ISSN 2415-7740 (Print)
ISSN 2415-7074 (Online)

Belarusian State University of Informatics and Radioelectronics

Open Semantic Technologies for Intelligent Systems

Research Papers Collection

Founded in 2017

Issue 5

Minsk
2021

UDC 004.822+004.89-027.31

The collection includes peer-reviewed articles approved by the Editorial board. Designed for university professors, researchers, students, graduate students, undergraduates, as well as for specialists of enterprises in the field of intelligent systems design.

Editorial address:
220005, Minsk, Platonova str., 39, rm. 606 b
Phone: +375 (17) 293-80-92
E-mail: ostisconf@gmail.com
Web-site: ***http://proc.ostis.net***

The collection is included in the Russian Science Citation Index

ISSN 2415-7740 (Print)
ISSN 2415-7074 (Online)

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Открытые семантические технологии проектирования интеллектуальных систем

Сборник научных трудов

Основан в 2017 году

Выпуск 5

Минск
2021

УДК 004.822+004.89-027.31

Сборник включает прошедшие рецензирование и утвержденные Редакционной коллегией статьи.

Предназначен для преподавателей высших учебных заведений, научных сотрудников, студентов, аспирантов, магистрантов, а также для специалистов предприятий в сфере проектирования интеллектуальных систем.

Адрес редакции:

220005, г. Минск, ул. Платонова 39, каб. 606 б

Телефон: +375(17)293-80-92

Электронный адрес: ostisconf@gmail.com

Сайт: **http://proc.ostis.net**

Сборник включён в Российский индекс научного цитирования

ОРГАНИЗАТОРЫ КОНФЕРЕНЦИИ OSTIS-2021

Министерство связи
и информатизации
Республики Беларусь

HIGH TECH PARK HTP BELARUS

BAAI

Оипи

БрГТУ БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Савушкин

РИВШ

Qulix SYSTEMS

ВирусБлокАда

ARISTEK SYSTEMS

billwerk.

REPLY WM

dwynn.

deltatre

Devs_Group devsgroup.net

eComCharge gate to payments

Semantic intelligent semantic systems

LeverX

Octonion TECHNOLOGY

Paralect the launch company

Spark Lab

innowise

FusionTech.

ТЕХНИЧЕСКАЯ И ИНФОРМАЦИОННАЯ ПОДДЕРЖКА

ИНФОРМАТИКА

Научный журнал
ОНТОЛОГИЯ ПРОЕКТИРОВАНИЯ

Научно-технический журнал
Автоматизация процессов управления
Automation of Control Processes
Journal of Science

международный журнал
ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

ВЕСНІК СУВЯЗІ

TABLE OF CONTENTS

СОДЕРЖАНИЕ

FOREWORD

Research papers collection “Open Semantic Technology for Intelligent Systems” is devoted to the flexible and compatible technologies development that ensure the rapid and high-quality design of intelligent systems for various purposes.

The collection reflects research in the field of artificial intelligence in the following areas:

- Hybrid intelligent computer systems;
- Intelligent human-machine systems;
- Computer vision;
- Fuzzy computing ;
- Intelligent agents ;
- Intelligent automation;
- Knowledge management;
- Knowledge engineering;
- Ontological design;
- Semantic networks;
- Machine learning;
- Neural networks;
- Natural-language interface

The main focus of this issue is on the development of models, methods and tools that ensure the semantic compatibility of intelligent computer systems and their ability to coordinate their activities in the collective solution of complex problems.

In total, the collection contains 42 articles. The editors are thankful for all authors who sent their articles. Scientific experts selected for publication the best of the submitted works, many of them were revised in accordance with the comments of reviewers.

We are grateful our scientific experts for their great job in reviewing the articles in close cooperation with the authors. Their work allowed to raise the level of scientific results presentation, and also created a platform for further scientific discussions.

We hope that, as before, the collection will perform its main function — to promote active cooperation between business, science and education in the field of artificial intelligence.

Editor-in-chief
Golenkov Vladimir

ПРЕДИСЛОВИЕ

Сборник научных трудов «Открытые семантические технологии проектирования интеллектуальных систем» посвящен вопросам разработки гибких и совместимых технологий, обеспечивающих быстрое и качественное построение интеллектуальных систем различного назначения.

В сборнике отражены исследования в сфере искусственного интеллекта по следующим направлениям:

- Гибридные интеллектуальные компьютерные системы;
- Интеллектуальные человеко-машинные системы;
- Компьютерное зрение;
- Нечеткие вычисления;
- Интеллектуальные агенты;
- Интеллектуальная автоматизация;
- Управление знаниями;
- Инженерия знаний;
- Онтологическое проектирование;
- Семантические сети;
- Машинное обучение;
- Искусственные нейронные сети;
- Естественно-языковой интерфейс.

Основной акцент в этом выпуске сборника сделан на разработку моделей, методов и средств, обеспечивающих семантическую совместимость интеллектуальных компьютерных систем и их способность координировать свою деятельность при коллективном решении сложных задач.

В общей сложности сборник содержит 42 статьи. Редакция сборника благодарит всех авторов, представивших свои статьи. Для публикации научными экспертами были отобраны лучшие из представленных работ, многие из них были переработаны в соответствии с замечаниями рецензентов.

Мы также благодарим экспертов за большой труд по рецензированию статей в тесном взаимодействии с авторами, который позволил повысить уровень изложения научных результатов, а также создал платформу для дальнейших научных дискуссий.

Надеемся, что, как и прежде, сборник будет выполнять свою основную функцию — способствовать активному сотрудничеству между бизнесом, наукой и образованием в области искусственного интеллекта.

Главный редактор
Голенков Владимир Васильевич

Methodological problems of the current state of works in the field of Artificial intelligence

Vladimir Golenkov, Natalia Guliakina
Belarusian State University
of Informatics and Radioelectronics
Minsk, Belarus
golen@bsuir.by, guliakina@bsuir.by

Vladimir Golovko
Brest State Technical University
Brest, Belarus
vladimir.golovko@gmail.com

Viktor Krasnoproshin
Belarusian State University
Minsk, Belarus
krasnoproshin@bsu.by

***Abstract*—The article describes the strategic goals of Artificial intelligence and the main problems of scientific and technological activities in this field. The problems relevant for the development of the main directions and forms of its activity are defined. Approaches to their solution based on a new technological wave are suggested and issues important for the successful development of this research and practical discipline as a whole are discussed.**

***Keywords*—OSTIS, ostis-system, ontological approach, intelligent computer system, General theory of intelligent systems, knowledge base, problem-solving model, semantic representation of information, SC-code**

I. Introduction

The paper describes the strategic goals of *Artificial intelligence* and the main problems of scientific and technological activities in this field. The problems relevant for the development of the main directions and forms of its activity are defined. Approaches to their solution based on a new technological wave are suggested and issues important for the successful development of this research and practical discipline as a whole are discussed.

The main ones among others are the following:

- what will the complex automation of various types of *human activity* look like, built on the basis of new *Artificial intelligence technologies*?
- how will the interaction *between humans* be organized in this case, i.e., what will the architecture of a modern *smart society* look like?
- is the current level of semantic compatibility sufficient for mutual understanding between virtual computer systems, intelligent computer systems and their users and what needs to be done to improve it?

For a deeper consideration of methodological problems, it is proposed to divide them into the following basic parts.

Problems that hinder the development of Artificial intelligence as a scientific and technical discipline:

- the development of scientific research in the field of *Artificial intelligence*;
- the development of technologies for building and forming the market of *ICS* (*intelligent computer systems*, artificial intelligent systems);
- educational problems in the field of *Artificial intelligence*;
- the development of business in the field of *Artificial intelligence*.

Problems of automation of complex activities:

- research activities in various scientific disciplines;
- the creation of *technologies* for developing complex technical systems [1], [2];
- engineering activities for the development of complex technical systems;
- *educational* activities in research-intensive technical specialties;
- the definition of construction principles of the *OSTIS Technology* (Open Semantic Technology for Intelligent Systems), designed to solve the above problems.

The structure of the Ecosystem, built using the OSTIS Technology and that provides complex automation of all types of human activity.

Let us first specify the structure of scientific and technological activities in the field of *Artificial intelligence* as a scientific discipline.

Artificial intelligence is a field of human activity, the main purposes of which are:

- the building up of the theory of *intelligent systems*;
- the creation of the technology for the development of *ICS*;
- the transition to a new level of complex automation of all types of *human activity*, based on the mass usage of *ICS*.

The last assumes the presence of *ICS* that can understand each other and coordinate their activities. It is also necessary to build up a General theory of *human activity*, carried out in the conditions of a new level of its automation (the theory of activity of a *smart society*). Such an activity requires a significant rethinking of its organization and should be "understandable" to *intelligent computer systems*.

II. Current state and main tendencies of development of *Artificial intelligence*

A. Research activities

Most directions of *Artificial intelligence* are characterized by the inconsistency of the concept system and, as a result, the lack of their semantic compatibility and convergence [3], [4], [5]. All this hinders the building up of a general (with a high level of formalization) theory of intelligent systems.

This is due to the lack of understanding and motivation in a convergence between different directions. And since there is no urgent practical need, there is also no movement aimed at building up of the *General theory of intelligent systems*.

B. Development of the basic complex technology for the design of ICS

Modern technology of *Artificial intelligence* is a family of particular technologies. They are usually focused on the development and maintenance of various types of *ICS* components. At the same time, they implement various models of representation and processing of information as well as various problem-solving models focused on the development of various classes of *ICS*. As a result:

- the complexity of development increases, which requires highly qualified performers;
- complex technologies and tools for the development of *ICS* are missing;
- there is no syntactic and semantic compatibility of particular technologies and system integration of certain components.

C. Development of technology for building up designed intelligent computer systems

Traditional *(von Neumann) computers* are not able to effectively interpret the all variety of models used in *ICS*. Attempts to develop new generation computers focused on *ICS* were unsuccessful. The development of specialized *computers* focused on the interpretation of a single model does not solve the problem.

D. Specialized engineering in the field of Artificial intelligence

The development and production of *ICS* based on existing models, methods and tools face the following problems:

- there is no clear systematization of the all variety of *ICS*, that corresponds to the systematization of automated types of *human activity*;
- there is no convergence of *ICS* involved in the process of automating one type of *human activity*;
- there is no semantic compatibility, unification and mutual understanding of *ICS*;
- the semantic “unfriendliness” of the user interface and the lack of a built-in help system lead to a low rate of usage of the *ICS* capabilities.

Further automation of *human activity* requires increasing the level of *intelligence* of the corresponding *ICS* and implementing their ability to perform the following actions:

- to establish semantic compatibility (mutual understanding) between *computer systems* and their users;
- to maintain semantic compatibility during the evolution of users and other *computer systems*;
- to coordinate their activities with users and *computer systems* in the collective solving of various *problems*;
- to participate in the allocation of work (subproblems) in the collective solving of various *problems*.

The implementation of these capabilities will create an opportunity for full automation of the processes of system *integration* of *ICS* (into complexes of interacting systems) and their reengineering. This:

- will enable the complexes of cybernetic systems to adapt independently to solving new problems and will improve the efficiency of the operation of such complexes;
- will significantly reduce the number of errors compared to the “manual” system integration.

Thus, the next level of automation of human activity requires the creation of *ICS*, which could independently unite with the complexes necessary for solving complex problems [6].

E. Educational activities in the field of artificial intelligence

Any activity in the field of *Artificial intelligence* combines a high degree of research intensity and complexity of engineering work. Therefore, the training of specialists in this field requires the simultaneous formation of their research and engineering-practical skills, culture and thinking. The combination of fundamental scientific and engineering-practical training of specialists is a complex educational-pedagogical problem, since:

- there is no systematic approach to training young specialists in the field of *Artificial intelligence*;
- the lack of *semantic compatibility* between academic subjects leads to a “tesselation” of information perception;
- there is no personification of training and an attitude to the identification and development of individual abilities;
- there is no purposeful formation of motivation for creativity and skills of working in real development teams;
- there is no adaptation to real practical activities.

Any modern technology should have a high rate of development and requires highly qualified executive human resources.

Therefore, educational activities in the field of *Artificial intelligence* and the corresponding technology should be deeply integrated into all activities in the field of *Artificial intelligence*.

F. Business activities in the field of Artificial intelligence

There is an urgent need to increase the level of automation in many areas of human activity (in industry, medicine, transport, education, construction activity and many others). In addition, the development of *Artificial intelligence technologies* has led to a significant expansion of work on the creation of *applied intelligent computer systems*. All this, in turn, has led to the setting up of many commercial organizations focused on the development of such applications. Currently, there are the following problems here:

- it is difficult to ensure a balance of tactical and strategic directions for the development of all forms of activity in the field of *Artificial intelligence*;
- there is no deep convergence of forms of activity, which complicates the development of each of them;
- the high research intensity of works requires highly qualified performers and their ability to work as part of creative teams.

For advanced training of employees and ensuring a high level of development, business needs active cooperation with various scientific schools and departments that train young specialists in the field of *Artificial intelligence* as well as active participation in the preparation and holding of corresponding conferences, seminars and exhibitions.

III. Modern problems of artificial intelligence

Based on the problems listed above, two main problems can be distinguished:

A. *Building up and permanent development of the general (formal) theory of intelligent systems, **which includes**:*
   - the clarification of the requirements and features of *ICS* that determine the level of their intelligence;
   - convergence and integration of various types of knowledge and problem-solving models within each *ICS*;
   - a focus on the development of unified semantically compatible formal models and their universal interpreters;
   - ensuring a clear stratification and independence of the processes of evolution of formal models and their interpreters;
   - ensuring communication compatibility that allows *ICS* to independently form groups of *ICS* and their users, independently coordinate activities within groups when solving complex problems in unpredictable conditions.

