intonation at the end, which is illustrated in fig.3. If such a sentence contains interrogative words (how ?, where? etc.), then they are logically emphasized. Then at the end of the sentence the intonation decreases, as in narrative sentences. The degree of amplification of sound characteristics determines how important it is to focus on this issue [9].

Construction of Intonation according to Punctuation

For further work with intonation and program processing of sentences, we will enter some mathematical designations.

The word consists of syllables: $w = \{n, \dots, n, k, n, \dots, n\}$, where n – unstressed syllable, k – stressed syllable.

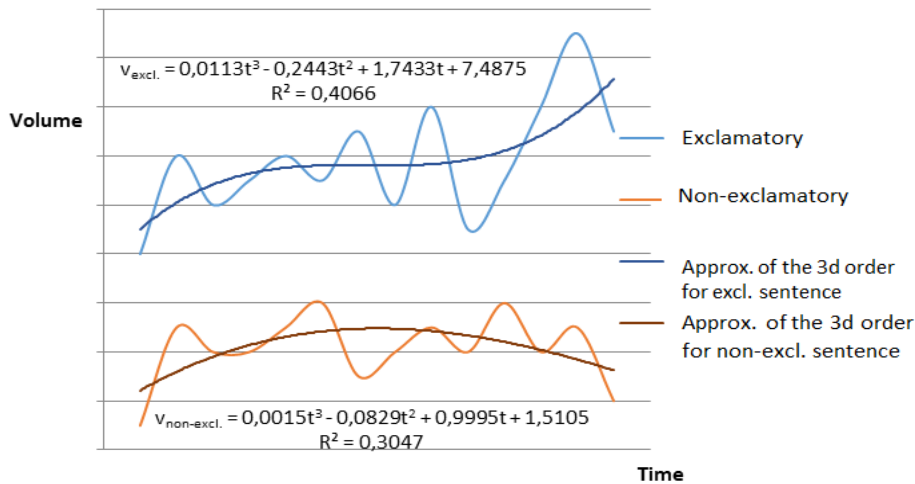


Fig. 1. Changing the volume of exclamatory and non-exclamatory sentences

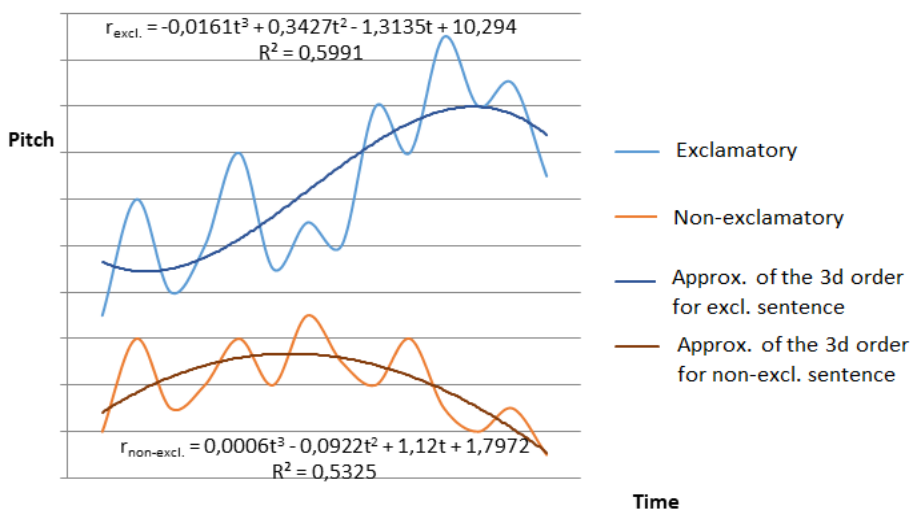


Fig. 2. Changing the pitch of exclamatory and non-exclamatory sentences

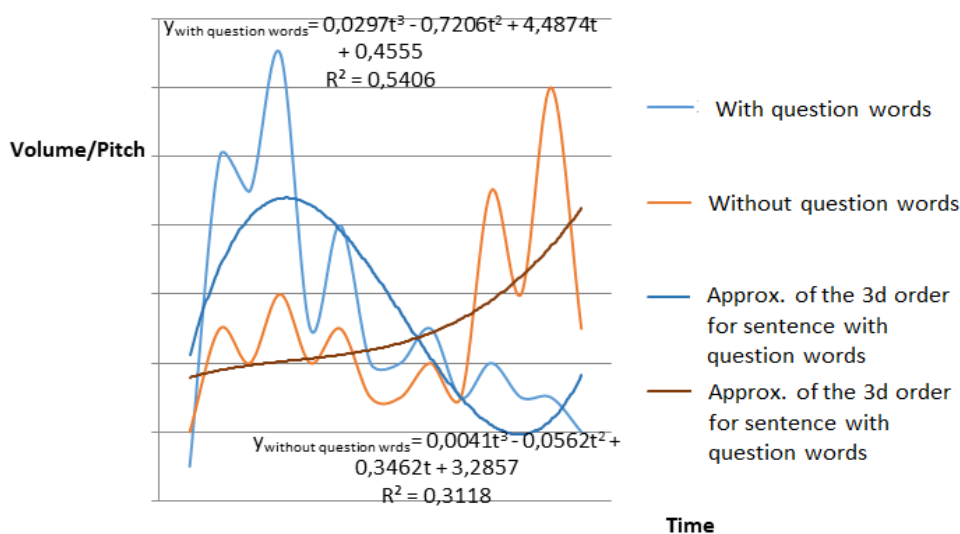


Fig. 3. Changing the sound characteristics of interrogative sentences

The sentence consists of the words:

$s = \{q, \dots, q, l, q, \dots, q\}$, where

q – logically unstressed word,

l – logically emphasized, the most significant word.

But a sentence can also be represented as a sequence of syllables, then you can write it like this:

$s = \{n, \dots, n, k, n, \dots, n\}$, where

n – syllable of a logically unstressed word or any unstressed,

k – stressed syllable of the most significant word.

Along with the tuple of syllables, we will write a tuple that contains commands for the computer. They contain information about the change of the main technical parameters of the sound. Enter the appropriate symbols for them:

- volume – v ;
- length – t ;
- frequency (tone of sound) – r ;
- pause (before/after the syllable) – p .

Then the phrase (sentence or word) for the computer will look, for example, like this: $s = \{t, v, r, q, \dots, v, r, q, \dots, v, r, p, q\}$ – the duration of the whole record is set, and then before each syllable indicates the required volume and frequency, if necessary, a pause, followed by the text of the syllable.

Consider the basic emotions that cover most human sensations. We introduce their corresponding notations, some of them are opposite to others:

- joy – j ;
- sadness – j' ;

- aggression (attacker's reaction) – a ;
- confusion / anxiety (protective reaction of the victim) – a' ;
- calm – c ;
- irritation / dissatisfaction – c' .

Each emotion has its own characteristic pattern in time, which is represented by a set of functions of the main technical parameters of sound, using the previous notation:

$v = j(t); r = j(t)$ – for joyful intonation;

$v = j'(t); r = j'(t)$ – for sad;

$v = a(t); r = a(t)$ – for angry, aggressive;

$v = a'(t); r = a'(t)$ – for anxious;

$v = c(t); r = c(t)$ – for calm;

$v = c'(t); r = c'(t)$ – for irritated.

To work with audio recording, we turn the emotions of sentences into functions for processing arrays, because the sound signal during processing is usually given as a set of values like an array.

For program work with the text we will read it from a file. In the beginning it is enough to use ready means of sounding and to receive monotonous reading of the text. It is inconvenient to work with an integral sound file – change of any characteristic will be superimposed on all sound series. Therefore, it is necessary to divide the record into a number of identical fragments and give them a numerical representation. Therefore, for further sound processing, the conversion of the sound series into an array of volume values with a certain frequency is

used. The wave format is used as standard for such purposes.

During sounding it is necessary to work simultaneously with the text (to process sound according to punctuation), and directly with its audio representation. The text from the file is generally a string. So, first we divide the text into an array of sentences, the following punctuation marks will serve as delimiters: «.», «!», «?», «...». Then in a cycle we process each of them. To determine the final punctuation mark, we analyze the sentence character by character from the end.

The main intonation emphasis is not given to the whole combination of words, but focuses on one syllable of the keyword in the phrase. The increase in intonation may be more or less smooth, but not sudden. Therefore, the syllable (or several syllables) before the most intonationally outlined and after it will also be somewhat pronounced. This will help smooth out the difference in volume and pitch and make the sound more natural and pleasant.

Multiplication of values by a certain factor should not be used to amplify the sound characteristics of a piece of audio recording. This approach can give an unexpectedly strong or too small result. In addition, if the recording has a large amplitude of volumes or pitches, the highest of them can be extremely amplified, which will lead to unpleasant intermittent sound and poor sound quality due to the appearance of noticeable noise. To adjust the sound characteristics, it is better to add a certain number to their values. You need to consider how quiet the original recording was and what the initial pitch was, so that the adjustment is not too sharp when adding a large

number or, conversely, the number is not too small and the changes are not noticeable. If the original recording already had inhomogeneous volume and pitch, it should be generalized.

The arithmetic mean of the values of one of the characteristics does not always give the desired result: for example, if there are values that are several times greater than the bulk [10]. Therefore, you must first take the middle range, discarding too large and small values. This can be done by averaging or other smoothing methods. You can also approximate an array of values. For such purposes, linear is enough, because the nature of the function itself does not interest us. The goal is to select the range in which most values are concentrated. Schematically, the principle of this method is illustrated in fig. 4. The middle (vertical) range used for further calculations, the upper and lower ranges contain discarded values.

Another way to discard redundant values, more convenient for software implementation – setting the lower and upper limits of acceptable values. After that, you can take the average value or mode as the basic initial value of a characteristic. The result of setting such thresholds is constructed by software tools and the principle is shown in fig. 5, where the value of the initial recording frequencies is indicated in the upper and lower ranges, and the selected range is indicated in the middle (vertical) range.

For software work with text and sound, a function was created that first sounds a given text from a file, and then converts it into an array of values and amplifies its corresponding fragments according to the desired condition. This is the main function, which is then referenced by oth-

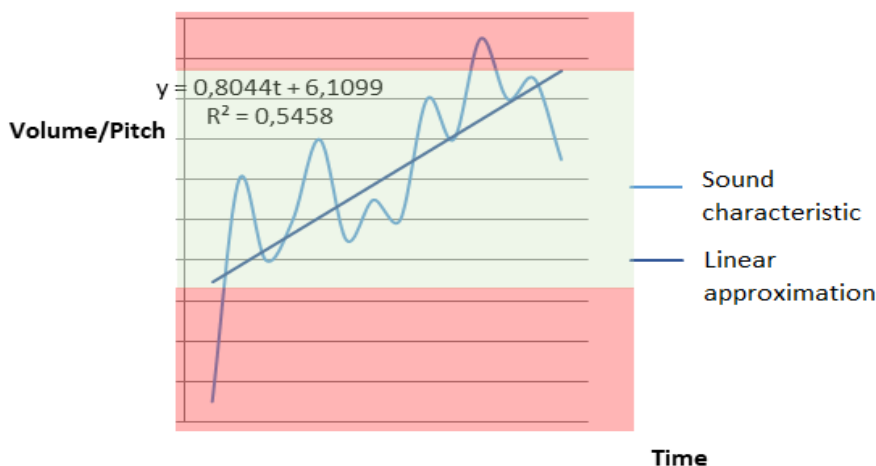


Fig. 4. Selection of the average range of values by means of linear approximation