B. *Creation and permanent development of a complex technology for the design and production of semantically compatible ICS that can independently coordinate their activities. **To do this, it is necessary to take into account**:*
   - the description of the standard of *ICS*, which ensures their semantic compatibility [7];
   - the development of libraries of semantically compatible and reusable components;
   - a low threshold for users and developers to enter the technology of the design of *ICS*;
   - high rates of technology development due to the active involvement of authors of various applications;
   - the development of new generation computers focused on the production of high-performance *ICS* of various purposes;
   - the creation of a global *ecosystem* that provides complex automation of all *types of human activity*;
   - the creation and permanent development of a global *sociotechnical ecosystem* (*ICS* and users), which provides complex automation of all *types of human activity*;
   - the transition from the eclectic building up of complex *ICS* to their deep **integration** and unification, when the same models of representation and processing of knowledge are implemented in the same way everywhere;
   - the reduction of the distance between the ***theory of ICS*** and the practice of their development.

Let us consider this in more detail. The epicenter of modern problems of development of activities in the field of *Artificial intelligence* is the *convergence* and *deep integration* of all forms, directions and results of this activity. The level of interrelation, interaction and *convergence* between various forms and areas of activity in the field of *Artificial intelligence* is insufficient. This leads to the fact that each of them develops separately, independently of the others. The question is about the *convergence* between such directions of *Artificial intelligence* as the representation of knowledge, the solution of intellectual problems, intelligent behavior, understanding, etc. as well as between such forms of *human activity in the field of Artificial intelligence* as scientific research, technology development, application development, education, business. Why, when contrasted with the long-term intensive development of scientific research in the field of *Artificial intelligence*, the market of *intelligent computer systems* and the complex technology of *Artificial intelligence*, which provides the development of a wide range of *intelligent computer systems* for various purposes and is available to a wide range of engineers, have not yet been created? Because the combination of a high level of research intensity and pragmatism of this problem requires for its solution a fundamentally new approach to the organization of interaction between *scientists* who work in the field of *Artificial intelligence*, *developers* of facilities for design automation of *intelligent computer systems*, *developers* of tools for implementing intelligent computer systems, including hardware support tools for intelligent computer systems, *developers* of applied *intelligent computer systems*. Such purposeful interaction should be carried out both within the framework of each of these forms of activity in the field of *Artificial intelligence* and between them. Thus, the basic tendency of further development of theoretical and practical works in the field of *Artificial intelligence* is the convergence of both the most different types (forms and directions) of *human activity* in the field of *Artificial intelligence* and a variety of products (results) of this activity. It is necessary to eliminate barriers between different types and products of activity in the field of *Artificial intelligence* to ensure their compatibility and integrability. The problem of creating a rapidly developing market of semantically compatible *intelligent systems* is a challenge addressed to specialists in the field of *Artificial intelligence*, which requires overcoming "the Babel" in all its occurrences, the formation of a high culture of negotiability and a unified, coordinated form of representation of collectively accumulated, improved and used knowledge. Scientists, who work in the field of *Artificial intelligence*, should ensure the convergence of the results of different directions of *Artificial intelligence* [8], [9] and build up:

- the general theory of intelligent computer systems;
- the general technology for designing semantically compatible intelligent computer systems, which includes the corresponding standards of intelligent computer systems and their components. Engineers, who develop intelligent computer systems, should cooperate with scientists and participate in the development of the technology of the design of intelligent computer system.

IV. OSTIS Technology (Open Semantic Technology for Intelligent Systems)

To solve the above problems, the *OSTIS Technology* is proposed – a complex open semantic technology for the design, production, operation and *reengineering* of hybrid, semantically compatible, active and negotiable *intelligent computer systems* [10], which is focused on:

- the development of *intelligent computer systems* that have a high level of *intelligence* and, in particular, a high level of *socialization*. The specified systems developed by the *OSTIS Technology* we will call ***ostis-systems***;
- the complex automation of all types and areas of *human activity* by creating a network of interacting and coordinating their activities *ostis-systems*. The specified network of *ostis-systems* together with their users will be called an ***OSTIS Ecosystem***;
- the support of the permanent evolution of *ostis-systems* during their operation;
- convergence and integration based on the semantic representation of knowledge of various research areas of *Artificial intelligence* (in particular, all kinds of basic knowledge and skills for solving intellectual problems) within the framework of the *General formal semantic theory of ostis-systems*;
- the development of new generation computers that provide effective (including productive) interpretation of logical-semantic models of *ostis-systems*, which are represented by knowledge bases of these systems that have a semantic representation.

The *OSTIS Technology* is implemented as a network of *ostis-systems*, which is part of the *OSTIS Ecosystem*. The key *ostis-system* of the specified network is the ***IMS.ostis*** *(Intelligent MetaSystem)* metasystem that implements the ***Core of the OSTIS Technology***, which includes basic (subject-independent) methods and tools for designing and producing of *ostis-systems* with the integration of typical embedded subsystems for supporting operation and reengineering of *ostis-systems* into their composition. The other *ostis-systems*, which are part of the network under consideration, implement various specialized *ostis-technologies* for designing various classes of *ostis-systems* that provide automation of any areas and *types of human activity*, except for designing *ostis-systems*.

The storage of information in the memory of the *ostis-system* focuses on the semantic representation of information – without synonymy and homonymy of signs, without semantic equivalence of information constructions. An abstract memory of the *ostis-system* is dynamic graph (i.e., nonlinear (graph) and restructurable). Information processing in the memory of the *ostis-system* is reduced not so much to changing the state of memory elements (this happens only when changing the syntactic type of elements and when changing the contents of those elements that denote files) as to changing the configuration of the relations between them. The organization of problem solving in the memory of the *ostis-system* is based on the *agent-oriented model of information processing*, controlled by situations and events that occur in the processed information (more precisely, in the processed *knowledge base*). From the point of view of architecture, the *ostis-system* is a hierarchical multi-agent system with shared memory (i.e., with memory that is available to all agents of the *ostis-system*).

Note that the shared memory of most of the *multi-agent systems* currently being studied is not common but distributed, i.e., it is an abstract (virtual) integration, which includes the memory of each agent of a multi-agent system. Coordination of the activities of agents of the *ostis-system* when performing complex *actions in the memory of the ostis-system* is also implemented through the *memory of the ostis-system* with the help of *methods* for solving various *classes of problems* as well as with the help of *plans* for solving particular problems, stored in memory. Based on this, it is possible to build up an unbound hierarchical system of *agents of the ostis-system* – from atomic agents that ensure the performance of basic actions in the memory of the *ostis-system* to non-atomic agents that are communities (groups) of atomic and/or non-atomic agents that provide the solution of various typical problems using appropriate methods and plans.

An important element of such a system is the implementation of decentralized situational control of the operation of *ostis-systems* not only at the level of internal information processes but also at the level of organization of individual activities in the environment and even at the level of participation in collective activities within various communities of *ostis-systems*. The organization of the performance of actions in the environment is carried out by the *ostis-system* as follows:

- classes of *atomic actions in the environment* are distinguished, for the implementation of each of which *effector agents of the ostis-system* are introduced;
- coordination of the activities of *effector agents of the ostis-system* when performing *complex actions in the environment* is carried out through the *memory of the ostis-system* with the help of *methods and plans* for solving various problems *in the environment*, stored in memory, as well as with the help of *receptor agents of the ostis-system* that provide feedback and, accordingly, sensorimotor coordination.

In the *ostis-system*, the basic set (basic system) of the used *concepts* is unified, which is the basis for ensuring *semantic compatibility* of all *ostis-systems*. The structuring of information (*knowledge base*) stored in the memory of the *ostis-system* is based on the hierarchical system of *subject domains* and their corresponding *formal ontologies*. **As a result, the ostis-system has the following features:**

- the ability for semantic immersion (semantic integration) of new acquired knowledge (including new skills) into the current state of the *knowledge base*;
- the ability for *semantic convergence* (detection of similarities) of new acquired knowledge (and, in particular, skills) with knowledge that is part of the current state of the *knowledge base of the ostis-system*;
- the ability for integration of different types of knowledge;
- the ability for integration of various problem-solving models;
- the ability of *ostis-systems* for understanding each other as well as any of its users by coordinating the system of used concepts (by terms and by denotational semantics);
- the ability of the *ostis-system* for providing and maintaining a high level of its *semantic compatibility* (a high level of mutual understanding) with other *ostis-systems* in the process of its evolution as well as the evolution of other *ostis-systems*, which leads to the extension and/or correction of the system of used concepts;
- the ability of *ostis-systems* for correlating, coordinating their activities with other systems in solving problems that cannot be solved either in principle or in a reasonable time by the efforts of one (individual) intelligent computer system;
- a high degree of individual learnability of *ostis-systems*, provided by a high degree of their flexibility, stratification, reflexivity as well as the universality of the facilities for the representation and creation of knowledge;
- **a high degree of collective learnability of *ostis-systems***, provided by a high degree of their semantic compatibility.

The listed features of *ostis-systems* demonstrate that they have a significantly higher *level of intelligence* and, in particular, a higher *level of socialization* compared to modern *intelligent computer systems*.

A. Semantic representation of information

Each syntactically elementary (atomic) fragment of the represented information is an indication of some entity, which can be real or abstract, concrete (fixed, constant) or certain (variable), permanent or temporary, coherent (reliable) or fuzzy (unreliable, with possible additional clarification of the likelihood degree). It follows herefrom that the semantic representation of information cannot include letters (they do not denote entities), words, word combinations (they are not elementary fragments), separation and limit symbols (they do not indicate entities).

Within the framework of the semantic representation of information, there is no synonymy (pairs of synonymous signs), homonymy (homonymous signs), semantic equivalence (pairs of semantically equivalent information constructions), i.e., there is no duplication of information as well as the ambiguity of the correlation between signs and their denotations. **Therefore**, the semantic representation of information cannot look like a linkage (string, sequence) of syntactically elementary fragments, since each described entity and its sign that mutually identically corresponds to it can be connected not with two but with any number of described entities. In other words, the semantic representation of information is a nonlinear (graph) information construction. It follows that if the internal representation of information in the memory of a computer system is a semantic representation, then the information processing in such memory is dynamic graph and is reduced not to changing the state of memory elements but to changing the configuration of relations between them.

As already noted, the key problem of the current stage of the development of the general theory of intelligent computer systems and the technology of their development is the problem of ensuring ***semantic compatibility*** of various types of knowledge that are part of *knowledge bases* of *intelligent computer systems*; various types of *problem-solving models*; various *intelligent computer systems* in general.

To solve this problem, it is necessary to unify (standardize) the form of representation of knowledge in the memory of intelligent computer systems. Our proposed approach for such unification is the orientation to the ***semantic representation of information*** (knowledge) in the memory of intelligent computer systems. The basis of the approach to ensuring a high level of learnability and semantic compatibility of intelligent computer systems as well as to developing a standard for intelligent computer systems is a unification of the semantic representation of information (knowledge) in the memory of intelligent computer systems and the creation of a global ***semantic space*** of knowledge. **It should also be noted that** the information in the sign construction is generally contained not in the signs themselves (in their structure) but in the relations between them. At the same time, these relations (syntactic relations) must have a coherent semantic interpretation.

B. Semantic network

We consider the semantic network not as a beautiful metaphor of complex structured sign constructions but as a formal clarification of the concept of semantic representation of information, as a principle of representation of information that underlies a new generation of computer languages and computer systems themselves – graph languages and graph computers.

A semantic network is a sign construction that has the following features:

- it is not necessary to take into account the "internal" structure (construction) of the signs included in the semantic network during its semantic analysis (understanding);
- the meaning of a semantic network is determined by the denotational semantics of all the signs included in it and the configuration of the incident relations of these signs;
- one of two incident signs included in the semantic network is a sign of linkage;
- the lack of synonymy, homonymy.

C. Standardization of ostis-systems

Standardization of *ostis-systems* includes:

- standardization of the language of the internal representation of information in the memory of *ostis-systems*;
- standardization of the principles of decentralized control of information processing in the memory of *ostis-systems*;
- standardization of the language for describing situations and events (in the memory of *ostis-systems*), which are the conditions for initiating various information processes in the memory of *ostis-systems*;
- standardization of the base language for specification (description, programming) of agents that perform the corresponding information processes in the memory of *ostis-systems*;
- standardization of base languages for input/output of information to/from the memory of *ostis-systems*.

The standard is a *knowledge base* being permanently improved, the evolution of which is supported by the corresponding portal.

D. Convergence, integration of knowledge and problem-solving models

For the convergence of knowledge, a universal base language of the internal semantic representation of knowledge in the memory of *ostis-systems (SC-code)* is introduced, according to the structure of which all internal languages focused on the representation of knowledge of various types (logical languages, languages for the representation of problem-solving models (in particular, programs), the language of problem definition, ontological languages and many others) are *sublanguages of the SC-code*, the syntax of which completely coincides with the syntax of the *SC-code*. **As a result**, the convergence of different knowledge is reduced to the coordination of systems of concepts used to represent different types of knowledge. Such coordination is focused on increasing the number of common concepts used in the representation of various knowledge.

Let us consider the convergence of ***problem-solving models in the ostis-system***. It is based on the following core principles:

- the syntax of the language for representing the corresponding class of problem-solving methods in memory is the syntax of the *SC-code*;
- the denotational semantics is described in the form of a corresponding ontology and is represented as a text of the *SC-code*;
- the operational semantics of each problem-solving model is a group of agents; it can be hierarchical based on different solver models, but there is a basic model for interpreting any methods:
  - *SCP language*:
    - the syntax is coincident with the syntax of the *SC-code*;
    - the denotational semantics is a procedural programming language in dynamic graph memory;
    - the operational semantics is implemented at the level of a software or hardware platform;
  - *sc-agents* work in a common environment (*sc-memory*) in parallel, asynchronously, based on a number of rules that allow them not to "hinder" each other.

The integration of knowledge in the memory of *ostis-systems* is reduced to the pasting together (matching) of synonymous signs. Since the problem-solving model used by the *ostis-system* is represented in the memory of the *ostis-system* as the corresponding type of knowledge, the integration of various problem-solving models can occur in the *ostis-system* in the same way as the integration of any other types of knowledge. In addition, when the question is about the integration of different problem-solving models, the possibility of simultaneous usage of different problem-solving models when processing the same knowledge and, in particular, when solving the same problem is supposed. This possibility in the *ostis-system* is provided by the *agent-oriented model of information processing* in the memory of the *ostis-system*. Thus, this kind of integration of various problem-solving models for *ostis-systems* is trivial.

Let us list the key advantages of the *ostis-system*:

- a high level of the ability of the *ostis-system* to perform semantic integration of knowledge in its memory (in particular, when immersing new knowledge in the current state of the knowledge base) is provided by the semantic nature of the internal encoding of information stored in the memory of the *ostis-system* and, in particular, by the fact that homonymous signs and pairs of synonymous signs are improper in the internal code of the knowledge base of the *ostis-system*;
- a high level of ability to integrate various types of knowledge in *ostis-systems* is ensured by the fact that each language focused on the representation of knowledge of the corresponding type is a sublanguage of the same base language of the *SC-code*;
- a high level of ability to integrate various problem-solving models in *ostis-systems*, which is ensured by the fact that all these models are focused on processing information represented in the *SC-code*; the same fragment of the knowledge base of the *ostis-system* (i.e., the same construction of the *SC-code*) can be simultaneously processed by several different problem-solving models; all problem-solving models in *ostis-systems* are integrated using the same basic problem-solving model – an *scp-model of problem solving*;
- a high level of learnability of the *ostis-system*, which is provided by a high level of semantic flexibility of information stored in the memory of the *ostis-system*, since each deletion or addition of a syntactically elementary fragment of stored information as well as the deletion or addition of each incident relation between such elements has a coherent semantic interpretation; by a high level of stratification of stored information, which is provided by the ontologically oriented structuring of the knowledge base of the *ostis-system*; by a high level of reflection of the *ostis-system*, which is provided by the powerful metalanguage capabilities of the language of the internal representation of information (knowledge) in the memory of *ostis-systems*. As a result, each *ostis-system* has a high *level of learning* (the ability to quickly extend their *knowledge and skills*) and a high *level of socialization* (the ability to effectively participate in the activities of various communities – ones that consist of ostis-systems and ones that consist of *ostis-systems* and humans.

E. Level of semantic compatibility

There are sharply defined formal notions that determine the *level of semantic compatibility* (level of semantic convergence) of various knowledge, skills, entire *ostis-systems* (more precisely, the knowledge bases of these systems). It is obvious that the *level of semantic compatibility* is primarily determined by the number of "adherent points" in the compared *knowledge, skills **and** knowledge bases* – ***there are signs*** that are present in <u>different</u> compared objects, but that have the same denotations (i.e., that denote the same entities). At the same time, among such signs that denote the same entities and are present in different objects being compared, the signs that denote *concepts* are of prime importance. The number of such common *concepts* in the compared *knowledge*, *skills*, *knowledge bases* determines the level of semantic compatibility (level of coherence) of the systems of the used *concepts* in the compared specified objects. An increase in the number of signs that denote the same entities and are present in different compared objects can lead to the fact that not only semantically equivalent signs but also whole semantically equivalent fragments (entire information constructions) will be present in different specified compared objects. It is important to emphasize that semantically equivalent sign constructions represented in the internal language of *ostis-systems* (in the *SC-code*), in the memory of different *ostis-systems*, are always syntactically isomorphic graph constructs, in which the isomorphism correspondence connects signs stored in the memory of different *ostis-systems*, but that denote the same entities (more precisely, the same entity). Note also that within the memory of each *ostis-system*, the synonymy of signs and, accordingly, the semantic equivalence of sign constructions are improper.

Due to the constantly developing semantic standards of the *OSTIS Technology*, which are represented by a system of formal ontologies for a variety of subject domains, the *ostis-systems* being developed initially have a sufficiently high *level of semantic compatibility* with all other *ostis-systems*. Moreover, in the *OSTIS Technology*, an entire core of all *ostis-systems* is allocated, that contains fundamental basic knowledge and basic skills that are the same for all *ostis-systems* and that allows each copy of this core to develop (communicate, specialize) in any direction.

Each *ostis-system*, interacting with humans (users) or with other *ostis-systems*, can increase the level of semantic compatibility (mutual understanding) with them as well as maintain a high level of such compatibility in the conditions of (1) its evolution, (2) the evolution of other *ostis-systems* and *users*, (3) the evolution of semantic standards of the *OSTIS Technology*. This interaction is generally focused on coordinating changes in the system of concepts used, i.e., adjustment of the corresponding fragments of ontologies.

V. Conclusion

Due to the high level of semantic compatibility of *ostis-systems* and the semantic representation of knowledge in the memory of *ostis-systems*, the complexity of semantic analysis and understanding of information received (reported, transmitted) to the *ostis-system* from other *ostis-systems* or users is significantly reduced and its quality is improved. Each *ostis-system* is able:

- to independently or by invitation join the *ostis-community* (community of *ostis-systems*) or the *ostis-group* that consists of *ostis-systems* and humans. Such communities and groups are created on a temporary (one-time) or permanent basis for the collective solution of complex problems;
- to participate in the distribution (including coordination of the distribution) of problems – both "one-time" and long-term ones (responsibilities);
- to monitor the state of the entire process of collective activities and coordinate own activities with the activities of other members of the group in case of possible unpredictable changes in the conditions (state) of the corresponding environment.

The high level of intelligence of *ostis-systems* and, accordingly, the high level of their independence and purposefulness allow *ostis-systems* to be full members of a wide variety of communities, within which *ostis-systems* gain the right to independently initiate (based on a detailed analysis of the current situation and the current state of the community action plan) a wide range of actions (problems) performed by other members of the community and thereby participate in the coordination of the activities of community members. The ability of the *ostis-system* to coordinate its activities with other *ostis-systems* as well as to adjust the activities of the entire group of *ostis-systems*, adapting to various changes in the environment (conditions), in which this activity is carried out, allows automating the activities of the system integrator both at the stage of assembling the group of *ostis-systems* and at the stage of its updating (reengineering).

The advantages of *ostis-systems* are provided by:

- the advantages of the *SC-code* – the language of internal encoding of information stored in the memory of *ostis-systems*;
- the advantages of the organization of *sc-memory* – the memory of *ostis-systems*;
- the advantages of *sc-models* of knowledge bases of *ostis-systems* – by means of structuring such knowledge bases;
- the advantages of *sc-models of problem solving* – agent-oriented problem-solving models used in *ostis-systems*.

Acknowledgment

The authors would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help and valuable comments.

The work was carried out with the partial financial support of the BRFFR (BRFFR-RFFR ***).

Список литературы

[1] A. Kolesnikov, *Gibridnye intellektual'nye sistemy: Teoriya i tekhnologiya razrabotki [Hybrid intelligent systems: theory and technology of development]*, A. M. Yashin, Ed. SPb.: Izd-vo SPbGTU, 2001.

[2] G. Rybina, "Dynamicheskye integrirovannye expertnye systemi: tekhnologiya avtomatizirovannogo polucheniya, predstavlniya i obrabotki temporalnykh znani [dynamic integrated expert systems: technology for automated acquisition, presentation and processing of temporal knowledge]," *Informatcionnye izmeritelnye i upravlyayuschye systemi [Information measuring and control systems]*, vol. 16, no. 7, p. 20–31, 2018.

[3] V. Taranchuk, "Vozmozhnosti i sredstva wolfram mathematica dlya razrabotki intellektual'nykh obuchayushchikh sistem [opportunities and means of wolfram mathematica for developing intelligent tutoring systems]," *«Nauchnye vedomosti Belgorodskogo gosudarstvennogo universiteta. Seriya: Ekonomika. Informatika» [«Scientific statements of Belgorod State University. Series: Economy. Computer science»]*, vol. 33, no. 1 (198), pp. 102–110, 2015.

[4] A. E. Yankovskaya, A. A. Shelupanov, A. N. Kornetov, N. N. Ilinskaya, and V. B. Obukhovskaya, "Gibridnaya intellektual'naya sistema ekspress-diagnostiki organizatsionnogo stressa, depressii, deviantnogo povedeniya i trevogi narushitelei na osnove konvergentsii neskol'kikh nauk i nauchnykh napravlenii [hybrid intelligent system of express diagnostics of organizational stress, depression, deviant behavior and anxiety of violators based on convergence of several sciences and scientific directions]," in *Trudy kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «IS&IT'17». Nauchnoe izdanie v 3-kh tomakh. [Works of congress on intelligent 17 scientific publication in 3 volumes]*, ser. T. 1. Stupin A. S. publishing House, Taganrog, 2017, pp. 323–329.

[5] A. Palagin, "Problemy transdisciplinarnosti i rol' informatiki [problems of transdisciplinarity and the role of informatics]," *Kibernetika i sistemnyj analiz [Cybernetics and Systems Analysis]*, no. 5, p. 3–13, 2013.

[6] M. Kovalchuk, "Konvergencija nauk i tehnologij – proryv v budushhee [convergence of science and technology - a breakthrough into the future]," *Rossijskie nanotehnologii [Russian nanotechnology]*, vol. 6, no. 1–2, p. 13–23, 2011.

[7] V. Golenkov, N. Guliakina, V. Golovko, and V. Krasnoproshin, "Artificial intelligence standardization is a key challenge for the technologies of the future," in *Open Semantic Technologies for Intelligent System*, V. Golenkov, V. Krasnoproshin, V. Golovko, and E. Azarov, Eds. Cham: Springer International Publishing, 2020, pp. 1–21.

[8] A. F. Tuzovsky and V. Z. Yampolsky, "Integracija informacii s ispol'zovaniem tehnologij semantic web [integration of information using semantic web technologies]," *Problemy informatiki [Problems of Informatics]*, no. 2, p. 51–58, 2011.

[9] V. Haarslev, K. Hidde, R. Möller, and M. Wessel, "The racerpro knowledge representation and reasoning system," *Semantic Web*, vol. 3, no. 3, p. 267–277, 2012. [Online]. Available: https://doi.org/10.3233/SW-2011-0032

[10] V. Golenkov, N. Guliakina, I. Davydenko, and A. Eremeev, "Methods and tools for ensuring compatibility of computer systems," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2019, pp. 25–52.

Методологические проблемы современного состояния работ в области Искусственного интеллекта

Голенков В.В., Головко В.А.,
Гулякина Н.А., Краснопрошин В.В.

В работе описаны стратегические цели Искусственного интеллекта и основные задачи научно-технической деятельности в этой области. Обозначены проблемы, актуальные для развития основных направлений и форм его деятельности. Предлагаются подходы к их решению, основанные на новом технологическом укладе, и обсуждается вопросы, важные для успешного развития данной научно-практической дисциплины в целом.

Received 31.05.2021

The technology for the development of viable intelligent services

Valeria Gribova, Philip Moskalenko,
Vadim Timchenko, Elena Shalfeyeva
Lab. of Intelligent Systems named after A.S. Kleshchev
Institute of Automation and Control Processes FEB RAS
Vladivostok, Russian Federation
gribova@iacp.dvo.ru, philipmm@iacp.dvo.ru,
vadim@iacp.dvo.ru, shalf@iacp.dvo.ru

***Abstract*—The paper presents a technology for the development of intelligent multi-agent services. This technology is put to reduce the labor-intensiveness of intelligent cloud application development and maintenance. The key aspect of the technology is an independent development of knowledge bases, a user interface and a problem solver as an assembly of agents, and their integration into an intelligent service. The principal specific of the technology are: two-level ontological approach to the formation of knowledge and data bases with a clear separation between the ontology and knowledge base (database) formed on its basis; the ontology-based specification of not only the domain terminology and structure, but also of the rules for knowledge and data formation, control of their integrity and completeness; division of all software components into declarative and procedural parts; unified semantic representation of all declarative components (ontologies, knowledge and data bases, declarative parts of software components), etc. The proposed approaches are supported by the IACPaaS cloud platform where expert-oriented formation of each service component with an appropriate tool is provided.**

***Keywords*—intelligent system, multi-agent system, intelligent software development technology, hierarchical semantic network, cloud platform, cloud service**

I. Introduction

Development and maintenance of an intelligent system (IS) with knowledge are hard and labor-intensive processes. This is primarily due to the fact that a knowledge base (KB) is a part of such a system's architecture which brings in some specificies. That's why special tools are used. The typical representatives are: Level5 Object, G2, Clips, Loops, VITAL, KEATS, OSTIS, AT-technology, RT Works, COMDALE/C, COGSYS, ILOG Rules, Protégé and others [1], [2], [3], [4], [5]. The considerable differences between them are determined by the formalisms used to represent knowledge, methods and tools for their acquisition, forming and debugging, the used inference mechanisms, and also technologies for IS with a KB (hereinafter KBS – knowledge based system) development and maintenance.