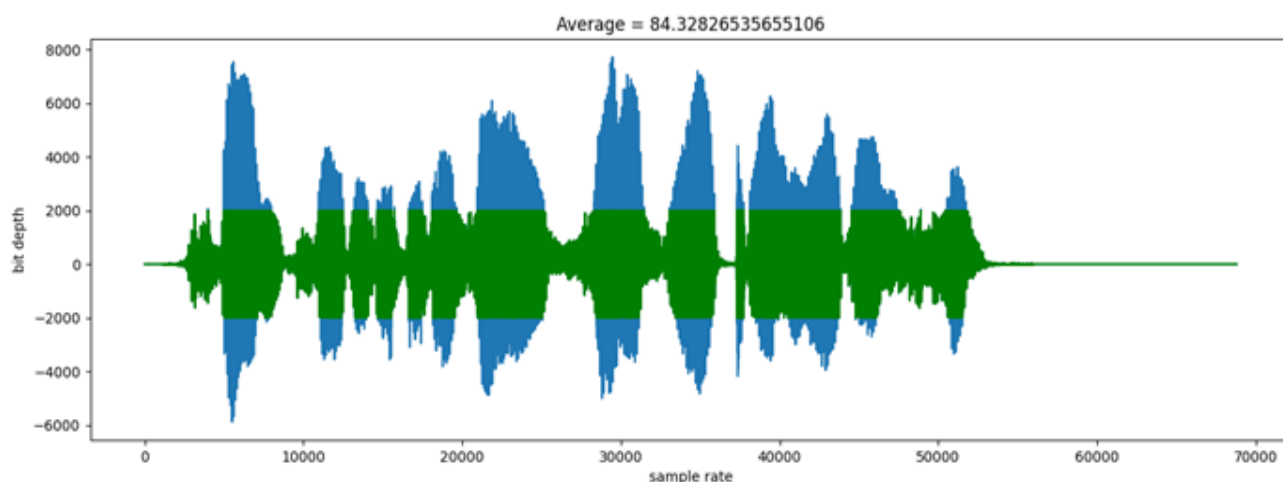


Fig. 5. Select a range of values for averaging

ers. It also controls the imposition of several gain conditions on the same fragment. For example, if the interrogative word «when» is defined as an adverb, it should be strengthened twice – as a question word and as an emotionally colored word (because it is adverbs and adjectives that most often indicate the shade in the meaning) [10]. This can have the undesirable effect of a sharp drop in values and an increase in volume or tone too much. Therefore it is necessary to check up, whether the word was strengthened already on other sign, and in that case not to strengthen it repeatedly (or to strengthen much less).

In addition to placing emphasis on words that belong to certain parts of speech, in natural language a person intonationally expresses a part of a phrase that contains objections. The usual marker of such words in the text is the negative part «no». Thus, the words after it are also emotionally amplified in the software implementation. To find such words in a sentence, a function is created that determines the ordinal numbers of words with a negative shade of meaning.

To speed up writing, we use a function that discards single values from an array with a certain step. At not very small step distortions of a sound are almost not appreciable and are corrected in the subsequent smoothing. A similar function to slow down the recording, on the contrary, adds elements. That is, it also duplicates single values with a certain step.

Fourier transform was used to shift the tone of the sound in the work. This allows you to make the voice both lower and higher while almost completely preserving the original re-

ording time. The degree of deterioration of sound quality is proportional to the length of the audio fragments with which it is processed – the smaller the pieces of the recording undergo changes, the better the sound.

Construction of Intonation that Expresses the Feeling of the Speaker

In general, all emotions can be divided into 3 groups: positive, negative and neutral.

The first group expresses high spirits and satisfaction. Such emotions are high in tone of voice, with normal or slightly increased volume, slight rhythm, but with smooth differences, ascending intonation of phrases. At the end of the sentence, the voice subsides smoothly, but not stretched. Polynomial approximation is used in the work to determine the general nature of the decrease or increase of graphs of sound parameters and their comparison with other emotions. It most accurately reflects intonation changes. In this case, the approximation of the 2nd order is not enough - the differences in volume or height are not taken into account, although they are important for building emotions. The approximation of the 4th and higher orders approaches the initial graph too accurately, reflecting even minor fluctuations in the voice. The best option is the 3rd order: sufficient smoothing of graphs is provided, the largest differences of values remain. Graphic generalization of positive emotions is shown in fig.6.

So you can mathematically generalize the functions of positive emotions. Since the

best result of the approximation is achieved in the 3rd order, the function is cubic. The coefficients of the approximation lines can be rounded, because the goal is an approximate form of the function. Then for volume and height of positive intonations (we will take a joyful voice as a basis) accordingly it turns out:

$$v = 0.01t^3 - 0.17t^2 + 1.34t + 1.63,$$

$$r = 0.01t^3 - 0.16t^2 + 1.11t + 0.95,$$

where t is the time.

Similar generalized functions are built for all the emotions considered below.

The second group, on the contrary, means unpleasant feelings, dissatisfaction with something and denial. If positive emotions, in general, are very similar in nature and sound, then negative ones give a much wider palette of such patterns.

To further improve our approach, we will analyze the features and differences of emotions in the text from a philological point of view in more detail.

The main difference in sound from the positive is the lack of smoothness, gradation of the signal. Also, most (not all) negative emotions are characterized by a decrease in pitch. But the volume can be both reduced and increased, depending on the severity of the emotion and the purpose of its expression. Let's analyze these shades in more detail.

As a rule, when a person feels irritated or dissatisfied, his voice becomes lower. But the feeling of fear and anxiety is accompanied by the opposite phenomenon: the voice becomes louder, very inhomogeneous, smooth and longer. Often vowel sounds are lengthened

and amplified, while consonants are lost and replaced by short pauses in live speech caused by minor sudden breaths.

An increase in volume indicates an «attacking» mood. Such emotions arise under critical psychological stress, develop rapidly and grow intonationally. This is evidenced by the sharp and confident ending of sentences expressing similar emotions in speech. Acceleration or a gradual increase in the speed of sound is sometimes used for additional expression.

The opposite tool (decrease in volume and stretching of phrases) is a protective reaction, excitement, confusion and helplessness. Such experiences depress a person, worsen mood, well-being, reduce productivity, prudence. The range of such soft emotions is extremely wide: sadness, grief, confusion, despair, guilt and many others. They actually differ in the root cause, i.e. in the text - in content. Intonations are almost identical, therefore, the program requires a single implementation.

Neutral emotions characterize a calm, balanced state of the speaker. In such emotions there are no jumps of sound characteristics, the voice is smooth. Usually correspond to narrative unpronounceable sentences. An example of such emotions is interest or indifference.

Programmatically, the main difference from the construction of intonation on the basis of punctuation of the sentence is that changes should be applied to the whole sentence, and not only to its individual parts (words, syllables). This will ensure a smooth transition of emotion, sound quality and proximity to natural human language.

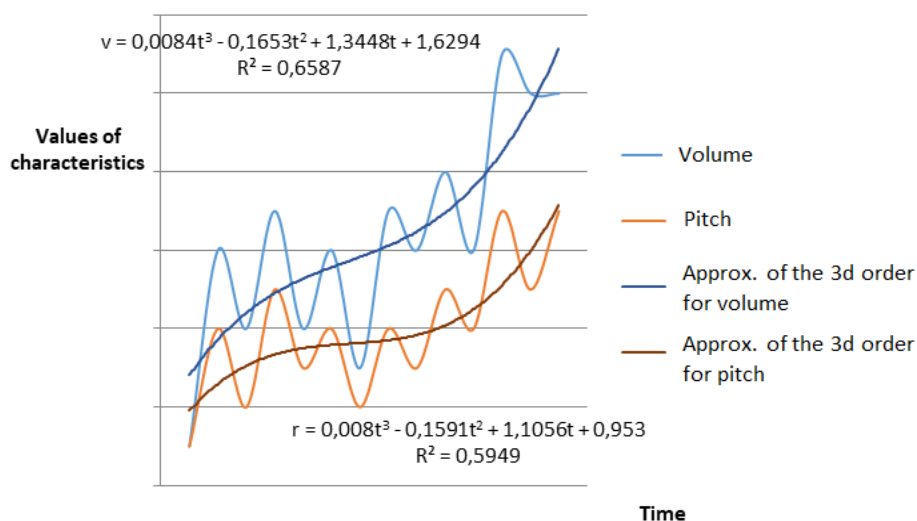


Fig. 6. Changing sound characteristics over time for positive emotions

Since it is not necessary to analyze the text itself to give the phrase a certain emotion based on the speaker's feelings, it was possible to process the recording without the initial textual representation. The main goal is to divide the sound series into fairly small phonetic units. Due to the fact that accents (both strong and weak) in pronunciation fall on vowel sounds, it is advisable to divide the recording into syllables, because one syllable contains exactly one vowel sound.

In the model, the soundtrack was divided into different fragments over time, for example, 0.5s. But there is a problem: the basic phonetic unit, which may contain an accent, is the syllable, but the syllables differ in duration of sound. Accordingly, after processing the recording, the accents may be misaligned, which will affect the sound quality.

Therefore, another method was used in the work. If we compare the graphical representation of the volume array of a record and the text version of the phrase, we see that the volume fluctuations occur in each syllable. This is illustrated in fig.7.

Thus, the volume increases at the vowel sound, and at the consonant level it decreases and almost completely subsides at the hissing and whistling. From this follows the conclusion that it is possible to break the audio recording into fragments-compositions, i.e. particles of amplification-attenuation, if you set the threshold of "silence". We will assume that the values that are higher than this threshold correspond to vowel sounds - the key component of the syllable (sound), and those that are lower - consonant (silence). Then we will pro-

cess the sound separately in parts (audio fragments of phrases of text or syllables), and then combine them back into one file.

It is these fragments of silence that serve as dividers when splitting a record into syllables. By default, it is assumed that the reading occurs at a speed of 100% (the value can be both lower and higher) and a volume of 1 (values from 0 to 1). The optimal "silence" interval for determining the composition limit is 50 ms, and the silence threshold is approximately minus 30 dB (dBFS). Then to adjust the silence time, reduce it by the same percentage as increase the speed by one percent (for 150% of the speed, the silence time will be approximately 40ms). To adjust the silence threshold, reduce the volume by the same percentage and reduce the threshold by the same percentage, i.e. increase the modulus of the threshold value (module, because this value remains negative). Thus, for a volume of 0.8 we obtain a threshold of silence minus 36 (approximately minus 40).

Conclusions

1. The speech techniques and voice characteristics that give emotional color to the read text were studied in the work; the regularities of change of sound characteristics for transfer of various intonations are formalized and programmatically realized.

2. The paper proposes a method of formalizing emotions when breaking sentences into syllables and mathematical formalization of patterns of change of sound characteristics of the synthesized voice in accordance with the intonation of sentences; proposed formulaic dependences for different types of sentences

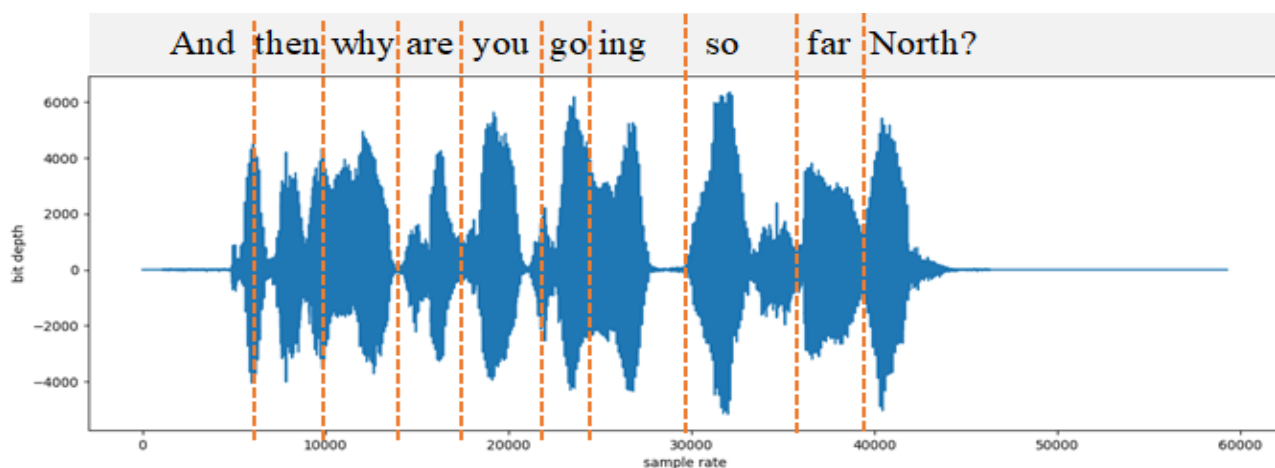


Fig. 7. Graphic correspondence of text and voice

by syntactic and semantic features; improvement of the method of numerical analysis of sentences using the method of moving average, approximation lines and Fourier transform; improving the method of sentence synthesis taking into account the given emotions with the help of lexical and syntactic analysis of sentences.

3. Special software has been developed in the Python algorithmic language, which allows you to voice text with appropriate intonations based on the use of built-in mobile processors.

4. In further research it is planned to investigate how the use of pauses - both short in words and larger in a phrase - affects the change of intonation; keep in mind that the final punctuation marks can be several: «?!», «!!!», «? ..», etc., so that the intonation may have different shades.

References

1. Shevchenko V. Dynamic Objects Emergency State Monitoring by Means of Smartphone Dynamic Data / Shevchenko A., Bychkov O., Shevchenko V. // 2017 14-th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). Proceeding. - Polyana, February 21-25, 2017. - p.292-294. <http://ieeexplore.ieee.org/document/7937138/> DOI: 10.1109/CADSM.2017.7916138
2. Bahmut A.Y. Semantics and intonation in the Ukrainian language. – 1991
3. Bahmut A.Y. Intonation structure of a simple narrative sentence in Slavic languages. – 1970
4. Minnihalimov R.T. Analysis and synthesis of Ukrainian speech / Minnihalimov R.T., Kyiv, 2015. - 90 p.
5. Official site AT&T Laboratories: https://about.att.com/sites/labs_research
6. Official site Sonantic: <https://www.sonantic.io/>
7. Blyznychenko L. A. Intonation of speech. – 1968
8. Peshkovskiy A. M. Punctuation marks and scientific grammar. – 1918
9. Peshkovskiy A. M. Intonation and grammar. – 1928
10. Zagumennov A. P. Computer sound processing [Electronic resource] / Zagumennov A. P. - Moscow: DMK Press, 2006.-- 384 p. : ill. - ISBN. - Text: electronic. - URL: <https://znanium.com/catalog/product/407267>

About the authors:

Viktor L. Shevchenko,
Dr.Sc., Prof., Professor of Software systems and technologies Department of Taras Shevchenko National University of Kyiv.
Publications - more than 300.
Publications in foreign scientometric publications – 17.
H=3.
<https://orcid.org/0000-0002-9457-7454>.

Yana S. Lazorenko,
Bachelor student
Publications – 2.
<https://orcid.org/0000-0002-3987-2338>.

Olena M. Borovska, Chief designer
at the Institute of
Software Systems of the National Academy of Sciences of Ukraine.
Kyiv, 03187, Acad. Hlushkov avenue, 40
building 5
Publications - 4

Affiliations:

Program system and technologies
Department
Taras Shevchenko National University
of Kyiv,
Bohdan Hawrylyshyn str. 24
UA-04116, Kyiv, Ukraine
E-mail: gii2014@ukr.net,
yana_lazorenko@knu.ua

Department of Automated Organizational
Management Systems (№23)
Institute of Software Systems of the National
Academy of Sciences of Ukraine.
Academician Hlushkov Avenue, 40, building
5, Kyiv, Ukraine, 03187.
E-mail: e.borovskaya@nas.gov.ua

Received: 18.05.2021

A.A. Triantafillu, M.A. Matashko, V.L. Shevchenko, I.P. Sinitsyn

ALGORITHM AND SOFTWARE FOR DETERMINING A MUSICAL GENRE BY LYRICS TO CREATE A SONG HIT

One of the needs of music business is a quick classification of the song genre by means of widely available tools. This work focuses on improving the accuracy of the song genre determination based on its lyrics through the development of software that uses new factors, namely the rhythm of the text and its morpho-syntactic structure. In the research Bayes Classifier and Logistic Regression were used to classify song genres, a systematic approach and principles of invention theory were used to summarize and analyze the results. New features were proposed in the paper to improve the accuracy of the classification, namely the features to indicate rhythm and parts of speech in the song.

Keywords: genre, rhythm, song, text, classification.

Introduction

The role of music in the modern world is difficult to overestimate - we hear it everywhere. It is hard to determine what affects the listener's consciousness more - the musical component of the song or its lyrics. In the era of streaming services that allow you to listen to music legally and unlimitedly for a small fee, music is a large and profitable business. Spotify, Apple Music and other companies are constantly trying to improve their recommendation algorithms. But how will new to the streaming service users find music they like? They will choose a selection of songs by genre, which they already enjoy. Besides, it is not always clear to young songwriters who to offer their songs, it is hard to understand in which genre they would be most successful potentially. Therefore, it was decided to develop theoretical approaches and corresponding software for mobile and embedded digital systems that would help songwriters and could be used to improve music recommendation algorithms.

Analysis of existing studies and task statement

To date, there are many algorithms of analyzing the genre of the song by musical component, but no product allows you to predict the success of song in a specific genre not only by the content of lyrics or by melody, but also by its rhythm.

In 2019, researchers from the University of South Carolina developed a system using artificial intelligence that recognizes the

song genre by lyrics and chords, and their model was trained on more than 5,500 songs that were typical for their genres. The research was closely related to song chords, but the researchers did not perform rhythm analysis [1].

In addition, numerous studies use the «bag-of-words» method to train a model to predict the genre of a song, which allows you to present each song as a set of «important» words. Important words are the words that have a meaning and do not serve just to bind other words. Adam Sadovsky and Xing Chen use approximately 150 songs of 4 genres (rap, hip-hop, rock and country) and they classify the genre of a song by using the «bag-of-words» technique and a special function to determine the weight of the lyrics [2].

The disadvantage of these studies, although the methods of creating features for models and the learning models themselves differ, is that the general essence remains the same: the prediction is carried out only on the basis of the words from the lyrics or its melody and the ways of processing are repeated. The difference of our research is that in addition to words importance metrics, we use features based on the song lyrics rhythm and morpho-syntactic structure, in order to increase the accuracy of the prediction result.

From the analysis of existing theoretical methods and researches it follows that there is a need for new methods to increase the accuracy of song genre classification. The aim of the scientific work is to increase the accuracy

of determining the song genre by its lyrics by developing software that uses new features, namely the rhythm of the lyrics and its morpho-syntactic structure.

The aim of the study is to analyze researches on classifying song genre by different factors, create a program to determine the rhythm of the lyrics, create a program to determine the number of different parts of speech in the text, train several models to determine the most effective one and compare the results for different models and features.

The scientific novelty of the work lies in the usage of a new method in conjunction with well-known ones to predict the genre in which the song is most likely to become popular by lyrics component.

Description of theoretical methods used

Our work on the analysis of the lyrics consisted of two parts: feature engineering and training of the model on created features. Creating features is the transition from a text representation of lyrics to a numerical representation since text cannot be the input of a machine learning model.

The following metrics were selected as features:

1) ratio of stressed, unstressed syllables to the number of all syllables in the song, the number of undefined words in a song;

2) TF-IDF (term frequency-inverse document frequency);

3) ratio of the number of different parts of speech to the number of all words in a song.

We selected two models, the Bayes classifier and Logistic Regression, and compared their results. The Bayes classifier is a classic model for solving such problems, but since we use features of different types, we assumed that Logistic Regression will show better results.

Preprocessing. The prediction is done using data from the MetroLyrics Dataset, which is a collection of songs-related data, including genre and lyrics from www.metrolyrics.com site. This data allows to analyze the lyrics and find its distinct features. Unfortunately, this file is currently not freely available, so it was taken from a similar project of researchers from the University of California [3]. First of all, all songs written in non-English were removed

from the dataset using the DETECTLANGUAGE feature from Google Sheets service. MetroLyrics dataset consists of more than 360 thousand songs in different languages. Since the dataset contains very few entries of Other, R&B, Indie and Folk genres, it was decided not to use them in model training. In addition, all songs without genre or without lyrics, songs labeled «instrumental», corrupted recordings (with a set of characters instead of lyrics) were removed. The distribution after data cleaning I displayed on Fig. 1.

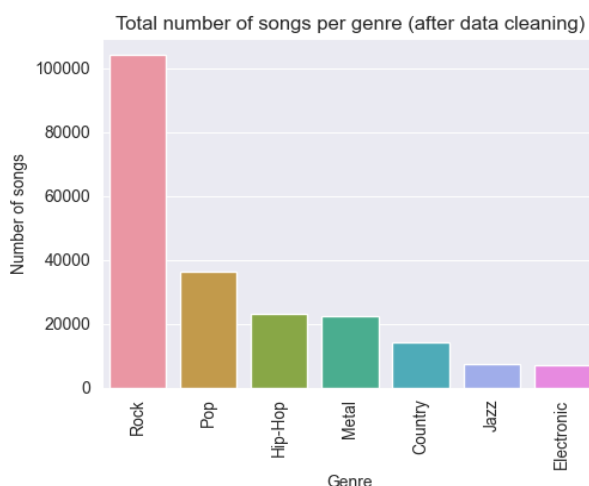


Fig. 1. Total number of songs per genre (after data cleaning)

Determining the rhythm of a song.

The method consists in presenting lyrics in the form of a binary sequence, where 1 is the stressed syllable, 0 is the unstressed syllable. The text is pre-cleaned of punctuation and reduced to a unified form - all the letters in the words text are in lowercase.

Let the set L be the set of rows l_1, l_2, \dots, l_n : $L = \{l_1, l_2, \dots, l_n\}$. Then the set W - is a set of words w_1, w_2, \dots, w_n in a row: $W = \{w_1, w_2, \dots, w_n\}$.

From a phonetic point of view, each word consists of sounds that in turn can be vowel and consonant. Research is conducted using songs in English, and the sounds in words are identified with the help of Carnegie Mellon University Pronouncing Dictionary [4], which consists of more than 134,000 words and their pronunciation. The set of vowels V , according to the Carnegie Mellon University Pronouncing Dictionary, looks like this: $V = \{ 'AA', 'AE', 'AH', 'AO', 'AW', 'AY', 'EH', 'ER', 'EY', 'IH', 'IY', 'OW', 'OY', 'UH', 'UW', 'Y' \}$

According to the CMU Dictionary, if there is a number one at the end of the sound designation (for example, “AH1”), then the stress in the word is primarily placed on this sound. Let S be a sentence where the words *a*, *b*, *c* are stop-words (words which serve only as a connection for other words): *S* = “A *d*, *b* *e* - *F*, *c* *a* *e*, *g*.”

Since the number of syllables corresponds to the number of vowel sounds and a vowel sound is present in each syllable, the algorithm of creating a binary rhythm sequence, which was proposed in the work, looks like this:

Algorithm 1

1. Break the text into lines;
2. Tokenize (break into elements) the line: *S* = [“A», «d», «,», «b», «e», « - «, «F», «,», « c», «a», «e», «,», «g», «.»]
3. Convert all the letters to lowercase: *S* = [«a», «d», «,», «b», «e», « - «, «f», «,», « c», «a», «e», «,», «g», «.»]
4. Clean the text from punctuation: *S* = [«a», «d», «b», «e», «f», « c», «a», «e», «g»]
5. For each word in a string:
 - 5.1 If the word is in the dictionary:
 - 5.1.1 For each sound in the word:
 - 5.1.1.1 If the sound is vowel:
 - 5.1.1.1.1 If the sound is stressed:
 - 5.1.1.1.1.1 Add 1 to rhythm;
 - 5.1.1.1.2 Otherwise:
 - 5.1.1.1.2.1 Add 0 to rhythm;
 - 5.2 Otherwise:
 - 5.2.1 Add «_» to the rhythm (notation of unknown words (abbreviations, neologisms, profanity, etc.)).