However, the problem of creating tools for development and maintenance of KBS is far from being solved. Scientific publications mention the following main problems that need to be dealt with. First of all, there is still an open issue of including domain experts in the process of developing and maintaining knowledge bases, ensuring their real impact on the quality of computer systems being developed. Experience in developing complex computer systems shows that the intermediation of programmers between designed computer systems and experts significantly distorts the contribution of the latter ones. When developing a new generation of tools, it is not programmers who should dominate, but experts who are able to accurately represent their knowledge [6], [7]. The developed models and methods of knowledge acquisition solve the problem of their initial formation, however, as noted in [8], it is the maintenance phase that is the most complex and significant. The KB is a component of KBS that changes much more often than other components, so expert-oriented knowledge creation and maintenance tools providing is an important and relevant task. KB maintenance implies its refinement or improvement, which at the same time does not break the performance of the IS (i.e., a change in the knowledge base should not lead to the need of changing the solver and its user interface). Among the problems of existing tools for creating KBS, the following is also worth mentioning: the use of various approaches and mechanisms for creating knowledge bases, solvers and interfaces, complex linking of these components, or vice versa, the lack of a clear separation between KBS components, which makes it difficult to reuse solutions when creating other KBS. Our long-term experience in creating practically useful KBS for solving problems in various domains has shown an urgent need to construct problem-oriented shells with accompanying instrumental support.

The key requirement for any complex software system, including KBS, is its viability, which is implemented in software engineering through architectural solutions (separation into loosely coupled components with logically clear functions), declarative representation of software components of a system, automation of code fragments generation, reuse of components, separation of competencies between developers of different types. One of the well-known solutions for implementing the requirement to involve domain experts in the KBS development process is the ontology-based (metainformation-based) KB formation, using a semantic knowledge repre-

This work was partially supported by the Russian Foundation for Basic Research (RFBR) (grant numbers 19-07-00244, 20-07-00670)

sentation model (the knowledge base is separated from the ontology) [9].

In software engineering, an agent-based approach is actively used to create viable software systems [10], which in comparison with object-oriented programming has potentially greater flexibility, gives the possibility of agent reuse, and simplifies parallelization (which is important for IS as many tasks have great computational complexity [11]). However, the issue of creating a comprehensive technology that supports the development of all KBS components with ontological knowledge bases remains open.

To address these challenges in the development and maintenance of viable KBS we have proposed a concept of the development toolkit [12], which supports the following technological principles:

- knowledge base, problem solver and user interface (UI) are developed separately;
- a single language is used to describe the ontology of knowledge (metainformation for the knowledge base) and models of all declarative components (their metainformation);
- a unified declarative semantic (conceptual) representation of all information (ontologies, knowledge bases, databases) resources and software components of KBS is used – a labeled rooted hierarchical binary digraph with possible loops and cycles (thus universality of their software processing and user editing can be achieved);
- formation and maintenance of knowledge bases is carried out by domain experts and is based on knowledge ontologies;
- UI of the knowledge base editor is generated for experts on the basis of ontology (metainformation);
- method of problem solving is divided into subproblems, each one is solved by a correspondent software agent, either a reflective or a reactive one [13] – a component of a problem solver;
- software agents are provided with an API for access to information resources;
- KBS is available as a cloud service.

This paper describes the technology for the development of a KBS as a cloud multi-agent service, using such toolkit.

II. Intelligent service development technology

Technological foundations supported by the proposed toolkit concept are consistent with the general trend when KBS development is based on the use of methods and tools for ontological modeling of design and specification processes of developed systems, i.e. ontological engineering tools (Ontology-Based (-Driven) Software Engineering [14]); reusability of ready-to-use solutions (components); involving stakeholders in the development process through knowledge portals.

Methods of formalizing knowledge based on the domain ontology and ontologies of known intelligent problems allow us to create structured and at the same time understandable to domain experts knowledge, as well as systematization of terms.

Further, knowledge, formalized on the basis of an ontology, are either integrated with ready-to-use problem-oriented software solvers, or require the creation of new software components. Thereby the proposed technology for the development of intelligent services consists of the following processes: *assembly of an intelligent service from components* and possibly *development of those components* (Fig. 1). Components are represented by rectangles, and activities – by rounded rectangles. The components are information ones (metainformation and information) and software ones (a problem solver, its agents, message templates for their interaction, UI). Symbols on arcs have the following meaning. **sel** – searching for and selecting the appropriate component (to which an arc is pointing). **[ ]** – optionality, i.e. the pointed activity is performed if the required component is absent or if the condition (italic text on the arc) is true. **+** ("plus") – multiplicity, meaning that the selection or creation (development) of the corresponding component can be performed more than once.

III. Intelligent service development

An intelligent *service* consists of a *problem solver* (which has been integrated with *formal parameters* and with a *UI*) and actual parameters (*input actual parameters* – information resources accessible only for reading and *output actual parameters* – information resources accessible for CRUD modification). Distinction of service components into information and software ones pursues the following objectives:

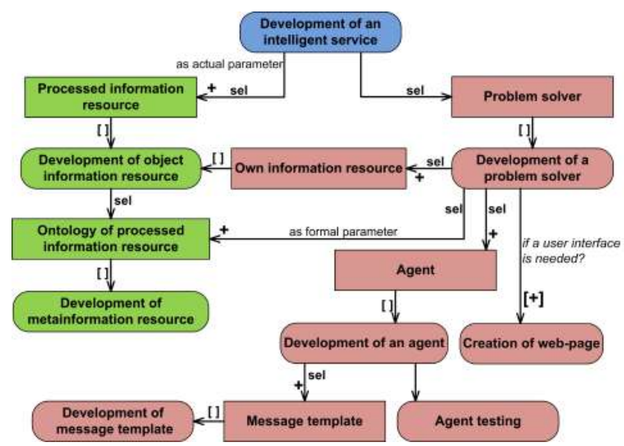


Figure 1. Intelligent service development technology.

- independent development by different groups of specialists;
- reuse – the same problem solver can be bound with various information resources and vice-versa.

In both cases the compatibility (between those two types of components) is provided at the level of *formal parameters* of the *solver*.

A *formal parameter* of the *problem solver* is an information resource which represents metainformation (ontology), an actual parameter is one of the information resources which represent information and which are formed in terms of this metainformation.

The declarative specification of a *service* is formed in two stages:

1) creation of a new information resource of “service” type with setting of its name and *Service structure* information resource (representing a language for declarative specification of platform services) as metainformation;
2) creation of content of this new information resource (service assembly) with the use of *Editor for digraphs of information* [15], where the process of editing is controlled by metainformation *Service structure* and consists of the following:
   - a link to the *problem solver* (to root vertex of digraph which represents it) is created,
   - for each *formal parameter* of the *solver* (in order of its appearance in the description of solver) a link to the corresponding actual parameter (to root vertex of digraph which represents it) is created.

IV. Development of knowledge bases and databases

A. General description

Network (graph) structures are now widely used as a visual and universal mean for representing various types of data and knowledge in different domains. In principle, with the help of such structures that are best suited for explicitly representing associations between different concepts, it is possible to describe any complex situation, fact or problem domain. At the same time, as noted, for example, in [16] various kinds of information (data and knowledge), regardless of the concrete syntax of the language for their representation, in the abstract syntax, in the general case, can be represented as (multi-) graphs, possibly typed.

In accordance with a 2-level approach for formation of information resources [17], [18], two types of them are distinguished by the abstraction level of represented information. They are information resources which represent ontology (i.e. metainformation – abstract level) and information resources which represent knowledge and data bases (i.e. information – object level).

B. Object information resources development

Development of an *information resource* which is processed by an intelligent service and represents information requires another information resource which represents its metainformation to be present in the storage. Otherwise it must be developed as described in the next subsection. The development of an *information resource* consists of two stages:

1) creation of a new information resource with setting of its name and metainformation;
2) formation of its content by means of the *Editor for digraphs of information* where the process editing is controlled by set metainformation.

During the work of the *Editor for digraphs of information* it forms and maintains a correspondence between the arcs of digraphs of information and metainformation. Formation is carried out in “top-down” way: from vertices which are composite concepts to vertices which are atomic concepts. This process starts from the root vertex. In this case the user doesn’t have to sharply envision and keep in mind the whole structure (connections between vertices) of the formed digraph as in the case of “bottom-up” way.

The UI of the *Editor for digraphs of information* is generated by the metainformation. This implies that as the latter one describes an ontology for knowledge or data in some domain so its experts can create and maintain knowledge bases or databases in terms of their customary systems of concepts (without mediators, i.e. knowledge engineers).

C. Metainformation resources development

The development of a *metainformation* resource consists of two stages:

1) creation of a new information resource with setting of its name and metainformation (in such case it is an information resource which contains the description of the *language for metainformation digraph representation*);
2) formation of its content in a “top-down” way (starting from the root vertex of the digraph) with the use of the *Editor for digraphs of metainformation* (this step also makes up the maintenance process which may automatically modify correspondent object information in order to keep it in consistency with modified metainformation).

A digraph of metainformation describes the abstract syntax of a structural language in terms of which digraphs of information are further formed. The language for metainformation digraph representation is declarative, simple, and at the same time powerful enough to describe arbitrary models, which are adapted to the domain terminology and to the form adopted by the developers of KBS components as well as tools for their creation. A detailed description of the language is given in [19].

Users of the *Editor for digraphs of metainformation* are metainformation carriers who are usually knowledge engineers and systems analysts from various fields of professional activities. The editing process model which is set into the basis of the *Editor for digraphs of metainformation* has much in common with the one which is the basis of the *Editor for digraphs of information*. The differences are caused only by formalism of representation of the correspondent digraphs and by the fact that metainformation digraph can have an arbitrary form (limitations are set only by expressive means of the *language for metainformation digraph representation*), whereas information digraph has a structure limited by its metainformation digraph.

V. Problem solver development

A *problem solver* is a component of some intelligent service which processes information resources and which encapsulates business logic for problem solving. It is a set of agents, each of them solves some subproblem(s) and interacts with others by message exchange. A lifecycle of a message starts from its creation by some agent, followed by it being sent to another agent which receives and processes it. Then a message ceases to exist.

The proposed approach for development of agent communication means is a multilanguage one. Messages must be represented in some language(s) which syntax and semantics must be understood by interacting agents. Each language can have an arbitrary complex syntax structure (represented by metainformation digraph), and contents of messages (represented by digraphs of information) are formed in accordance with it.

There are two agents within the solver (in general case) which have particular roles:

- *root agent* – an agent to which an *initializing message* is sent by utility agent *System agent* when a service starts to run after its launch (meaning that this agent runs first among all agents of problem solver);
- *interface controller agent* – an agent whose interaction with the utility agent *View agent* (see section VIII) provides coupling of UI with problem solver (in case when a service with UI is developed).

The mentioned *initializing message* is represented in the communication language which belongs to the class of utility ones. Another such distinguished languages are used for interacting with utility agents and for stopping the work of the problem solver.

During the process of service execution agents which are parts of the problem solver connect with each other dynamically. In case of a service without UI this interaction starts from the *root agent*. In the case of a service with a UI it usually starts from the *interface controller agent* – for processing the events which are generated in the UI (the *root agent* may do no work at all in that case).

In order for a problem solver to become usable for various services at the stage of their assembly the information resource which represents the declarative specification of this problem solver must be created. It is formed in two stages:

1) creation of a new information resource of "problem solver" type with setting of its name and *Problem solver structure* information resource as metainformation;
2) formation of content of this new information resource (problem solver assembly) with the use of *Editor for digraphs of information* where the process of editing is controlled by the set metainformation and consists of setting of the following:
   - information about the purpose of the problem solver,
   - a link to its *root agent*,
   - links to *formal parameters* (in case when services should process information resources, i.e. must have actual parameters at runtime),
   - a *UI* which includes a link to *interface controller agent* (in case when a service with the UI is being developed), links to own information resources (shared among all running instances of the problem solver).

The described organization of a problem solver (which implies dedication of a special *interface controller agent* among others) gives an opportunity to separate development and maintenance of problem solver business-logic from same work on UI. This leads to the possibility of involving independent appropriate specialists to these types of work.

VI. Agent development

An *agent* is a (possibly reusable) software component which interacts with other agents with the use of message reception/sending. It is capable of processing (reading and modification) information resources. Reusability means that an agent can become a part of various problem solvers with no modification. Data processing is organized in form of productions which are grouped into one or several production blocks – by the amount of message templates that an agent can process.

An agent consists of two parts: a declarative one and a procedural one. Declarative part of an agent consists of two sections: agent's documentation (which is presented by a set of descriptions: for an agent itself and for each of its production block, written in natural language) and formal specification for its set of production blocks (which comprise an agent). The declarative part is used as a basis for the support of automation of agent's documented source code development and maintenance. A digraph 2-level model of information resources representation allows storing of declarative description and code of an agent (procedural part) in a single information resource which represents this agent.

Development of an agent consists of the following stages:

1) creation of new information resource of "agent" type with setting of its name and *Agent structure* information resource as metainformation;
2) formation of content of this new information resource with the use of *Editor for digraphs of information* (with extensions) where the process of editing is controlled by the set metainformation and consists of setting of the following initial data: agent name, agent class name, description, local data structure and production blocks specification (description, templates of incoming messages and corresponding templates of outgoing messages)[1];
3) acquiring agent source code (by generating its sketch using its declarative description or using pre-saved source code) and executable code of used message templates;
4) writing (modifying) the source code of an agent (code for its production blocks in particular) and forming its executable code (as a result of compilation of its source code);
5) uploading source code and executable code into the correspondent information resource (thus extending agent's declarative specification);
6) agent testing (optional).