Text representation in a form of «Bag of Words». The Bag of Words model considers only the frequency of words appearing in the text and converts the document into a vector of numbers. Each word is assigned a unique number and the number of occurrences in the document. Our work uses an existing CountVectorizer model that uses this principle. Let *S* be the sentence used in the previous example. Then algorithm of CountVectorizer looks like this:

Algorithm 2

1. Tokenize sentences (Algorithm 1, p.2);
2. Unify the words (Algorithm 1, p.3);
3. Remove punctuation (Algorithm 1, p. 4);
4. Remove stop-words: *S* = [«d», «e», «f», «e», «g»]

5. Assign an index and a number of occurrences to each word: *S* = [«d», «e», «f», «e», «g»]; Index = [«d»:0, «e»:1, «f»:2, «g»:3];

$$\text{Result} = [«d»:1, «e»:2, «f»:1, «g»:1]$$

Statistical metric TF-IDF. TF-IDF (TF - term frequency, IDF - inverse document frequency) is a metric used to assess the importance of a word in a document that is part of a collection of documents. A weight of a word *i* in a document *j* is calculated this way:

$$\omega_{i,j} = tf_{i,j} * idf_{i,j} = tf_{i,j} * \log\left(\frac{N}{df_i}\right) \quad (1)$$

Where $tf_{i,j}$ is the number of occurrences of the word *i* in the document *j*; df_i - the number of documents in which there is a word *i*; *N* - the number of documents.

Let *S* be the sentence used in previous examples. In our work we used an existing TfidfVectorizer model with our own modification of the algorithm. By default, it works according to the following algorithm:

Algorithm 3

1. Tokenize sentences (Algorithm 1, p.2);
2. Unify the words (Algorithm 1, p.3);
3. Remove punctuation (Algorithm 1, p. 4);
4. Find TF of the words: *S* = [“a”, “d”, “b”, “e”, “f”, “ c”, “a”, “e”, “g”]; TF = [“a” : 1, “d” : 1, “b” : 1, “e” : 2, “f” : 1, “ c” : 1, “a” : 1, “g” : 1]
5. Find IDF of the words (depends on the frequency of words in all documents);
6. Vectorize a normalized result.

To improve accuracy, we modified the algorithm by creating our own function for text preprocessing. Let *S* be the sentence used in previous examples. Then the algorithm looks like this:

Algorithm 4

1. Tokenize sentences (Algorithm 1, p.2);
2. Unify the words (Algorithm 1, p.3);
3. Remove punctuation (Algorithm 1, p. 4);
4. Remove stop words (Algorithm 2, p.4);
5. Lemmatize words (treat word forms as one word): *S* = [“d”, “e”, “f”, “e”, “g”]
6. Find TF of the words: *S* = [“d”, “e”, “f”, “e”, “g”]; TF = [“d” : 1, “e” : 2, “f” : 1, “g” : 1]
7. Find IDF of the words (depends on the frequency of words in all documents);
8. Vectorize a normalized result.

Determining parts of speech in songs. To determine the entry of different parts of speech in the song, the pre-trained

model «en_core_web_sm» from the SpaCy library was used. We used this model on songs that were pre-tokenized and lemmatized. The library functionality returns a tag to indicate a part of speech for each word. The feature is created by counting the number of different parts of speech and foreign words (not English) and their ratio to the number of all words in the song.

Bayes classifier. Since the analysis of text data involves long and multidimensional feature vectors, the learning algorithm should be effective both in terms of classification and in terms of computational speed. These qualities are present in the Bayes training model (Naive Bayes). The method involves dividing the feature into several independent variables and finding an estimate for each of them.

Naive Bayes classifier is based on Bayes theorem, which is an equation that describes the relationship between conditional probabilities. For our question, the Bayes classifier calculates the probability of a particular genre for a given feature. According to Bayes theorem, this probability is calculated as follows [5]:

$$P(G|f) = \frac{P(f|G)P(G)}{P(f)} \quad (2)$$

Where G is the genre, and f is a feature. The feature is divided into independent metrics from the metric (feature) vector, so:

$$P(f|G) = P(f_1|G) * P(f_2|G) * ... * P(f_n|G) \quad (3)$$

The Naive Bayes classifier has certain drawbacks, namely the fact that the absence of a word in the document (in our case, song) has the same weight as its presence. Obviously, this affects the accuracy of the results because the song is defined by the words that are present in it, not absent. In addition, this classifier does not take into account the frequency of words, which, of course, is extremely important for the song analysis.

To solve these problems, we used a modification of the Naive Bayes classifier – the Multinomial Bayes classifier. It works as follows: the following formula is used to divide the feature into independent metrics:

$$P(f|G) = N! * \prod_{i=1}^k \frac{p_i^{n_i}}{n_i!} \quad (4)$$

Where p_i is the probability of the word i appearing in all songs of genre G ; n_i – the

number of occurrences of this word in this song; N – the number of all words in the song.

Typically, the features used by the classifier are only features to indicate the frequency of occurrence of words in songs (TF-IDF features, for example), but we have added our own features to them to indicate the rhythms of songs and their morpho-syntactic structures. For example, let X be a set of input features, and Y is a set of labels to indicate genres. x_w – classical features to indicate the frequency of words in songs (BoW, TF-IDF), x_r – our features to indicate the rhythm, x_s – our features to indicate the number of different parts of speech in a song, y – genres. In the classical solution of such problem, the sets X and Y look like this:

$$X = x_{w1}, x_{w1}, x_{w2}, \dots, x_{wn} \quad (5)$$

$$Y = y_1, y_2, \dots, y_n \quad (6)$$

After we added our features, the sets began to look like this:

$$X = x_{w1}, x_{w1}, x_{w2}, \dots, x_{wn}, x_{r1}, x_{r1}, x_{r2}, \dots, x_{rn}, x_{s1}, x_{s1}, x_{s2}, \dots, x_{sn} \quad (7)$$

$$Y = y_1, y_2, \dots, y_n \quad (8)$$

Logistic Regression. By definition, Logistic Regression is intended for the classification of binary classes. Since we use 7 genres for classification, we have chosen the type of logistic regression that can be used to predict more than two classes, namely Multinomial Logistic Regression.

The difference between a Multinomial Logistic Regression and a classic one is that instead of a standard logistic function [6] or a sigmoid function that predicts the probability of a binary event by comparing it with a logistic curve (sigmoid), a softmax function or a normalized exponential function is used, which compresses all values to the range [0,1] and the sum of all elements is 1. The normalized exponential function gives the answer in the form of the probability of the event, which can take more than 2 values. The classifier, which is based on a standard logistic function, calculates the probability of a result Y for a given feature X as follows [7]:

$$P(Y|X) = \frac{e^{f(x)}}{e^{f(x)} + 1} = \frac{1}{1 + e^{-f(x)}} \quad (9)$$

Where $f(x)$ is a function that consists of the features x , and their weight β , which is assigned to the features by the classifier:

$$f(x) = x_0 + x_1\beta_1 + \dots + x_n\beta_n + \varepsilon \quad (10)$$

The normalized exponential function, in turn, looks like this:

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (11)$$

As is the case with the Bayes classifier, the features, which are commonly used for such studies, were combined with our own ones. For example, let X be a set of input features, and Y is a set of labels to indicate genres. x_w – classical features to indicate the frequency of words in songs (BoW, TF-IDF), x_r – our features to indicate the rhythm, x_s – our features to indicate the number of different parts of speech in a song, y – genres. In the classical solution of such problem, the sets X and Y look like this:

$$X = x_{w1}, x_{w1}, x_{w2}, \dots, x_{wn} \quad (12)$$

$$Y = y_1, y_2, \dots, y_n \quad (13)$$

After we added our features, the sets began to look like this:

$$X = x_{w1}, x_{w1}, x_{w2}, \dots, x_{wn}, x_{r1}, x_{r1}, x_{r2}, \dots, x_{rn},$$

$$x_{s1}, x_{s1}, x_{s2}, \dots, x_{sn} \quad (14)$$

$$Y = y_1, y_2, \dots, y_n \quad (15)$$

Analysis of results

To analyze the results, we created a structural and logical scheme of the research (Fig.3-8). Each scheme block has the following structure (Fig. 2):

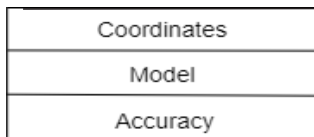


Fig. 2. Structure of structural and logical scheme

To understand the scheme, we need to enter some notations:

- rhythm 1 – 3 features: ratio of the number of stressed syllables, unstressed syllables, unrecognized words to the number of all syllables;
- rhythm 2 – 10 features: 3 of rhythm 1, percentage of unfamiliar words, number of lines, average number of syllables per line,

number of words, average number of syllables in a word, average number of letters in a word, average number of letters in a line;

- parts of speech 1 – the ratio of the number of parts of speech to the number of all words, without removing stop-words;
- parts of speech 2 – the ratio of the number of parts of speech to the number of all words, with the removal of stop words.

The scheme is divided into parts according to the model and the main feature: CountVectorizer (Bag of Words) or TF-IDF (Term Frequency – Inversed Document Frequency). Italics highlight the best accuracy result from the part. The result of accuracy was calculated under the following conditions:

- 70% of the data was used to train the model, and to verify it the remaining 30% was used.
- the random state parameter was set to the same value for each experiment. This is necessary so that the same songs always fall into the test set of songs.

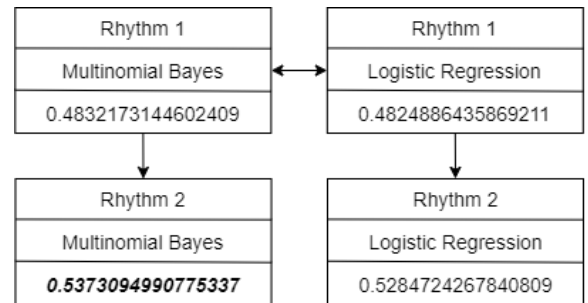


Fig. 3. Structural and logical scheme, p.1

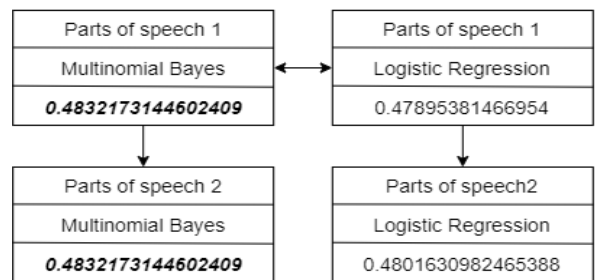


Fig. 4. Structural and logical scheme, p.2

From the results on the scheme it became obvious that the assumption that the Logistic Regression would give a better result was confirmed. To assess the depth of innovation we created the following table (Table 1):

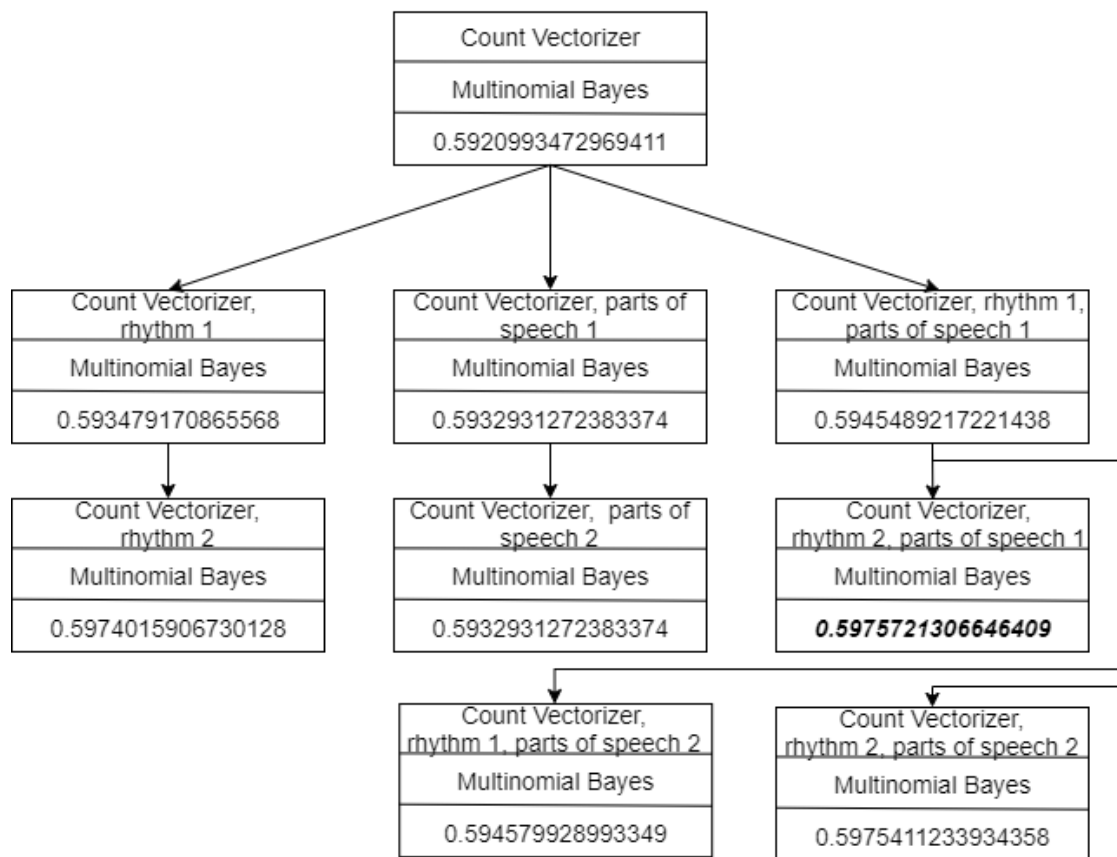


Fig. 5. Structural and logical scheme, p.3

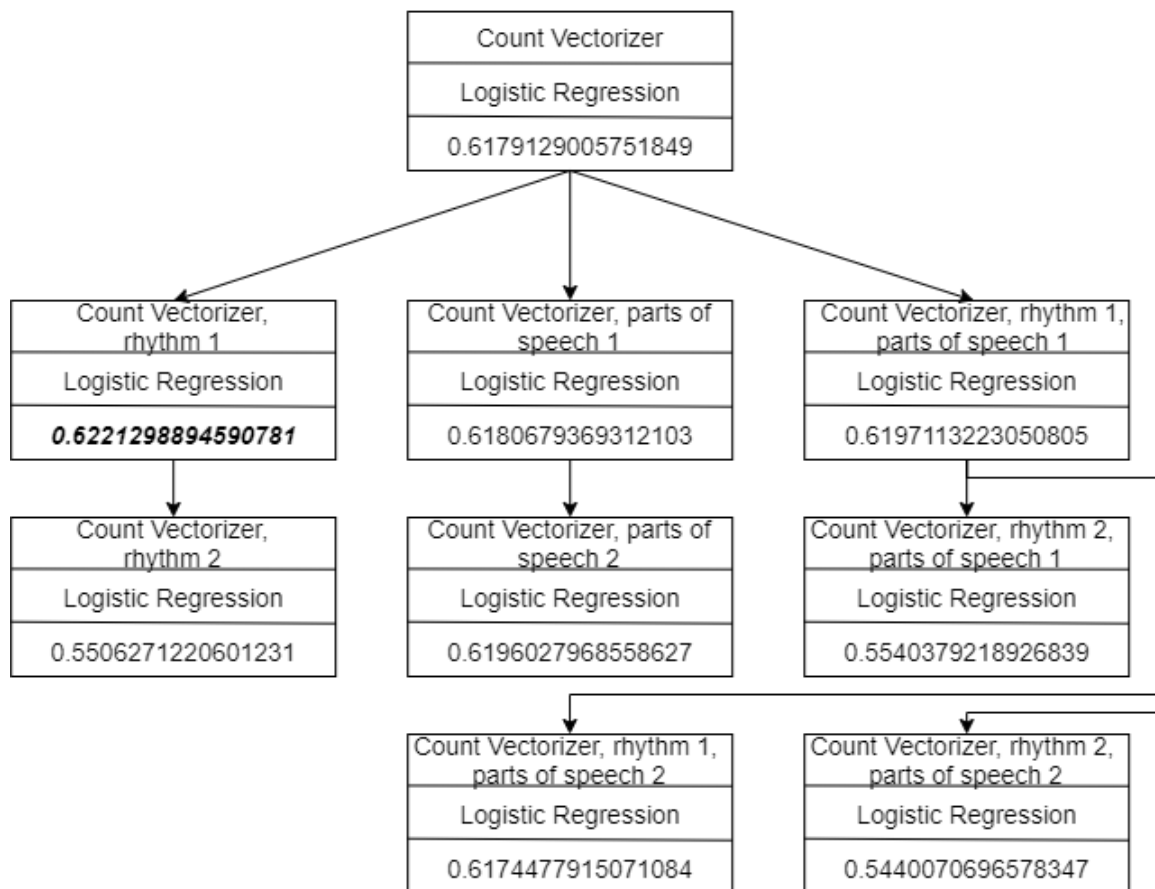


Fig. 6. Structural and logical scheme, p.4

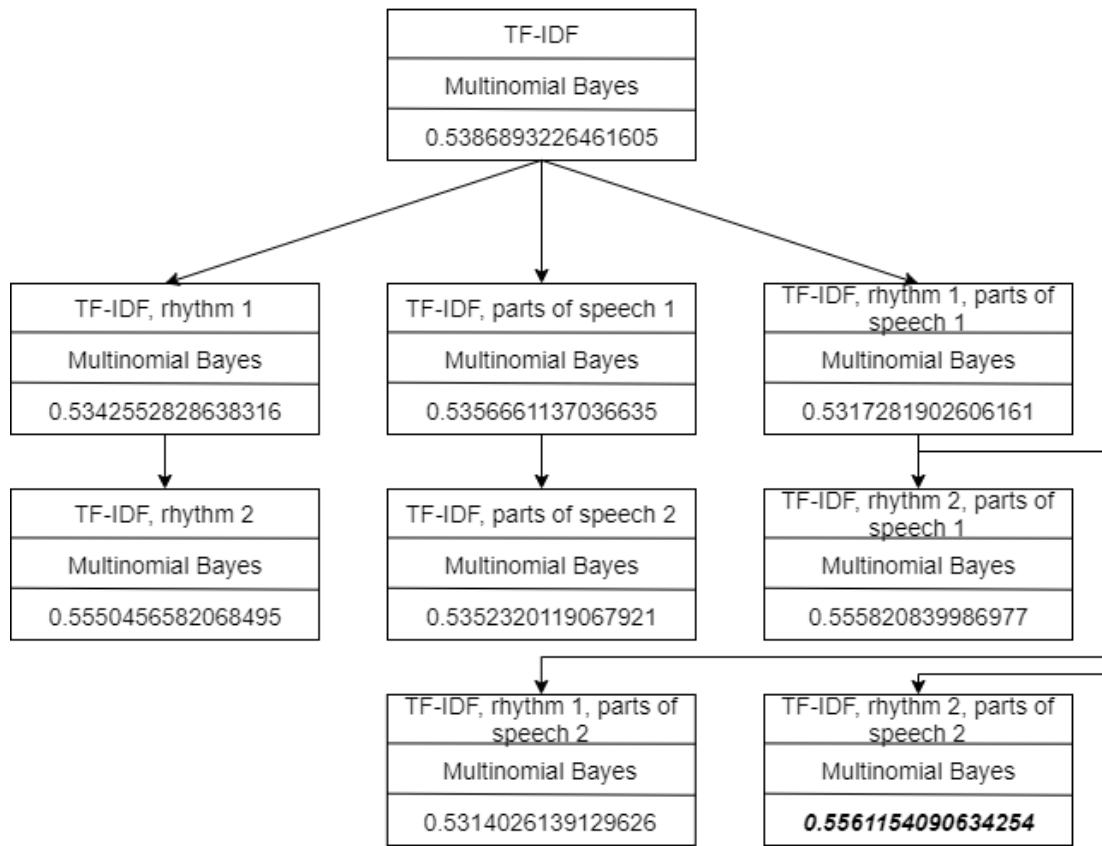


Fig. 7. Structural and logical scheme, p.5

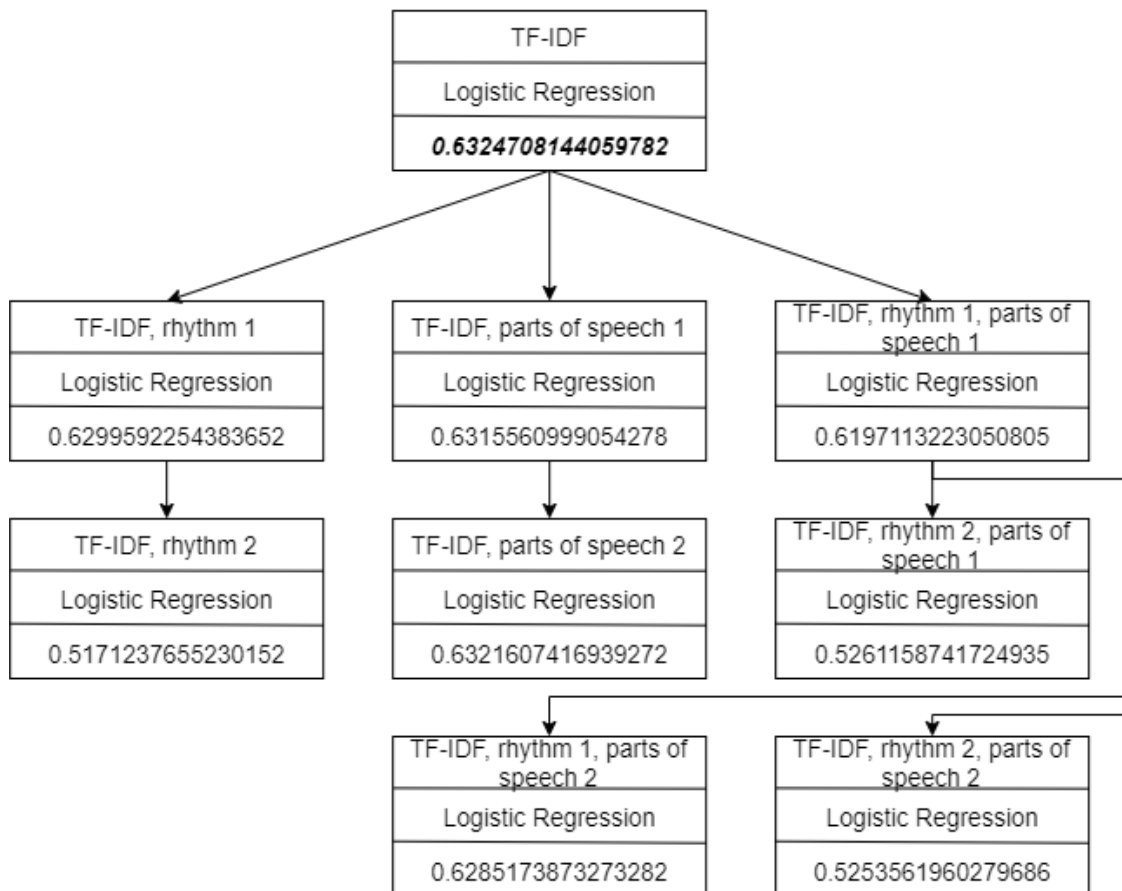


Fig. 8. Structural and logical scheme, p.

Table 1
Evaluation of the research results

№	Name	Labor units (points 0 - 9)	Depth of Innovation (points 0 - 9)	Result	Influ ence
Before					
1.0	BoW ^a	3	0	0.6179	
Proposed					
1.1	R1 ^b	7	9	0.6221	+
1.2	PS1 ^c	5	5	0.6180	+
1.3	R2 ^d	8	8	0.5506	-
1.4	PS2 ^e	5	5	0.6196	+
1.5	R1,PS1	5	6	0.6197	+
1.6	R1,PS2	5	6	0.6174	-
1.7	R2,PS1	5	6	0.5540	-
1.8	R2,PS2	5	6	0.5440	-
Before					
2.0	TF-IDF	3	0	0.6324	
Proposed					
2.1	R1	7	9	0.6299	-
2.2	PS1	5	5	0.6315	-
2.3	R2	8	8	0.5171	-
2.4	PS2	5	5	0.6321	-
2.5	R1,PS1	5	6	0.5261	-
2.6	R1,PS2	5	6	0.6285	-
2.7	R2,PS1	5	6	0.5261	-
2.8	R2,PS2	5	6	0.5253	-

^aBoW – Bag of Words, ^bR1 – Rhythm 1, ^cPS1 – Parts of speech 1, ^dR2 – Rhythm 2, ^ePS2 – Parts of speech 2

Thus, not only the approaches were found to improve the accuracy of defining song genres. Also, the difference in the effectiveness of solution when changing methods was tracked. This allowed us to build a trajectory in space of possible solutions, which led to the best solution. In addition, this trajectory can be improved according to the change in the conditions of the task, in order to obtain the best method for new conditions. This result lies in the plane of management of purposeful receipt of new ideas and can be considered an extension of existing approaches to the theory of invention, for example, a morphological table of possible solutions.