[1]Different production blocks of the same agent must have different incoming message templates.

Acquiring agent source code and executable code of reused message templates must be done with extended functionality of the *Editor for digraphs of information*. For agent source code it provides downloading of either sketch of source code which is generated on the basis of new agent declarative description or downloading of the stored source code of modified agent.

Writing the source code of an agent must be done using some modern programming language powerful enough for solving intelligent problems. A suitable IDE can be used or such functionality can be implemented within the toolkit (within some extension of the *Editor for digraphs of information*). The source code of agent's production block implements the whole of part of the ontology-based algorithm for knowledge-based processing using toolkit API methods for: incoming message data reading, knowledge base traversing, outgoing message creation. Note that an agent can make an arbitrary number of messages to be sent to a set of other agents (including system ones and itself).

After the source code of an agent is ready it is necessary to form its ready-to-run version (e.g. bytecode) and load it into the information resource which represents an agent. In order to achieve this a compilation of the source code must be committed either locally or online (if the toolkit has such support). While code uploading and/or compilation it is checked for correctness and safety.

Agent testing is carried out by a separate tool which performs multiple start and execution of the set of its production blocks on a provided set of tests (formed as information resources with metainformation *Agent tests structure* by use of the *Editor for digraphs of information*). Reports are saved with results of test executions. A single test generally includes the information resource representing incoming message, a set of information resources representing expected outgoing messages, and tuples of information resources which act as input, output (initial and final states) actual parameters and own ones (initial and ending states). A test is considered to be passed if the amount and the contents of the outgoing messages and of processed information resources are as expected. Formed sets of tests can be used for regressive testing during the stage of agent maintenance.

VII. Message template development

A *message* in multi-agent systems is a mechanism for coordinated interaction of agents which provides exchange of information between them and transfer of requests [20]. Agents communicate in different languages whose amount is extensible (by means of creating new message templates and adding new production blocks into agents or extending existing ones) and is limited only by the total amount of message templates. Specific languages of agent interaction (message templates) are set at the level of separate production blocks.

As messages are object information resources so languages for sets of messages are represented by information resources that are metainformation – *Message templates*. They not only contain the structure for a set of messages but may also hold a set of methods for processing these messages. Thus, like an agent, a message template consists of two parts – a declarative one and a procedural one. Its structure is simpler though: name, class name, description, message structure, source code and code.

Development of the message template consists generally of the same stages as of an agent. The differences appear at the stages of creation (when the *Message template structure* information resource is set as metainformation) and of writing the source code (due to differences in the structure of declarative specifications of agents and message templates).

Message template testing can be done only through testing of an agent (see section VI).

VIII. User interface development

Development of an interface for an intelligent cloud service is a development of web-interface (as services are available online and are accessed via web-browsers). The interface consists of interface controller agent (development described in section VI) and a set of web-pages (with one selected as the starting one). Each *web-page* has a name and data which can be of *content* or *design* type:

- *content* – a mixture of text, *ui tags* and a set of names of *design web-pages* (of the same problem solver);
- *design* – a description of CSS classes (a set of CSS rules).

Such separation allows:

- to apply various CSS to the same contend of the interface and vice versa – to use the same CSS for interface of various services, which leads to increase of flexibility and adaptability of developed interfaces and to simplification of their maintenance;
- to divide the processes of development and maintenance (thus, to make them independent) of these parts and to involve independent appropriate specialists to these types of work (their only interaction would lie in setting/using same names for classes in CSS).

The *ui tags* of a *web-page* are processed as *ui requests* (one request per tag) sent through *View agent* to agents which act as *interface controllers* within solvers at the initial *web-page* display. Further interactions of user with web-page elements produce other *ui requests* (determined by those elements). Each *ui request* is a set of pairs: *parameter* = *value*. After the request is passed to solver agent, the *View agent* waits for result and in case it is a fragment of UI – puts it into the shown web-page content in place of the processed *ui tag*.

A detailed scheme of interaction of the *View agent* with Web-server and *interface controller agent* (which is a part of some problem solver) is shown in Fig. 2. A process of interaction consists of request transferring from browser to agents and returning of result to browser. A processing of a request is performed in between. Numeration of arrows in Fig. 2 sets the order of interaction.

The basis of the UI presentation model is an "MVC" (Model-View-Controller) conception [21]. Its projection on this conception on the proposed interface model is as follows.

1. Model. This component includes:

- *Model of abstract user interface* – an information resource which represents metainformation and holds a description of structure for standard interface elements of WIMP-interfaces and a way of their organization into a single nested structure.
- *Abstract interface generation API* – a set of high-level functions for creation of fragments of abstract interfaces. Performing calls of these functions (with necessary arguments) significantly increases the level of abstraction at which the information resource that represents some abstract interface is formed. To form a description for some fragment of an abstract interface one has to construct a superposition of function calls of this API.

2. View. This component is presented by the utility agent *View agent*, which is a "hybrid" one. It is divided into two parts so that it can mediate between the two:

- Web-server – it interacts with external part of the *View agent*,
- *interface controller agents* of problem solvers – they interact with agent part by receiving, processing and replying to messages created by *Request from View agent* utility message template.

The other tasks of the *View agent* are the following:

- the production of specific interface (HTML-code) on the basis of the its abstract model (with the use of built-in mapping rules for all supported interface elements);
- uploading/downloading binary data to/from Web-server.

3. Controller. This component is represented by agents which play a role of *interface controller agent* (within problem solvers). These agents interact with the *View agent* by message exchange. Such messages are created by particular message templates. Agents implement (possibly by interaction with other agents) logic for processing *ui requests* of the following origins: *ui tags* (from web-pages content data) and ui events (generated by interface elements in response to user actions).

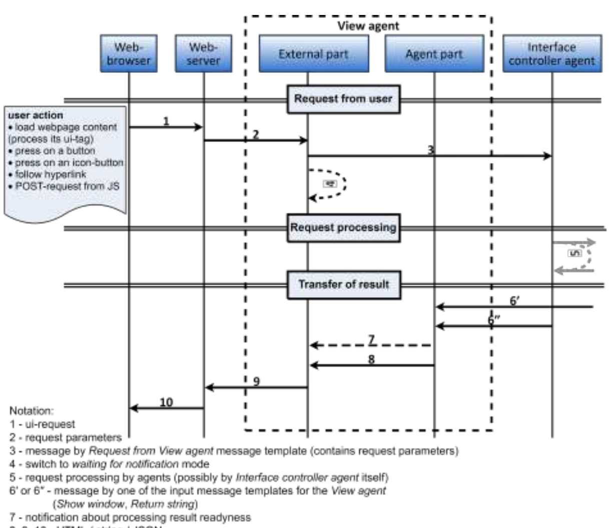


Figure 2. A scheme of interaction of the *View agent* with Web-server and *interface controller agent*.

A result of processing a *ui request* is either an information resource (which represents a description of an abstract interface which is passed to the agent part of the *View agent*) or an arbitrary string of characters.

The development of UI consists of the following steps:

1) creation of a set of *web-pages* within the declarative specification of the used problem solver – with formation of content for each page;
2) development of *interface controller agent*, which must implement the logic for processing *ui requests*.

IX. Discussion and future work

Let's highlight the main features that distinguish the described technology for the development of cloud services and their components from other available solutions.

The possibility to include domain experts in the process of developing knowledge bases and databases. The proposed technology supports a two-level ontological approach to the formation of information resources. Its feature is a clear separation between the ontology and the knowledge base (database) formed on its basis. The ontology (metainformation) sets not only the structure and terminology for a knowledge base, but also the rules for its formation, control of the integrity and completeness. The ontology is formed by a knowledge engineer (possibly in cooperation with domain expert) with the use of the ontology editor. On its basis, a UI is automatically generated that allows domain experts to create (form) knowledge bases and databases without involving professional intermediaries. All information resources have a semantic representation that is understandable to domain experts.

Providing reuse of ontologies and problem solvers. A clear separation between ontology and knowledge bases (databases) also provides another significant advantage. When an ontology is formed – the solver is designed in its terms. Then using this ontology, an arbitrary number of knowledge bases can be developed and their binding with the solver (which is integrated with UI and plays the role of KBS shell) makes the new KBS. So this moves us from developing of specific KBSs each time from a scratch to developing of shells first which can then be used with different knowledge bases to comprise different KBSs. Thus, the ontology and the solver are reused for a whole class of problems.

Modification of knowledge bases without changing the problem solver's code. This is also provided by the separation between the knowledge base and its ontology, the lack of domain knowledge in the solver. The ontological solver is implemented not as an inference in calculus, but as an algorithm that traverses the knowledge base in accordance with its ontology to match statements in the knowledge base with the input data and, thus, consistently confirming or refuting elements of knowledge.

Transparency and maintainability of problem solvers. The development of KBS solvers is based on the processing of hierarchical graphs of concepts, which makes it possible to create a set of common APIs (application programming interfaces) for working with such graphs. Based on the specifics of the problem, the expected repeatability of their smaller subproblem, the developer can choose software components that will provide a more transparent architecture of solvers. The use of an agent-based approach provides the possibilities to parallelize the execution of subtasks, which is important for problems with high solvation speed requirements.

It is important that the model (metainformation) of each software component is hard-set and allows one to create them using the common editor or its appropriate specialized adaptations (Fig. 3). This increases components' transparency and maintainability, as it is known from software engineering [22] that declarative components are easier to maintain than procedural ones. A unified declarative representation of agents, as well as message templates made it possible to automate the creation (process of generation) of code templates for them.

The proposed technology is implemented on the IACPaaS platform [12] – a cloud computing system for support of development, control and remote use of multi-agent services. Thus, it provides not only users with remote access to applied intelligent systems but also developers with remote access to appropriate tools for creation of such systems and their components which make this process more automatic (Figs. 4, 5, 6).

Figure 3. Usage of the development toolkit for creating information and software components of intelligent services.

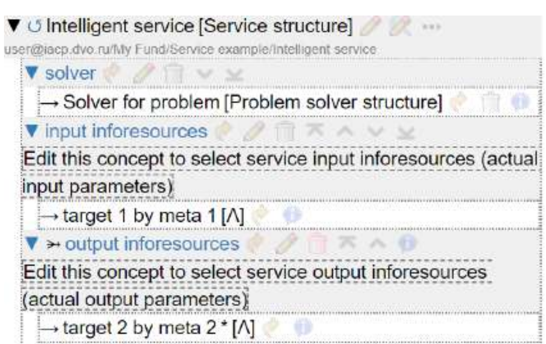


Figure 4. UI fragment of the *Service editor*.

Currently, the IACPaaS platform has several hundreds of users who, using the proposed technology, have created ontologies, knowledge bases, tools and problem solvers for various domains: a set of tools for developing professional virtual

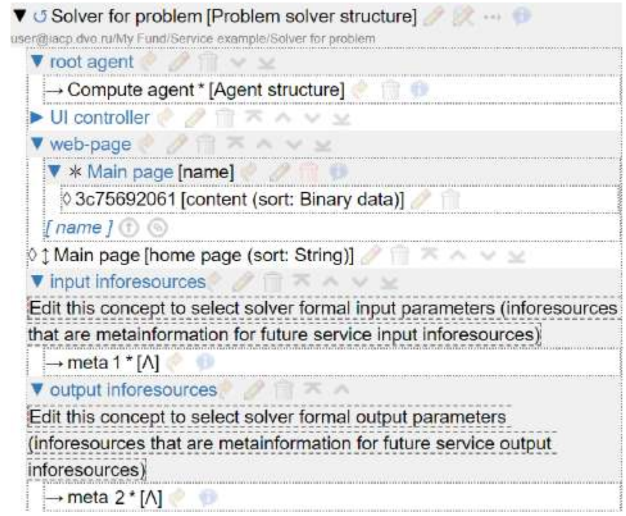


Figure 5. UI fragment of the *Problem solver editor*.

Figure 6. UI fragment of the *Agent editor*.

cloud environments [23], services for interactive verification of intuitive mathematical proofs, underwater robotics, practical psychology, agriculture and others. In particular, the formed portal of medical knowledge includes a wide range of medical ontologies, complex knowledge bases formed on their basis (for example, the knowledge base for medical diagnosis of diseases of various nosologies contains more than 130 000 concepts) and problem solvers (computer-based training simulator using classical research methods in ophthalmology, diagnosis of diseases [24], differential diagnosis of acute and chronic diseases: infectious diseases, gastro-intestinal diseases, hereditary diseases, diseases of the cardiovascular and respiratory systems, etc.), the prescription of personalized medical and rehabilitation treatment, etc.).

By now, users of the platform have created more than 2 thousands of resources, rough numbers are: almost 1750 information resources (ontologies – 650, knowledge and data bases – 1100) and 500 software components (problem solvers – 180, agents – 250, message templates – 70). So we can say that the active use of the platform demonstrates that the technology proposed by the authors meets modern requirements for the development of viable KBS and fits the needs of users.

However, we continue to work on improving the platform and technology for KBS development. Our main efforts are aimed at creating tools for generating adaptive interfaces, tools for intelligent user support and design automation. Special attention is paid to the creation of specialized technologies for developing classes of KBS and increasing the level of instrumental support for various technologies. Important tasks are the improvement of tools for safety and security of the platform, of the common APIs for processing the storage units and creation of high level abstraction operations for information resources.