Since the preliminary results were calculated for the same test data, 10 experiments were conducted to calculate the mathematical expectation, using logistic regression for the features that give the best result, namely: TF-IDF; TF-IDF, rhythm 1; TF-IDF, rhythm 1, parts of speech 1. The results of experiments are shown in the graph (Fig. 9):

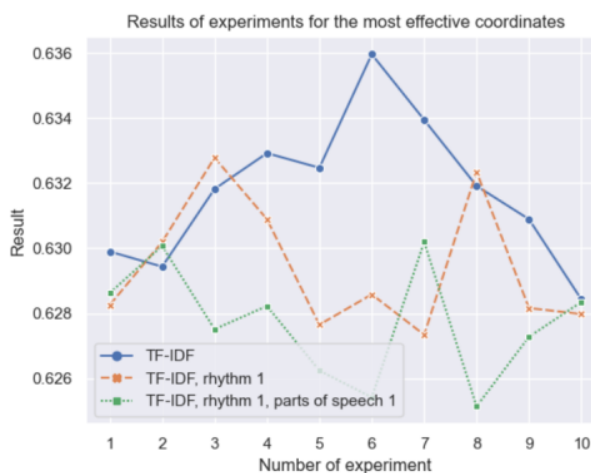


Fig. 9. Results of experiments for the most effective features

To verify the distribution of results, Q-Q (quantile-quantile) graphs were built and Shapiro-Wilk tests were conducted. This showed that the resulting distribution is slightly different from the normal. Based on these results and considering the possible error associated with a small data sample, we concluded

Table 2
Result table

	TF-IDF	TF-IDF, Rhythm 1	TF-IDF, rhythm 1, parts of speech 1
Average	0.6318	0.6294	0.6277
Median	0.6319	0.6284	0.6279
Maximum	0.6359	0.6328	0.6302
Minimum	0.6284	0.6273	0.6251
Standard Deviation	0.0021	0.0019	0.0016
Dispersion	0.0000045	0.0000036	0.0000027

that the results are distributed normally. Thus, the results can be represented by the following table (Table 2):

Therefore, our features in some cases add up to 2% of accuracy to the main method. Therefore, even though the highest accuracy was obtained without the use of new features, further studies of rhythm have great potential.

Conclusions

1. As a result of the work, its purpose was achieved, namely, the accuracy of determining the genre of the song by its lyrics was increased through the development of theoretical approaches and corresponding software for mobile and embedded digital systems that use new factors, namely the rhythm of the text and its morpho-syntactic structure.

2. Scientific novelty is the use of a new method, namely the determining of the text rhythm and its parts of speech, to classify songs by genres by creating new features. The best results were obtained due to the feature TF-IDF (Term Frequency - Inversed Document Frequency) and its combinations with features to indicate rhythm and rhythm with parts of speech. During the research, we realized that the rhythm potential is much larger than the scale of our project, since the presentation of text in the form of a rhythm allows you to binarize any text-like information.

3. The study progressed according to the principles of invention theory, which is reflected in the structural and logical scheme of research, which was built to analyze the results. This made it possible to see that not only approaches were found to improve the accuracy of defining song genres, but also the effec-

tiveness of the solution when changing methods was tracked. Thus, a trajectory was built in the space of possible solutions, which led to the best solution.

4. The created program code was successfully approbated by placing it on a web service for joint software development GitHub [8].

5. The practical application of this innovation is not limited to the obvious musical application, namely the improvement of music recommendation algorithms or assistance for young authors. More generally, the analysis of rhythm will allow to find non-obvious patterns in such texts as:

- political speeches: to edit speech text to achieve the best perception by the audience;
- historical documents: to analyze their authenticity or belonging to a particular historical period;
- promotional texts: to edit the text for the best targeting;
- songs: to find musical plagiarism;
- dialogues from movies: to script features that make movies popular.

References

1. Greer, T. and Narayanan, S., 2019. Using Shared Vector Representations of Words and Chords in Music for Genre Classification. SMM19, Workshop on Speech, Music and Mind 2019, [online] pp.46-49. Available at: <https://www.isca-speech.org/archive/SMM_2019/pdfs/SMM19_paper_19.pdf>.
2. Sadovsky, A. and Chen, X., 2006. Song Genre and Artist Classification via Supervised Learning from Lyrics. [online] pp.1-18. Available at: <<https://nlp.stanford.edu/courses/>

- cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf>.
3. Brennan, C., Paul, S., Yalamanchili, H. and Yum, J., 2018. Classifying Song Genres Using Raw Lyric Data with Deep Learning. [online] GitHub. Available at: <<https://github.com/hiteshyalamanchili/SongGenreClassification>>.
 4. Speech.cs.cmu.edu. 2021. The CMU Pronouncing Dictionary. [online] Available at: <<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>>.
 5. Brownlee, J., 2021. Logistic Regression Tutorial for Machine Learning. [online] Machine Learning Mastery. Available at: <<https://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning/>>.
 6. Shevchenko, V., 2011. Optimization Modeling in Strategic Planning. CVSD NUOU, pp. 283.
 7. Brownlee, J., 2019. Logistic Regression Tutorial for Machine Learning. [online] Machine Learning Mastery. Available at: <<https://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning/>>.
 8. Triantafillu A., 2021. Song Genre Predictor GitHub Project. [online] Available at: <<https://github.com/triantafillu/SongGenrePredictor>>.

Received: 18.05.2021

About the authors:

Aleksandra A. Triantafillu

Bachelor student of Taras Shevchenko National University of Kyiv
Publications: 2
29 Lobanovskoho ave., ap.10, 03037, Kyiv, Ukraine
<https://orcid.org/0000-0003-2595-8699>

Mykola A. Mateshko

Bachelor student of Taras Shevchenko National University of Kyiv
Publications: 2
52-V Evhena Svertyuka str., ap. 42, 02002, Kyiv, Ukraine
<https://orcid.org/0000-0002-6203-0577>

Viktor L. Shevchenko

Dr.Sc., Prof., Professor at Software systems and technologies Department of Taras Shevchenko National University of Kyiv.
Shovkunenka st. 3-72, 03049, Kyiv, Ukraine.
Publications: more than 300
Foreign scientometric publications: 17
H=3
<https://orcid.org/0000-0002-9457-7454>

Igor P. Sinitsyn

Dr.Sc., Senior Researcher,
Head of Department at the Institute of Software Systems of the National Academy of Sciences of Ukraine.
Publications: more than 300
Foreign scientometric publications: 3
H=1
<https://orcid.org/0000-0002-4120-0784>

Affiliations:

Taras Shevchenko National University of Kyiv
Volodymyrska St, 60, Kyiv, 01033
Phone: +380442393333
Fax: +380442393388
E-mail: office.chief@univ.net.ua
E-mail: alexandra00219@gmail.com
nmateshko@gmail.com
gii2014@ukr.net

Institute of Software Systems of the National Academy of Sciences of Ukraine.
03187, Kyiv-187, Acad. Hlushkov avenue,
Phone: +380445263319
E-mail: iss@isofts.kiev.ua

PROBLEMS OF PROGRAMMING

2021 – №2

UDC 004.822:519.15

Development of domain thesaurus as a set of ontology concepts with use of semantic similarity and elements of combinatorial optimization / A.Ya. Gladun, J.V. Rogushina

We consider use of ontological background knowledge in intelligent information systems and analyze directions of their reduction in compliance with specifics of particular user task. Such reduction is aimed at simplification of knowledge processing without loss of significant information. We propose methods of generation of task thesauri based on domain ontology that contain such subset of ontological concepts and relations that can be used in task solving. Combinatorial optimization is used for minimization of task thesaurus. In this approach, semantic similarity estimates are used for determination of concept significance for user task. Some practical examples of optimized thesauri application for semantic retrieval and competence analysis demonstrate efficiency of proposed approach.

Keywords: domain ontology, task thesaurus, semantic similarity, combinatorial optimization

UDC 004.02+004.05+005.93+ 510.3

Security basic model for applied tasks of the distributed information system / Y.S. Rodin, I.P.Sinitsyn.

The tasks of modelling and the components of the basic model of applied task protection of a distributed information system have been considered. The measurement and relationship of security parameters, protection, new and reference attacks, anomalies, and threat environments have been proposed. The conditions of threats, attacks and, consequently, inconsistencies in the results of applied tasks are proved. At the beginning of

УДК 004.822:519.15

Розробка тезаурусу домену як сукупності концепцій онтології з використанням семантичної подібності та елементів комбінаторної оптимізації / А.Я. Гладун, Ю.В. Рогушина

Автори розглядають використання онтологічних фонових знань в інтелектуальних інформаційних системах та аналізують методи їх зменшення відповідно до потреб та особливостей конкретної задачі користувача. Таке зменшення спрямоване на спрощення обробки знань без значних втрат у кількості оброблюваної інформації. Було запропоновано методи генерування тезаурусу задачі на основі онтології домену, що містить підмножину онтологічних понять та відношень, які можуть бути використані для вирішення задачі. Для мінімізації тезаурусу задачі використано методи комбінаторної оптимізації. Показники оцінки семантичної подібності у цьому підході було використано для визначення значущості концепту для задачі користувача. Ефективність запропонованого підходу було продемонстровано на деяких практичних прикладах оптимізованого застосування тезаурусу для семантичного пошуку та аналізу компетенцій.

Ключові слова: онтологія домену, тезаурус задачі, семантична подібність, комбінаторна оптимізація.

УДК 004.02+004.05+005.93+ 510.3

Базова модель захисту прикладних задач розподіленої інформаційної системи / Є. Родін, І.Сініцин.

Розглянуто задачі моделювання та складові базової моделі захисту прикладних задач розподіленої інформаційної системи. Запропоновано вимір та зв'язок параметрів безпеки, захищеності, нових та еталонних атак, аномалій, середовищ функціонування загроз. Запропоновано використання експертної семантичної системи для розширення бази знань новими загрозами, атаками і, засобами про-

the article the concept of a distributed information system, system of applied tasks, modern trends of zero-trust architecture in building information security systems are discussed. Further, it gives an overview of existing methods of detection and counteraction to attacks based on reference knowledge bases. To improve the level of security it is proposed to analyze the causes of attacks, namely hazards and threats to the system. Attacks, hazards and threats are considered as structured processes that affect the internal and external environment of the system of the applied tasks with a further impact on the output of these tasks. The concepts of security level and security level of a distributed information system are introduced, as well as the concepts of applied task, environment, and user contradictions. As the logical metrics of discrepancy detection the apparatus of semantic analysis is proposed, which (based on the reference knowledge base, the apparatus of text transformations) should be applied at the stage of loading of applied task and describe the input and output data, requirements to the environment of the task solution.

The result of the research is the proposed method for identifying additional data about hazards, threats, attacks, countermeasures to attacks, applied task-solving. This data is generated from the reference and augmented textual descriptions derived from the proposed contradictions. By building additional reference images of threats, attacks, countermeasures, it becomes possible to prevent the activation of new attacks on the distributed information system.

Keywords: information, security, anomaly, attack, model, application problem, distributed system, semantics.

тидії. На початку статті йдеться про поняття розподіленої інформаційної системи, системи прикладних задач, сучасні тенденції архітектури нульової довіри при побудові систем інформаційного захисту. Далі пропонується огляд існуючих методів виявлення та протидії атакам на базі еталонних баз знань. Для підвищення рівня безпеки пропонується аналіз причин виникнення атак, а саме, небезпек та загроз системи за допомогою розширення бази знань семантичними інтерпретаціями й суперечностями між ними.