References

[1] Gensym G2. The World's Leading Software Platform for Real-Time Expert System Application. Available at: http://www.gensym.com/wp-content/uploads/Gensym-l-G2.pdf (accessed 2021, May).

[2] V. V. Golenkov, N. A. Gulyakina, I. T. Davydenko, D. V. Shunkevich, Semantic technologies of intelligent systems design and semantic associative computers, Doklady BGUIR, 2019, 3, pp. 42–50.

[3] M. A. Musen, The Protégé Project: A Look Back and a Look Forward. AI Matters, 2015, 4, pp. 4–12.

[4] G. V. Rybina, Intellektual'nye sistemy: ot A do YA. Seriya monografij v trekh knigah. Kn. 3. Problemno-specializirovannye intellektual'nye sistemy. Instrumental'nye sredstva postroeniya intellektual'nyh system (Intelligent systems: A to Z. A series of monographs in three books. Book 3. Problem-specialized intelligent systems. Tools for building intelligent systems), M.: Nauchtekhlitizdat, 2015, 180 p. (in Russian).

[5] G. Rodríguez, Á. Soria, M. Campo, Artificial intelligence in service-oriented software design. Engineering Applications of Artificial Intelligence, 2016, 53, pp. 86–104.

[6] I. Gupta, G. Nagpal, Artificial Intelligence and Expert Systems. Mercury Learning & Information, 2020, 412 p.

[7] G. Tecuci, D. Marcu, M. Boicu, D. Schum, Knowledge Engineering: building Cognitive Assistants for Evidence-based Reasoning, Cambridge, U.K. Cambridge University Press, 2016.

[8] C. E. Grant, D. Z. Wang, A Challenge for Long-Term Knowledge Base Maintenance, Journal of Data and Information Quality, 2015, 6(2-3):7.

[9] R. S. Pressman, B. G. Maxim, Software Engineering: Practitioner's Approach, New York, 9th ed. McGraw-Hill, 2019, 704 p.

[10] A. Torreño, Ó. Sapena, E. Onaindia, FMAP: A platform for the development of distributed multi-agent planning systems. Knowledge-Based Systems, 2018, 145, pp. 166–168.

[11] M. Gholamian, G. Fatemi, M. Ghazanfari, A Hybrid System for Multiobjective Problems – A case study in NP-hard problems, Knowledge-Based Systems, 2007, 20(4), pp. 426–436.

[12] V. V. Gribova, A. S. Kleschev, Ph. M. Moskalenko, V. A. Timchenko, L. A. Fedorischev, E. A. Shalfeeva, A cloud computing platform for lifecycle support of intelligent multi-agent internet-services. In proc. of International Conference on Power Electronics and Energy Engineering (PEEE-2015) (19-20 Apr. 2015), Hong Kong, pp. 231–235.

[13] AI portal. Agent classification. Available at: http://www.aiportal.ru/articles/multiagent-systems/agent-classification.html (accessed 2021, May).

[14] V. F. Khoroshevsky, Proyektirovaniye sistem programmnogo obespecheniya pod upravleniyem ontologiy: modeli, metody, realizatsii [Ontology Driven Software Engineering: Models, Methods, Implementations], Ontologiya proyektirovaniya [Ontology of designing], 2019, 9(4), pp. 429–448. (in Russian).

[15] V. V. Gribova, A. S. Kleshchev, F. M. Moskalenko, V. A. Timchenko, Implementation of a Model of a Metainformation Controlled Editor of Information Units with a Complex Structure. Automatic Documentation and Mathematical Linguistics, 2016, 1, pp. 14–25.

[16] G. Taentzer, K. Ehrig, E. Guerra, J. Lara, L. Lengyel, T. Levendovszky, U. Prange, D. Varro, S. Varr´o-Gyapay, Model transformation by graph transformation: A comparative study, in: Proceedings Workshop Model Transformation in Practice, Montego Bay, Jamaica, 2005, pp. 1-–48.

[17] C. Atkinson, T. Kuhne, Model-driven development: A metamodeling foundation, IEEE Software, 2003, 5, pp. 36–41.

[18] A. Kleppe, S. Warmer, W. Bast, MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley Professional, Boston, 2003.

[19] V. V. Gribova, A. S. Kleshchev, F. M. Moskalenko, V. A. Timchenko, A two-level model of information units with complex structure that correspond to the questioning metaphor. Automatic Documentation and Mathematical Linguistics, 2015, 5, pp. 172–181.

[20] V. S. Dyundyukov, V. B. Tarasov, Goal-resource networks and their application to agents communication and co-ordination in virtual enterprises, IFAC Proceedings Volumes, 2013, 46(9), pp. 347–352.

[21] R. Trygve, MVC. Xerox PARC 1978-79. Available at: https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html (accessed 2021, May)

[22] I. Sommerville, Software Engineering, 10th edition, Pearson, 2015.

[23] V. V. Gribova, L. A. Fedorischev, Software toolset for development of cloud virtual environments. Software products and systems, 2015, 2, pp. 60–64.

[24] V. Gribova, Ph. Moskalenko, M. Petryaeva, D. Okun, Cloud environment for development and use of software systems for clinical medicine and education. Advances in Intelligent Systems Research, 2019, 166, pp. 225–229.

Технология разработки жизнеспособных интеллектуальных сервисов

В.В. Грибова, Ф.М. Москаленко,
В.А. Тимченко, Е.А. Шалфеева

Предложена технология разработки интеллектуальных мультиагентных облачных сервисов. Ее ключевым аспектом является независимая разработка баз знаний, пользовательского интерфейса и решателя задач в виде ансамбля агентов, а также их интеграция в интеллектуальный сервис. Технология реализована на облачной платформе IACPaaS, обеспечивающей экспертно-ориентированное создание каждого компонента сервиса с помощью соответствующего инструментария.

Received 28.05.2021

Method for Evaluating the Hybrid Intelligent Multi-Agent System's Cohesion: Consistency of the Problem-Solving Protocol

Kirikov Igor
Kaliningrad Branch of the Federal Research Center
"Computer Science and Control"
of the Russian Academy of Sciences
Kaliningrad, Russia
baltbipiran@mail.ru

Listopad Sergey
Kaliningrad Branch of the Federal Research Center
"Computer Science and Control"
of the Russian Academy of Sciences
Kaliningrad, Russia
ser-list-post@yandex.ru

***Abstract*—The paper is aimed at developing of a new class of distributed artificial intelligence systems, namely, cohesive hybrid intelligent multi-agent systems. The concept of such systems was proposed in order to simplify the integration processes of heterogeneous intelligent agents created by various development teams. It is assumed that the agents of such systems should be able to independently agree on their goals, domain models and develop a problem-solving protocol. This paper presents one of the elements of the method for evaluating the cohesion of agents in the system, namely, evaluating the consistency of the problem-solving protocol.**

***Keywords*—cohesion, hybrid intelligent multi-agent system, team of specialists, protocol consistency**

I. Introduction

The impossibility of constructing an omniscient agent for solving practical problems, i.e. possessing all the necessary abilities, knowledge and resources to solve them [1], is one of the reasons for the emergence of the multi-agent approach to developing artificial intelligence systems. It assumes that an individual agent can have only a partial view of the general problem and is able to solve only some of its subproblems, and to tackle the problem as a whole it is necessary to organize their effective interaction.

Within the multi-agent approach, systems are developed to solve problems of varying degrees of modeling complexity [2], which is largely due to differences in approaches to the definition of the "agent" concept. In a weak sense, an agent is a software or hardware implemented system with the properties of autonomy, social behavior, reactivity, and proactivity [3], [4]. Stronger definitions of the "agent" concept imply endowing it with such anthropomorphic qualities as knowledge, beliefs, intentions, obligations and emotions [5]–[7]. This corresponds to the social aspect of solving practical problems in teams of specialists under the guidance of a decision maker. To model such teams, the concept of hybrid intelligent multi-agent systems (HIMAS) is proposed [2].

The reported study was funded by RFBR according to the research project №20-07-00104A.

HIMAS is a hybrid intelligent system (HIS) [2], elements of which are realized as autonomous agents [1]. HIMAS, like HIS, integrate several artificial intelligence technologies to obtain problem-solving method overcoming disadvantages of its components and capable to solve problems with high modeling complexity [2]. Like in multi-agent systems (MAS), in HIMAS relations between agents and their environment are dynamically rebuilt depending of roles played by agents in certain conditions and interactions between them. Thus HIMAS combine the positive aspects of hybrid intelligent and multi-agent systems compensating their shortcomings such as lack of autonomy of HIS's elements and unclear specification of MAS's notion "agent": HIMAS have to contain heterogeneous intelligent agents with extensive domain models and goal-setting mechanisms, modelling the knowledge and reasoning of the relevant specialists "at a round table".

Traditionally, HIMAS were built by a single development team, and the agents had a common domain model, interaction protocol, and goals predetermined by the system developers. If agents are created by different independent development teams, a situation may arise when agents "speak" in different languages, use incompatible protocols, have conflicting goals and domain models, which requires additional labor costs to integrate them into a single system. To reduce labor costs for the integration of heterogeneous intelligent agents, it is proposed to implement mechanisms for modeling the processes of uniting a team of specialists within a new class of intelligent systems, namely cohesive hybrid intelligent multi-agent systems (CHIMAS) [8]. They will make it possible not only to synthesize a method for solving a problem over a heterogeneous model field [2] and to imitate the group work of specialists [1], but also to form a cohesive team of agents who understand each other and share common goals and norms.

II. Group cohesion in teams of specialists

Group mechanisms in small groups, and in particular in teams of specialists solving problems "at a round table", are studied within the group dynamics, i.e. the social psychology direction created by K. Levin [9]. Group dynamics studies social tension, aggression and discrimination, conflicts, leadership, psychological atmosphere, adaptation in a group, forming and deviation from group norms, group reaction, and others macro-level phenomena and processes in groups of people. One of important roles in group dynamics is played by researches on cohesion, which cover the processes of building of a single socio-psychological community in the group, and emergence of team's group properties preventing violation of its psychological integrity [10]. Studies of various group types, such as sport, business teams, military units and laboratory groups confirm the increase in work efficiency, the uniformity of the contribution of participants to the overall result and the stability of the group with increasing cohesion, when norms of the group behavior contribute to this [11]. Comprehensive reviews of the current state of group cohesion research are given in [12]–[16].

A. V. Petrovsky developed one of the most fundamental model in understanding the cohesion of groups and teams [17]. According to his stratometric concept (SC), cohesion of a small group of specialists consists of three layers (strata), representing three levels of group development:

- external level, at which emotional interpersonal relationships are investigated [10], [18]–[22];
- value-orientational unity (VOU), which considers the relations, mediated by joint activities that serve as the basis for formation of the basic values unity [10], [12], [17], [23]–[27];
- the core, which reveals the motives of the group members' choices of each other, mediated by common values that arise because members share the team goals [10], [16], [20], [21], [25], [28]–[33].

The similarity of goals, unity of opinions, points of view to the problem and group norms characterizing cohesion should be distinguished from the conformity, i.e. change in the behavior and attitudes of people aimed at promoting the actions and decisions of others [34]. Conformal behavior determines the appearance of a groupthink, i.e. a style of thinking when the desire for agreement among group members becomes more important than a realistic assessment of the situation and alternative solutions [35]. As shown in [36], the emergence of a groupthink strongly depends on the type of cohesion that exists in the group, and is typical mainly for groups with a high cohesion of the external level (the first stratum of the SC), due to emotional-interpersonal relationships. The groups with predominance of the cohesion of the second and third levels of the SC, called task cohesion in [36], show low levels of conformity and groupthink.

III. Cohesive hybrid intelligent multi-agent system model

Based on the HIMAS model [2], and taking into account SC, the CHIMAS model can be formulated as follows:

$$chimas = < AG, env, INT, ORG, \{glng, ontng, protng\} >, \quad (1)$$

where AG is the set of agents, containing decision-making agent ag^{dm}, agents-specialists AG^{sp}, agent-facilitator ag^{fc} and other agents, presented in [8]; env is the model of the CHIMAS environment; INT is the set, described by expression (2), of the elements formalizing agents' interactions; ORG is the set of CHIMAS architectures; $\{glng,\ ontng,\ protng\}$ is the set of conceptual models of macro-level processes, where $glng$ is the model of the goal coordination by agents among themselves, ensuring cohesion at the core level of the SC; $ontng$ is the model of the agents' ontologies negotiation, corresponding to the exchange of knowledge, experience and beliefs between specialists and modeling of the VOU level of the SC; $protng$ is the model of a cohesive problem-solving protocol development by agents, which is relevant to the coordination of interaction norms at the VOU level of the SC. Due to the absence of an emotional component in CHIMAS agents, the stratum of emotional interpersonal relationships is not considered.

The set INT of elements for structuring the interactions of agents from formula (1) are described by the expression

$$INT = \{prot_{bsc}, PRC, LANG, ont_{bsc}, chn\}, \quad (2)$$

where $prot_{bsc}$ is the basic protocol that ensures the interaction of agents to form a cohesive interaction protocol to solve the problems posed to CHIMAS; PRC is the set of elements for constructing problem-solving protocol by agents-specialists and the decision-making agent; $LANG$ is the set of languages for coding agents' messages; ont_{bsc} is the basic ontology, described by expression (4), that provides agents' interpretation of the semantics of messages, when negotiating their ontologies (domain models), goals, and constructing a cohesive problem-solving protocol; chn is the degree of cohesion of agents, described by expression (6).