Атаки, небезпеки та загрози розглядаються як структуровані процеси, що впливають на внутрішнє та зовнішнє середовище функціонування системи прикладних задач з подальшим впливом на вихідні дані цих задач. Вводяться поняття рівня безпеки і рівня захищеності розподіленої інформаційної системи, а також поняття суперечностей прикладних задач, середовища, користувача. Як логічні метрики виявлення суперечностей пропонується апарат семантичного аналізу, що на основі еталонної бази знань, апарату текстових перетворень має бути застосований на етапі завдання прикладної задачі й описувати вхідні та вихідні дані, вимоги до середовища вирішення цієї задачі. Також апарат семантичного аналізу пропонується використовувати для аналізу записів інцидентів, протоколів протидій, наслідків.

Результатом дослідження є запропонований метод виявлення додаткових даних про небезпеки, загрози, атаки, засоби протидії атакам розв'язання прикладних задач. Ці дані формується з еталонних та розширених текстових описів, отриманих на базі запропонованих суперечностей. Завдяки нарощуванню додаткових еталонних образів загроз, атак, засобів протидії стає можливим запобігти активізації нових атак на розподілену інформаційну систему.

Ключові слова: інформація, безпека, аномалія, атака, модель, прикладна задача, розподілена система, семантика.

UDC 004.94

УДК 004.94

Defining degree of semantic similarity using description logic tools / O. Zakharova

Визначення ступеня семантичної подібності з використанням апарату дескриптивних логік / О. В. Захарова

Establishing the semantic similarity of information is an integral part of the process of solving any information retrieval tasks, including tasks

Встановлення семантичної подібності інформації є невід'ємною складовою процесу вирі-

related to big data processing, discovery of semantic web services, categorization and classification of information, etc. The special functions to determine quantitative indicators of degree of semantic similarity of the information allow ranking the found information on its semantic proximity to the purpose or search request/template. Forming such measures should take into account many aspects from the meanings of the matched concepts to the specifics of the business-task in which it is done. Usually, to construct such similarity functions, semantic approaches are combined with structural ones, which provide syntactic comparison of concepts descriptions. This allows to do descriptions of the concepts more detail, and the impact of syntactic matching can be significantly reduced by using more expressive descriptive logics to represent information and by moving the focus to semantic properties. Today, DL-ontologies are the most developed tools for representing semantics, and the mechanisms of reasoning of descriptive logics (DL) provide the possibility of logical inference. Most of the estimates presented in this paper are based on basic DLs that support only the intersection constructor, but the described approaches can be applied to any DL that provides basic reasoning services.

This article contains the analysis of existing approaches, models and measures based on descriptive logics. Classification of the estimation methods both on the levels of defining similarity and the matching types is proposed. The main attention is paid to establishing the similarity between concepts (conceptual level models). The task of establishing the value of similarity between instances and between concept and instance consists of finding the most specific concept for the instance / instances and evaluating the similarity between the concepts. The term of existential similarity is introduced. In this paper the examples of applying certain types of measures to evaluate the degree of semantic similarity of notions and/or knowledge based on the geometry ontology is demonstrated.

Key words: semantic similarity of information, a value of similarity of concepts, least concept subsumer, measures for similarity evaluating, most specific concept, most specific is-a ancestor, similarity function, similarity measure information content, features-based similarity measure, measure of distance between concepts, features-based models, semantic-network based models, information content based models, existential concepts similarity, similarity between two individuals, similarity between concept and individual, similarity between DL-descriptions of concepts, GCS-similarity.

шення будь-яких задач інформаційного пошуку, в тому числі задач, що пов'язані з обробкою великих даних, виявленням семантичних веб-сервісів, категоризації та класифікації інформації тощо. Введення спеціальних функцій для визначення кількісних показників ступеня семантичної відповідності інформації дозволяють ранжувати знайдену інформацію за її семантичною близькістю до цілі або пошукового запиту/шаблону. Формування таких оцінок повинно враховувати багато аспектів від сутності самих понять, що оцінюються, до особливостей бізнес-задачі, в межах вирішення якої це робиться. Зазвичай, при побудові функцій подібності семантичні підходи поєднуються зі структурними, що забезпечують синтаксичне порівняння описів концептів. Це дозволяє деталізувати опис концепта, а вплив синтаксичної відповідності можна значно зменшити, використовуючи для представлення інформації більш виразні дескриптивні логіки (ДЛ) та шляхом перенесення фокусу на семантичні властивості. ДЛ-онтології, на сьогодні, є найбільш розвиненим засобом представлення семантики, а механізми міркувань ДЛ забезпечують можливість логічного виводу. Більшість наведених у роботі оцінок будуються на основі базових ДЛ, що підтримують лише конструктор перетину, але описані підходи можуть бути застосовані для будь-якої ДЛ, що забезпечує базові сервіси міркувань.

В роботі проведений аналіз існуючих підходів, моделей та мір оцінювання, що засновані на застосуванні апарату ДЛ, запропонована їх класифікація як за рівнем визначення подібності, так й за видами співставлення. Головна увага приділяється встановленню подібності концептів. Задачі встановлення подібності між екземплярами/концептом та екземпляром зводяться до знаходження найбільш специфічного концепта для екземпляра/екземплярів та оцінювання подібності відповідних концептів. Введено поняття екзистенціональної подібності та продемонстровано застосування певних видів оцінок для визначення ступеня подібності понять/знань на прикладі онтології геометричних понять.

Ключові слова: семантична подібність інформації, найменше спільне покриття, оцінки вимірювання подібності, найбільш специфічний концепт, найбільш специфічний попередник, функція подібності, подібність за інформаційним змістом, семантична подібність за відповідністю ознак, функція відстані шляху, моделі оцінювання на основі властивостей, моделі оцінювання на основі семантичної мережі, моделі оцінювання на основі інформаційного контенту, екзистенціональна подібність концептів, подібність екземплярів, подібність концепта та екземпляра, подібність ДЛ описів, GCS-подібність.

Ontology-based semantic similarity to metadata analysis in the information security domain /

A.Ya. Gladun, K.A. Khala

It is becoming clear with growing complication of cybersecurity threats, that one of the most important resources to combat cyberattacks is the processing of large amounts of data in the cyber environment. In order to process a huge amount of data and to make decisions, there is a need to automate the tasks of searching, selecting and interpreting Big Data to solve operational information security problems. Big data analytics is complemented by semantic technology, can improve cybersecurity, and allows you to process and interpret large amounts of information in the cyber environment. Using of semantic modeling methods in Big Data analytics is necessary for the selection and combination of heterogeneous Big Data sources, recognition of the patterns of network attacks and other cyber threats, which must occur quickly to implement countermeasures. Therefore to analyze Big Data metadata, the authors propose pre-processing of metadata at the semantic level. As analysis tools, it is proposed to create a thesaurus of the problem based on the domain ontology, which should provide a terminological basis for the integration of ontologies of different levels. To build a thesaurus of the problem, it is proposed to use the standards of open information resources, dictionaries, encyclopedias. The development of an ontology hierarchy formalizes the relationships between data elements that will be used in future for machine learning and artificial intelligence algorithms to adapt to changes in the environment, which in turn will increase the efficiency of big data analytics for the cybersecurity domain.

Keywords: big data analytics, information security, cyber security, ontology, thesaurus, unstructured data, metadata, semantic similarity.

UDC 517.958:57 +519.711.3 + 612.51.001

Specialized software for simulating the multiple control and modulations of human hemodynamics /

Grygoryan R.D., Yurchak O.I., Degoda A.G., Lyudovyk T.V.

Most models of human hemodynamics describe only a small part of physiological mechanisms that directly or indirectly alter activities of the heart pump and vascular tones. Therefore, a very nar-

Онтологічний підхід до аналізу метаданих в домені інформаційної безпеки /

А.Я. Гладун, К.О. Хала

Із зростанням і частим ускладненням загроз кібербезпеки, стає очевидним, що одним із найважливіших ресурсів для боротьби з кібератаками є оброблення великого обсягу даних у кіберсередовищі. Для оброблення величезної кількості даних та для прийняття рішень постає потреба у автоматизації задач пошуку, відбору та інтерпретації Великих Даних для вирішення оперативних задач інформаційної безпеки. Однак традиційні технології аналітики Великих Даних мають обмежені можливості і потребують нового підходу – застосування знань для керування життєвим циклом Великих Даних. Аналітика Великих Даних доповнена семантичними технологіями, може покращити кіберзахист, та дозволяє обробляти і інтерпретувати великі обсяги інформації в кіберсередовищі. Для аналізу метаданих Великих Даних автори пропонують попередню обробку метаданих на рівні семантики. Детальний опис знань про домен інформаційної безпеки має ієрархічну структуру, яка складається з декількох рівнів. Для побудови тезаурусу задачі запропоновано використати стандарти відкритих інформаційних ресурсів, словники, енциклопедії. Розробка ієрархії онтологій формалізує взаємозв'язки між елементами даних, які в майбутньому будуть використані для машинного навчання та алгоритмів штучного інтелекту для адаптації до змін у середовищі, що у свою чергу підвищить ефективність аналітики великих даних для домену кібербезпеки.

Ключові слова: аналітика великих даних, інформаційна безпека, кібербезпека, онтологія, тезаурус, неструктуровані дані, метадані.

УДК 517.958:57 +519.711.3 + 612.51.001

Спеціалізоване програмне забезпечення для моделювання множинного керування та модуляцій гемодинаміки людини /

Григорян Р.Д., Юрчак О.І., Дегода А.Г., Людовик Т.В.

Більшість моделей гемодинаміки людини описують лише незначну частину фізіологічних механізмів, які прямо чи опосередковано змі-

row range of tasks related to cardiovascular physiology can be solved using these models. To essentially widen this range, special software based on quantitative models of mechanisms providing the overall control of circulation is created. In the complex model, a multi-compartmental lumped parametric model of hemodynamics, provided under stable values of blood volume and cardiovascular parameters, forms the core model. It consists of two ventricles and 21 vascular compartments. Additional dynamic models represent mechanisms of mechanoreceptor reflexes, chemoreceptor reflexes, main effects of angiotensin-II, antidiuretic hormone, vasopressin, adrenalin, and cardiac or brain ischemia. The software has a physiologist-oriented user interface. It provides the investigator with multiple capabilities for simulating different states of each included mechanism. The interface also allows creating arbitrary combinations of the chosen mechanisms. In particular, the chosen model of these mechanisms is activated or deactivated via the user interface. The activated model modulates initial values of the core model. Special opportunities have been created for simulating different hypotheses concerning the etiology of arterial hypertension. Simulation results are presented with graphs. The user interface documents each simulation as a special file that can be saved for later independent analysis. The software, created in the frame of .NET technology, is an autonomous .EXE file for executing on PC. Software is also a good computer program to be used for educational purposes for illustrating the main physiological and certain pathological regularities to medical students.

Key words: physiology, cardiovascular system, acute and long-term control, model, simulator.

нюють діяльність серцевого насоса та судинний тонус. Отже, за допомогою цих моделей можна виконати лише дуже вузький діапазон завдань, пов'язаних із серцево-судинною фізіологією. Щоб істотно розширити цей діапазон, створено спеціальне програмне забезпечення, засноване на кількісних моделях механізмів, що забезпечують загальний контроль кровообігу. У комплексній моделі основна модель формує багатокамерну модель гемодинаміки, яка забезпечується при стабільних значеннях об'єму крові та серцево-судинних параметрів. Ця модель складається з двох шлуночків та 21 судинного відділу. Додаткові динамічні моделі представляють механізми механорецепторних рефлексів, хеморецепторних рефлексів, основних ефектів ангіотензину-II, антидіуретичного гормону, вазопресину, адреналіну та ішемії серця або мозку. Програмне забезпечення орієнтоване на фізіологів та має користувальницький інтерфейс. Це надає досліднику безліч можливостей для моделювання різних станів кожного включеного механізму. Інтерфейс також дозволяє створювати довірливі комбінації обраних механізмів. Зокрема, обрана модель цих механізмів активується або деактивується через користувальницький інтерфейс. Активована модель модулює початкові значення базової моделі. Спеціальні можливості створені для моделювання різних гіпотез, що стосуються етіології артеріальної гіпертензії. Результати моделювання представлені графіками. Користувальницький інтерфейс документує кожне моделювання у вигляді спеціального файла, який можна зберегти для подальшого незалежного аналізу. Програмне забезпечення, створене в рамках технології .NET, є автономним файлом .EXE для запуску на персональному комп'ютері. Розроблене програмне забезпечення є також хорошим засобом для ілюстрування студентам-медикам основних фізіологічних та певних патологічних закономірностей.

Ключові слова: фізіологія, серцево-судинна система, гострий та тривалий контроль, модель, тренажер.

UDC 681.3

Extended performance accounting using Valgrind tool / D.V. Ragoza, A. Yu. Doroshenko – P.

Modern workloads, parallel or sequential, usually suffer from insufficient memory and computing performance. Common trends to improve

УДК 681.3

Розширений аналіз швидкодії програм за допомогою Valgrind / Д.В. Рагозін, А. Ю. Дорошенко - С.

Сучасні паралельні або послідовні програми-навантаження (workloads) звичайно мають обмеження за швидкістю процесора або за

workload performance include the utilizations of complex functional units or coprocessors, which are able not only to provide accelerated computations but also independently fetch data from memory generating complex address patterns, with or without support of control flow operations. Such coprocessors usually are not adopted by optimizing compilers and should be utilized by special application interfaces by hand. On the other hand, memory bottlenecks may be avoided with proper use of processor prefetch capabilities which load necessary data ahead of actual utilization time, and the prefetch is also adopted only for simple cases making programmers to do it usually by hand. As workloads are fast migrating to embedded applications a problem raises how to utilize all hardware capabilities for speeding up workload at moderate efforts. This requires precise analysis of memory access patterns at program run time and marking hot spots where the vast amount of memory accesses is issued. Precise memory access model can be analyzed via simulators, for example Valgrind, which is capable to run really big workload, for example neural network inference in reasonable time. But simulators and hardware performance analyzers fail to separate the full amount of memory references and cache misses per particular modules as it requires the analysis of program call graph. We are extending Valgrind tool cache simulator, which allows to account memory accesses per software modules and render realistic distribution of hot spot in a program. Additionally the analysis of address sequences in the simulator allows to recover array access patterns and propose effective prefetching schemes. Motivating samples are provided to illustrate the use of Valgrind tool.

Keywords: workload, performance analysis, coprocessors, prefetch, computer system simulator.

потужністю каналів пам'яті. Також сучасною тенденцією є залучення спеціалізованих сопроцесорів для підвищення швидкодії програм-навантажень, які виконують не тільки обчислення, але й доступ до пам'яті зі складною адресацією. Такі сопроцесори практично неможливо використати за допомогою компілятора, лише ручним кодуванням програми. Обмеження за потужністю каналу пам'яті також може вирішуватися складною системою передвибірки даних з пам'яті у кеш-пам'ять процесора, але компілятор теж може оптимізувати передвибірку лише у простих випадках побудови коду. Оскільки програми-навантаження дуже швидко мігрують у бік вбудованих обчислень, виникає проблема спрощення використання вбудованих сопроцесорів для підвищення швидкодії. Це потребує аналізу послідовностей доступу до пам'яті та визначення вузьких місць у коді програми. Точний аналіз доступу можливий за допомогою симуляторів, наприклад Valgrind, який дозволяє аналізувати великі програми-навантаження, наприклад, вивід у нейромережах і за адекватний час. Наявні симулятори та засоби аналізу навантаження процесора не дозволяють коректно визначати навантаження у прив'язці до програмних компонентів, оскільки це потребує аналізу графу викликів у програмі. Тому ми розширюємо симулятор Valgrind можливостями аналізу прив'язки доступу до пам'яті до конкретних програмних модулів і визначенням уточнених вузьких місць доступу до пам'яті. Додатково аналіз послідовності адрес доступу до пам'яті дозволяє визначити шаблони доступу до масивів і рекомендувати використання певних алгоритмів передвибірки даних до кеш-пам'яті. Додаються ілюстративні приклади використання симулятора Valgrind.

Ключові слова: програма-навантаження, аналіз швидкодії, сопроцесор, передвибірка даних з пам'яті, симулятор комп'ютерної системи.

UDC 517.9:621.325.5:621.382.049.77

УДК 517.9:621.325.5:621.382.049.77

Specific features of the use of artificial intelligence in the development of the architecture of intelligent fault-tolerant radar systems / M. Kosovets, L. Tovstenko

Особенности использования искусственного интеллекта при разработке архитектуры интеллектуальных устойчивых радиолокационных систем / М. Косовец, Л. Товстенко

The problem of architecture development of modern radar systems using artificial intelligence technology is considered. The main difference

Розглянуто проблему розробки архітектури сучасних когнітивних радіолокаційних сис-

is the use of a neural network in the form of a set of heterogeneous neuromultimicroprocessor modules, which are rebuilt in the process of solving the problem systematically in real time by the means of the operating system. This architecture promotes the implementation of cognitive technologies that take into account the requirements for the purpose, the influence of external and internal factors. The concept of resource in general and abstract resource of reliability in particular and its role in designing a neuromultimicroprocessor with fault tolerance properties is introduced. The variation of the ratio of performance and reliability of a fault-tolerant neuromultimicroprocessor of real time with a shortage of reliability resources at the system level by means of the operating system is shown, dynamically changing the architectural appearance of the system with structural redundancy, using fault-tolerant technologies and dependable computing.

Keywords: neuromultimicroprocessor, probability of trouble-free operation, initialization, resource, interface, modularity, supervisor, multiprogramming, reconfiguration system, access method.

UDC 004.04:004.942

Intonation expressiveness of the text at program sounding / V. L. Shevchenko, Y. S. Lazorenko, O. M. Borovska

As the amount of media content increases, there is a need for its automated sounding with the most accessible built-in and mobile means. The factors influencing the formation of different intonations were analyzed in the article, the dependences of the change of sound characteristics in accordance with the intonations were mathematically described. In the course of the study, the numerical analysis of sentences was improved using the moving average method for smoothing audio recording, approximation lines for approximate generalization of emotions as mathematical functions, and Fourier transform for volume control. The obtained dependences allow to synthesize the necessary intonations according to the punctuation of the sentence, the presence of emotionally colored vocabulary and psycho-emotional mood of the speaker when reading such a text.

тем у вигляді набору гетерогенних нейромульти-мікропроцесорних модулів з використанням технологій штучного інтелекту та урахуванням вимог по призначенню, впливу зовнішніх та внутрішніх факторів. Оптимальний вибір архітектури забезпечується її максимальним наближенням до класу задач, що вирішуються в радіолокації і представлені у вигляді послідовності багатовимірних масивів чисел, що постійно змінюються з часом від потокових датчиків. Введено поняття абстрактного ресурсу надійності, та його роль у проектуванні нейромультимікропроцесора з властивостями відмовостійкості. Показано залежність надійності від співвідношення продуктивності та надійності нейронної мережі при дефіциті ресурсу надійності, яка перебуває в процесі вирішення задач радіолокації в режимі реального часу на системному рівні засобами операційної системи, яка динамічно змінює архітектурний облік системи зі структурною надмірністю, відмовостійкістю та гарантоздатними обчисленнями в реальному масштабі часу.

Ключові слова: нейромультимікропроцесор, ймовірність безперебійної роботи, ініціалізація, ресурс, інтерфейс, модульність, супервізор, мультипрограмування, система реконфігурації, метод доступу.

УДК 004.04:004.942

Інтонаційна виразність тексту при програмному озвучуванні / В. Л. Шевченко, Я. С. Лазоренко, О. М. Боровська

Із збільшенням обсягів медіа-контенту виникає потреба в його автоматизованій обробці, зокрема озвучуванні, за допомогою найбільш доступних вбудованих та мобільних засобів. Тому було проаналізовано фактори, що впливають на формування різних інтонацій, математично описано залежності зміни звукових характеристик відповідно до інтонацій. У ході роботи чисельний аналіз речень було удосконалено за допомогою методу ковзного середнього для згладжування аудіо запису, ліній апроксимації для наближеного узагальнення емоцій як математичних функцій та перетворення Фур'є для регулювання висоти звуку. Отримані залежності дозволяють синтезувати потрібні інтонації відповідно до пунктуації речення, наявності в ньому емоційно забарвленої лексики та психоемоційного настрою

As a result of our study, software for emotional sounding of texts was developed, which provides the perception of audio information easier, clearer and more comfortable based on the use of built-in processors of mobile devices.

Key words: text analysis, sound characteristics, intonation expressiveness.

UDC 004.04:004.942

Algorithm and software for determining a musical genre by lyrics to create a song hit / A.A. Triantafillu, M.A. Mateshko, V.L. Shevchenko, I.P. Sinitsyn

One of the needs of music business is a quick classification of the song genre by means of widely available tools (such as built-in smartphone processors). This work focuses on improving the accuracy of the song genre determination based on its lyrics through the development of software that uses new factors, namely the rhythm of the text and its morpho-syntactic structure. In the research Bayes Classifier and Logistic Regression were used to classify song genres, a systematic approach and principles of invention theory were used to summarize and analyze the results. Programs were written on Python programming language. New features were proposed in the work to improve the accuracy of the classification, namely the features to indicate rhythm and parts of speech in the song.

Keywords: genre, rhythm, song, text, classification.

мовця при прочитанні подібного тексту. В результаті виконання роботи було розроблено програмне забезпечення для емоційного озвучування текстів, яке робить сприйняття аудіо-інформації легшим, зрозумілішим і більш комфортним, на основі використання вбудованих процесорів мобільних пристроїв.

Ключові слова: аналіз тексту, звукові характеристики, інтонаційна виразність.

УДК 004.04:004.942

Алгоритм та програмне забезпечення для визначення жанру пісні задля створення музичного хіта / А. А. Тріантафіллу, М. А. Матешко, В. Л. Шевченко, І. П. Сініцин

В ХХІ столітті музика – це дуже прибутковий бізнес як для стримінгових сервісів, що пропонують музику користувачу, так і для авторів пісень, що намагаються продати свої тексти. Однією з потреб цього бізнесу є визначення жанру майбутньої або вже існуючої пісні, щоб вигідно продати її замовнику, або запропонувати зацікавленому користувачу. Практика вимагає все більшої і більшої точності, але існуючі практичні підходи не спроможні її надати, оскільки методи класифікації пісень за жанром недостатньо розвинуті на теоретичному рівні. Ця робота зосереджена на підвищенні точності визначення жанру пісні за її текстом шляхом розробки програмного забезпечення, що використовує нові фактори, а саме ритм тексту та його морфо-синтаксичну структуру. У дослідженні використовувалися класифікатор Байеса та логістична регресія для класифікації жанрів пісень, систематичний підхід та принципи теорії винахідництва для узагальнення та аналізу результатів. Програми були написані на мові програмування Python. У роботі було запропоновано нові метрики для підвищення точності класифікації, а саме метрики на позначення ритму тексту та кількості різних частин мови в пісні.

Ключові слова: жанр, ритм, пісня, текст, класифікація.

ДО УВАГИ АВТОРІВ!

У журналі «Проблеми програмування» публікуються наукові матеріали, які раніше не публікувалися в інших виданнях.

Мова статті: українська, 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: alengoro@isofts.kiev.ua. FAX: +380 (44) 526 6263, Телефон: +380 (96) 418 3082.

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 десятиковою 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. Джерела з заголовками на латиниці наводяться без перекладу. Інші джерела подаються мовою оригіналу. Приклади оформлення бібліографічних посилань згідно з вимогами **Harvard Style** наведені в багатьох публікаціях, наприклад: 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 – англійською і українською 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 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”.

*16.07.2020 р. набули чинності положення Закону України «Про забезпечення функціонування української мови як державної». Відповідно до статті 22 «Державна мова у сфері науки» у наукових виданнях не повинно бути вміщено матеріалів іншими мовами, окрім державної, англійської та мов ЄС.

Примітка: Підписний індекс журналу «Проблеми програмування» – **90853**.