An agent $ag \in AG$ from formula (1) is described by the expression

$$ag = < id^{ag}, gl^{ag}, LANG^{ag}, ont^{ag}, ACT^{ag}, prot^{ag} >, \quad (3)$$

where id^{ag} is the agent identifier; gl^{ag} is the agent's fuzzy goal; $LANG^{ag} \subseteq LANG$ is the set of languages, messages in which can be read or written by the agent; ont^{ag} is the agent's ontology (domain model), described by the expression (4); ACT^{ag} is the set of actions implemented by the agent, among which for agents-specialists and the decision-making agent the goal negotiation act^{ag}_{glng}, the ontology negotiation act^{ag}_{ontng}, and the development of a cohesive problem-solving protocol act^{ag}_{protng} are distinguished, that is $\forall ag \in (AG^{sp} \cup \{ag^{dm}\})(\{act^{ag}_{glng}, act^{ag}_{ontng}, act^{ag}_{protng}\} \subset ACT^{ag})$; $prot^{ag}$ is the model of the problem-solving protocol, developed by the agent and described by expression (5).

The action $act^{ag} \in ACT^{ag}$ of an agent $ag \in AG$ is described by the expression

$$act^{ag} =< met^{ag}_{act}, it^{ag}_{act} >,$$

where met^{ag}_{act} is the method for solving the problem; it^{ag}_{act} is the intelligent technology, within which the method met^{ag}_{act} is implemented.

Thus, the CHIMAS function is described by the expressions

$$act_{chimas} = \left(\bigcup_{ag \in AG^*} ACT_{ag} \right) \cup act_{col},$$

$$\left| \bigcup_{ag \in AG} \bigcup_{act \in ACT^{ag}} it^{ag}_{act} \right| \geq 2,$$

where act_{col} is the collective function of CHIMAS, constructed by the agents dynamically in accordance with the developed problem-solving protocol; the condition requires that at least two intelligent technologies to be used [2].

The model of basic ontology ont_{bsc} from expression (2) and agent ontologies ont^{ag} from expression (3) are described by the expression [37]

$$ont =< L, C, R, AT, FC, FR, FA, H^c, H^r, INST >, \quad (4)$$

where $L = L^c \cup L^r \cup L^{at} \cup L^{va}$ is the lexicon, i.e. the set of lexemes consisting of subsets of lexemes denoting concepts L^c, relations L^r, attributes L^{at}, and their values L^{va}; C is the set of concepts; $R : \{C \times C\}$ is the set of relationships between concepts, the first component of the relationship tuple is called the domain $dm(r) = proj_1(r)$, and the second is the range of values $rn(r) = proj_2(r)$ of the relationship; $AT : C \times L^{va}$ is the set of concepts' attributes; $FC : 2^{L^c} \rightarrow 2^C$ is the function linking the lexicon with concepts; $FR : 2^{L^r} \rightarrow 2^R$ is the function linking the lexicon with relationships; $FA : L^{at} \rightarrow AT$ is the function linking the lexicon with attributes; $H^c = C \times C$ is the taxonomic hierarchy of concepts; $H^r = R \times R$ is the hierarchy of relations; $INST$ is the set of instances, each of which is a concept of a single volume or a "ground-level", specific element of a concept [38]. Functions FC and FR assume that, in general, one lexeme can correspond to several concepts or relationships, and, conversely, one concept or relationship can be described by several lexemes.

The protocol $prot^{ag}$ (3), constructed by agents from a set of elements PRC (2), defines a scheme (distributed algorithm) for the exchange of information, knowledge and coordination of agents [39]. It is described by the expression

$$prot^{ag} =< ROL, MTP, MRC, sch >, \quad (5)$$

where $ROL \subseteq C$ is the set of ontology concepts that describe the roles of agents; $MTP \subseteq C$ is the set of ontology concepts describing the types of messages transmitted by agents; MRC is the correspondence of pairs of agent roles and admissible types of messages for each pair; sch is the model of the message exchange scheme between pairs of agent roles, which determines the reaction of the agent playing the role to each type of messages, and their sequence.

When describing the message exchange scheme model sch from expression (5), the formalism of Petri nets is used [40]

$$pn =< PL, TR, IR >,$$

where $PL \subseteq C$ is the set of places; $TR \subseteq C$ is the set of transitions; $IR \subseteq (PL \times TR) \cup (TR \times PL)$ is the incidence relationship.

Messaging scheme is a multi-agent interaction protocol net (MIP-net), consisting of a set of synchronized Petri nets, which can be divided into two types: agent workflows net an (A-net) and interaction protocol net ipn (IP-net) [41]. A-net is a connected Petri net, in which there is a source-place, indicating the beginning of the process, and a sink-place, denoting the end of the process. IP-net is a Petri net, containing an input transition, before which there are no other elements of the network, a set of output transitions, after which there are no other elements of the network, as well as two disjoint subsets, the transitions of each of which are connected by synchronous communication elements $tr^{SC} \in TR^{SC}$ with transitions of the A-net, corresponding to the subset, based on multiple synchronization relationships R^{SC}.

Thus, the messaging scheme sch is a multi-agent interaction protocol network (MIP-net), defined by the following expression [41]:

$$sch^{ag} =< AN, IPN, TR^{SC}, R^{SC}, RAC, MRIPC >,$$

where $RAC \subseteq ROL \times AN$ is the mapping of the set of agent roles to the set of A-nets; $MRIPC \subseteq MRC \times IPN$ is the mapping of the correspondence of pairs of agent roles and admissible types of messages for each pair to a set of IP-nets.

IV. Evaluating agent cohesion

The key characteristic describing the CHIMAS state is the degree of cohesion of its agents

$$chn =< gls, onts, protc >, gls, onts, protc \in [0, 1], \quad (6)$$

where gls is the degree of similarity of agents' goals; $onts$ is the degree of similarity of the agents' ontologies; $protc$ is the degree of consistency of the problem-solving protocol. It is used as an optimality criterion when negotiating goals act^{ag}_{glng} and ontologies act^{ag}_{ontng}, as well as developing a problem-solving protocol act^{ag}_{protng}. In addition, it is necessary when implementing the function of the agent-facilitator to analyze the current decision-making situation in CHIMAS.

The cohesion $chn^{ag}_{i\,j}$ of a pair of agents $ag_i, ag_j \in (AG^{sp} \cup \{ag^{dm}\})$ is evaluated by calculating the corresponding components of the tuple, that is $gls^{ag}_{i\,j}$, $onts^{ag}_{i\,j}$ and $protc^{ag}_{i\,j}$. The cohesion values of each pair of CHIMAS agents form a matrix $\mathbf{CHN}^{ag}$ of tuples defined in accordance with expression (6).

As a result, the cohesion of CHIMAS as a whole is described by the expression

$$chn_{chimas} = \sum_{i=1}^{nex+1} \sum_{j=1, j \neq i}^{nex+1} \frac{chn^{ag}_{i\,j}}{(nex+1)nex}.$$

The degree of cohesion of CHIMAS as a whole is necessary for the agent-facilitator to assess the current decision-making situation and to choose the methods of influencing the agent-specialists and the decision-making agent in order to increase the efficiency of their work. Agent-facilitator have to set up collective behavior methods, that increase cohesion, when agents are disunited, i.e. have incompatible goals, ontologies and problem-solving protocol models, and decrease it, when agents are too similar to prevent conformal behavior. For this, the agent-facilitator uses a fuzzy knowledge base about the required level of cohesion, depending on the characteristics of the problem, the stage of its solution and the assessment of the current situation in CHIMAS. The fuzzy knowledge base is developed based on the results of solving problems of various classes by CHIMAS.

The model gls for evaluating the degree of similarity of agents' goals is presented in [42]. The model $onts$ for evaluating the degree of similarity of agents' ontologies is considered in [43]. The model $protc$ for evaluating the degree of consistency of problem-solving protocols is considered in the following section.

V. Evaluating problem-solving protocol consistency

To determine the similarity of problem-solving protocols, developed by different agents, it is required to calculate the similarity of each of the components of the tuple (5). For this purpose the measure of similarity of two concepts is introduced as the geometric mean of their lexicographic LSC (8) and taxonomic similarity TS (9)

$$S^C(c_k, c_m) = \sqrt{LSC(c_k, c_m)TS(c_k, c_m)}. \quad (7)$$

The lexicographic similarity of concepts LSC is determined by the expression

$$LSC(c_k, c_m) = LSL(FC^{-1}(c_k), FC^{-1}(c_m)), \quad (8)$$

where $FC^{-1} : C \to L^c$ is the function, inverse to FC, that establishes a correspondence between the concept and the lexeme describing it; LSL is the similarity of lexemes defined by the expression

$$LSL(l_k, l_m) = \max\left(0,\ 1 - \frac{ed(l_k, l_m)}{\min(|l_k|, |l_m|)}\right),$$

where ed is Levenshtein's editorial distance [44], defined as the number of characters that must be added, removed, or changed to make one lexeme from another.

To evaluate the taxonomic similarity TS of concepts, the measure is used, which is based on the upper cotopy, i.e. a set of vertices containing all the overlying vertices (superconcepts) in the taxonomic hierarchy H^{con} with respect to a given vertex and the vertex itself [45]

$$UC(c, H^c) = \{c_k \in C | H^c(c, c_k) \vee (c = c_k)\}.$$

The taxonomic measure of the concept similarity is the ratio of the number of common superconcepts of both vertices to the number of all superconcepts of both vertices

$$TS(c_k, c_m, H^c_k, H^c_m) = \\ = \frac{|FC^{-1}(UC(c_k, H^c_k)) \cap FC^{-1}(UC(c_m, H^c_m))|}{|FC^{-1}(UC(c_k, H^c_k)) \cup FC^{-1}(UC(c_m, H^c_m))|}. \quad (9)$$

Thus, to evaluate the degree of similarity of the first components of the tuples, describing the problem-solving protocol (5), namely, the sets ROL^{ag}_i, ROL^{ag}_j of concepts-roles of agents, the correspondence, based on the measure of similarity of concepts (7), is established between the roles, used in the description of the protocol by each agent,

$$MRL_{i\,j} = \{(u, v)|(u, v) \in ROL^{ag}_i \times ROL^{ag}_j \wedge \\ \wedge v = \arg\max_{v' \in C_j} S^C(u, v') \wedge u = \arg\max_{u' \in C_i} S^C(u', v)\}. \quad (10)$$

Based on the correspondence (10), the similarity of the sets of concept-roles is determined as follows

$$ROLS(ROL^{ag}_i, ROL^{ag}_j) = \left|MRL^{-1}_{i\,j}\right| * \\ * \sum_{mrl \in MRL_{i\,j}} S^C\left(proj_1(mrl),\ proj_2(mrl)\right). \quad (11)$$

To evaluate the degree of similarity of the sets MTP^{ag}_i, MTP^{ag}_j of concept-types of messages the correspondence, based on the measure of similarity of concepts (7), is established between the types of messages used in the description of the protocol by each agent,

$$MMT_{i\,j} = \{(u, v)|(u, v) \in MTP^{ag}_i \times MTP^{ag}_j \wedge \\ \wedge v = \arg\max_{v' \in C_j} S^C(u, v') \wedge u = \arg\max_{u' \in C_i} S^C(u', v)\}. \quad (12)$$

Based on the correspondence (12), the similarity of the sets of message types is determined as follows

$$MTPS(MTP^{ag}_i, MTP^{ag}_j) = \left|MMT^{-1}_{i\,j}\right| * \\ * \sum_{mmt \in MMT_{i\,j}} S^C\left(proj_1(mmt),\ proj_2(mmt)\right). \quad (13)$$

When evaluating the degree of similarity of correspondences MRC^{ag}_i, MRC^{ag}_j between pairs of agents' roles and admissible types of messages for each pair, abbreviated correspondences are preliminarily formed between compatible pairs of roles and types of messages of both agents

$$MRC^*_i = \{(t, u, v)|(t, u, v) \in MRC^{ag}_i \wedge \\ \wedge t \in proj_1(MRL_{i\,j}) \wedge u \in proj_1(MRL_{i\,j}) \wedge \\ \wedge v \in proj_1(MMT_{i\,j})\}, \quad (14)$$

$$MRC^*_j = \{(t, u, v)|(t, u, v) \in MRC^{ag}_j \wedge \\ \wedge t \in proj_2(MRL_{i\,j}) \wedge u \in proj_2(MRL_{i\,j}) \wedge \\ \wedge v \in proj_2(MMT_{i\,j})\}. \quad (15)$$

On the basis of expressions (15), the degree of similarity of correspondences MRC^{ag}_i, MRC^{ag}_j between pairs of agents' roles and admissible types of messages for each pair is determined in accordance with the expression

$$MRCS(MRC^{ag}_i, MRC^{ag}_j) = \frac{\left|MRC^*_i \cap MRC^{**}_j\right|}{\left|MRC^*_i \cup MRC^{**}_j\right|}, \quad (16)$$

where MRC^{**}_j is the correspondence (15) expressed using the most similar concepts of the agent ag_i by the formula

$$MRC^{**}_j = \{(MRL^{-1}_{i\,j}(t), MRL^{-1}_{i\,j}(u), MMT^{-1}_{i\,j}(v))| \\ (t, u, v) \in MRC^*_j\}.$$

The calculation of the degree of similarity of the message exchange scheme is performed on the basis of the notion of the transition adjacency relation (TAR) [46]. The TAR in a Petri net defines a set TAR of ordered pairs $< tr_i, tr_j >$ of transitions that can be performed one after another. Suppose there are two Petri nets pn_i, pn_j of type A or IP, for which TARs TAR_i, TAR_j are constructed in accordance with the algorithm proposed in [47], then the similarity of two nets based on the measure of similarity of their TARs is determined by the expression

$$PS(pn_i, pn_j) = |TAR_i \cap TAR_j|\,|TAR_i \cup TAR_j|^{-1}. \quad (17)$$

Based on the similarity measure of Petri nets (17), the similarity of the message exchange schemes is calculated in accordance with the following expression

$$SCHS(sch^{ag}_i, sch^{ag}_j) = \\ = \Bigg(\left(|MRL_{i\,j}|\,|MRC^*_i|\,\left|MRC^*_j\right|\right)^{-1} * \\ * \sum_{mrl \in MRL_{i\,j}} PS(proj_1(RAC_i(proj_1(mrl))), \\ proj_1(RAC_j(proj_2(mrl))))* \\ * \sum_{mrc \in MRC^*_i} PS(proj_1(MRIPC_i(mrc)), \\ proj_1(MRIPC_j(F^{MRC}_{i\,j}(mrc))))* \\ * \sum_{mrc \in MRC^*_j} PS(proj_1(MRIPC_i(F^{MRC}_{j\,i}(mrc)), \\ proj_1(MRIPC_j(mrc))) \Bigg)^{1/3}, \quad (18)$$

where $F_{i\,j}^{MRC}$, $F_{j\,i}^{MRC}$ are the functions expressing the elements of correspondences (14) and (15) of each agent using the concepts of another agent, which are calculated in accordance with the following expressions

$$F_{i\,j}^{MRC}(mrc_i^*) = (MRL_{i\,j}(proj_1(mrc_i^*)),$$
$$MRL_{i\,j}(proj_2(mrc_i^*)), MMT_{i\,j}(proj_3(mrc_i^*))),$$
$$F_{j\,i}^{MRC}(mrc_j^*) = (MRL_{i\,j}^{-1}(proj_1(mrc_j^*)),$$
$$MRL_{i\,j}^{-1}(proj_2(mrc_j^*)), MMT_{i\,j}^{-1}(proj_3(mrc_j^*))).$$

As a result, the consistency of the problem-solving protocols developed by each of the agents is determined based on expressions (11), (13), (16), and (18) as the geometric mean of the consistency of each of the components of the tuple (5)

$$prots_{i\,j}^{ag} = \big(ROLS(prot_i^{ag}, prot_j^{ag})MTPS(prot_i^{ag}, prot_j^{ag}) *$$
$$*MRCS(prot_i^{ag}, prot_j^{ag})SCHS(prot_i^{ag}, prot_j^{ag})\big)^{0.25}.$$

Thus, the calculated value of the consistency of the problem-solving protocols together with the values of the similarity of goals [42] and agent ontologies [43] form the value of the cohesion vector of agent pair (6). It is used both by agents-specialists to estimate the interaction effectiveness, and by the agent-facilitator to assess the problem-solving situation in CHIMAS. Thanks to the elements of cohesion modeling in the intelligent system, the behavior of agents is provided, which allows to overcome disagreements and avoid conflicts caused by differences in the models of the problem and the goals of its solution. As a result, by analogy with the team of specialists, CHIMAS dynamically develops a relevant solution method every time a problem is solved. Testing of CHIMAS methods is planned to be carried out on the example of the problem of planning the restoration of the power supply system [48].

VI. Conclusion

The paper substantiates the need to model cohesion processes in intelligent systems in order to reduce labor costs when integrating agents created by various teams of developers. The CHIMAS model is presented, the agents of which have mechanisms for independent, without the intervention of the system developers or its users, agreeing the goals and ontologies, as well as developing problem-solving protocol. The method for evaluating cohesion and, in particular, the detailed description of one of its parts, namely the evaluation of the consistency of problem-solving protocols developed by the agents is presented. This method makes it possible to model the cohesion of the team at two of the three levels of the stratometric concept of A.V. Petrovsky, simulating the convergence of goals, the exchange of knowledge and the development of common norms of behavior without conformity, which allows CHIMAS to more relevantly model the processes of solving problems by long-existing teams of specialists.

References

[1] V.E. Karpov, and V.B. Tarassov, “Synergetic Artificial Intelligence and Social Robotics,” in: Proceedings of the Second International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’17). Cham: Springer, 2018, pp. 3–15.

[2] A.V. Kolesnikov, I.A. Kirikov, and S.V. Listopad, Gibridnye intellektual’nye sistemy s samoorganizatsiey: koordinatsiya, soglasovannost’, spor [Hybrid intelligent systems with self-organization: coordination, consistency, dispute]. Moscow: IPI RAN, 2014. (in Russian)

[3] M. Wooldridge, and N. Jennings, “Intelligent Agents: Theory and Practice,” The Knowledge Engineering Review, vol. 10(2), pp. 115–152, 1995.

[4] M. Hajduk, M. Sukop, and M. Haun, “Multi-agent Systems—Terminology and Definitions,” in: Cognitive Multi-agent Systems. Studies in Systems, Decision and Control, vol 138, Springer, Cham, 2019, pp. 1–9.

[5] Y. Shoham, “Agent-oriented programming,” Artificial Intelligence, vol. 60(1), pp. 51–92, 1993.

[6] J. Bates, “The role of emotion in believable agents,” Communications of the ACM, vol. 37(7), pp. 122–125, 1994.

[7] Y. Sanchez-Lopez, and E. Cerezo, “Designing emotional BDI agents: Good practices and open questions,” The Knowledge Engineering Review, vol. 34, e26, 2019.

[8] S. Listopad, “Cohesive Hybrid Intelligent Multi-Agent System Architecture,” in: Proceeding of the 26th conference of FRUCT association. Helsinki: FRUCT Oy, 2020, pp. 262–269.

[9] K. Lewin, Field theory in social science: selected theoretical papers. New York: Harper, 1976.

[10] K. von Treuer, J. McLeod, M. Fuller-Tyszkiewicz, and G. Scott, “Determining the components of cohesion using the repertory grid technique,” Group Dynamics: Theory, Research, and Practice, vol. 22(2), pp. 108–128, 2018.

[11] D.R. Forsyth, Group Dynamics. Belmont, CA: Wadsworth, Cengage Learning, 2010.

[12] A.O. Savinkina, “A Test to Express Diagnostic Cohesion of Football Team,” International Scholarly and Scientific Research and Innovation, vol. 11(5), pp. 1089–1092, 2017.

[13] A.A. Cota, C.R. Evans, K.L. Dion, L. Kilik, and R.S. Longman, “The Structure of Group Cohesion,” Personality and Social Psychology Bulletin, vol. 21, pp. 572-580, 1995.

[14] J.B. Severt, and A.X. Estrada, “On the Function and Structure of Group Cohesion,” in: Team Cohesion: Advances in Psychological Theory, Methods and Practice. Research on Managing Groups and Teams, vol. 17. Bingley: Emerald Group Publishing Limited, 2015, pp. 3-24.

[15] J. McLeod, and K. von Treuer, “Towards a Cohesive Theory of Cohesion,” International Journal of Business and Social Research, vol. 3(12), pp. 1–11, 2013.

[16] B.P. Acton, M.T. Braun, and R.J. Foti, “Built for unity: assessing the impact of team composition on team cohesion trajectories,” Journal of Business and Psychology, vol. 35, pp. 751–766, 2020.

[17] A.V. Petrovskiy, “Opyt postroeniya sotsial’no-psikhologicheskoy kontseptsii gruppovoy aktivnosti” [“The experience of building a socio-psychological concept of group activity”], Voprosy psikhologii [Psychology Issues], vol. 5, pp. 3–17, 1973. (in Russian)

[18] N.V. Bakhareva, “Shkala priemlemosti kak metod izucheniya vzaimootnosheniy” [“Acceptability scale as a method of studying relationships”] in Chelovek i obshchestvo [Human and society]. Leningrad, 1975, pp. 34-37. (in Russian)

[19] A.A. Karimova, O.V. Khasanova, O.V. Shemshurenko, and A.R. Ganieva, “The Problem of Raising Group Cohesiveness in the EFL Classroom,” in: V International Forum on Teacher Education, R. Valeeva (ed.), ARPHA Proceedings 1, 2019, pp. 1025-1035.

[20] E. Brisimis, E. Bebetsos, and Ch. Krommidas, “Does group Cohesion predict team sport athletes’ satisfaction?,” Hellenic Journal of Psychology, vol. 15, pp. 108–124, 2018.

[21] C.D. McLaren, and K.S. Spink, “Team Member Communication and Perceived Cohesion in Youth Soccer,” Communication & Sport, vol. 6(1), pp. 111–125, 2018.

[22] A.J. Vanhove, and M.N. Herian, “Team Cohesion and Individual Well-Being: A Conceptual Analysis and Relational Framework,” in: Team Cohesion: Advances in Psychological Theory, Methods and Practice. Research on Managing Groups and Teams, vol. 17. Bingley: Emerald Group Publishing Limited, 2015, pp. 53–82.

[23] A.I. Dontsov, Problemy gruppovoy splochennosti [Group cohesion problems]. Moscow: Moscow State University, 1979. (in Russian)

[24] R.S. Nemov, Psikhologicheskie usloviya i kriterii effektivnosti raboty kollektiva. Moscow: Znanie, 1982. (in Russian)

[25] A.V. Sidorenkov, and R.V. Pavlenko, “GROUP PROFILE Computer Technique: A Tool for Complex Study of Small Groups,” SAGE Open, vol. 5(1), pp. 1–13, 2015, https://doi.org/10.1177/2158244015569418.

[26] A. Al-Yaaribi, and M. Kavussanu, “Teammate Prosocial and Antisocial Behaviors Predict Task Cohesion and Burnout: The Mediating Role of Affect,” Journal of Sport and Exercise Psychology, vol. 39(3), pp. 199-208, 2017.

[27] A.B. Hassan, “Explaining Cohesion Linkages in Workgroups: The Cooperative Communication in Collectivism and High Power Distance Workgroup Context,” Jurnal Komunikasi: Malaysian Journal of Communication, vol. 33, pp. 157–177, 2017.

[28] Yu.L. Neymer, “Splochennost’ kak kharakteristika pervichnogo kollektiva i ee sotsiologicheskoe izmerenie” [“Cohesion as a characteristic of the primary team and its sociological dimension”] Sotsiologicheskie issledovaniya [Sociological research], vol. 2, pp. 160 -162, 1975. (in Russian)

[29] A.Ya. Dikareva, and M.I. Mirskaya, Sotsiologiya truda [Sociology of Labor]. Moscow: Vysshaya shkola, 1989. (in Russian)

[30] N.N. Obozov, Mezhlichnostnye otnosheniya [Interpersonal relationships]. Leningrad: LGU, 1979. (in Russian)

[31] L.G. Pochebut, and V.A. Chiker, Organizatsionnaya sotsial’naya psikhologiya. Uchebnoe posobie [Organizational social psychology. Tutorial]. Saint-Petersburg: Rech’, 2002. (in Russian)

[32] L.I. Umanskiy, “Metody eksperimental’nogo issledovaniya sotsial’no-psikhologicheskikh fenomenov” [“Methods of experimental research of socio-psychological phenomena”] in Metodologiya i metody sotsial’noy psikhologii [Methodology and methods of social psychology]. Moscow: Nauka, 1977. pp. 54-71. (in Russian)

[33] V.T. Kondrashenko, and D.I. Donskoy, Obshchaya psikhoterapiya [General psychotherapy]. Minsk: Nauka i tekhnika, 1993. (in Russian)

[34] Suhendri, D.Y.P. Sugiharto, M.E. Wibowo, and Mulawarman, “Study of Conformity Behaviour and Self-Confidence From Self Control Perspective,” in: Proceedings of the 1st International Conference on Education and Social Science Research (ICESRE 2018). Dordrecht, Atlantis Press, 2019, pp. 116–118.

[35] G.S. Carlson, “Groupthink,” in: The International Encyclopedia of Political Communication, G. Mazzoleni (ed.). John Wiley and Sons, 2016, pp. 1–5, https://doi.org/10.1002/9781118541555.wbiepc026

[36] P.R. Bernthal, and C.A. Insko, “Cohesiveness without Groupthink: The Interactive Effects of Social and Task Cohesion,” Group & Organization Management, vol. 18(1), pp. 66-87, 1993.

[37] K.V. Kryukov, L.A. Pankova, V.A. Pronina, V.S. Sukhovev, and L.B. Shipilina, “Mery semanticheskoy blizosti v ontologii” [“Measures of semantic similarity in ontology”], Problemy upravleniya [Control problems], vol. 5, pp. 2–14, 2010. (in Russian)

[38] I. Dubielewicz, B. Hnatkowska, Z. Huzar, and L. Tuzinkiewicz, “Domain Modeling in the Context of Ontology,” Foundations of Computing and Decision Sciences, vol. 40(1), pp. 3–15, 2015.

[39] V.I. Gorodetskii, “Self-organization and multiagent systems: I. Models of multiagent self-organization,” Journal of Computer and System Sciences International, vol. 51, pp. 256–281, 2012.

[40] J.L. Peterson, “Petri nets” Computing Surveys, vol. 9(3), pp. 223–252, 1977.

[41] S. Ling, and S.W. Loke, “MIP-Nets: A Compositional Model of Multiagent Interaction” Lecture Notes in Computer Science, vol 2691, pp. 61-72, 2003.

[42] S. Listopad, “Estimating of the Similarity of Agents’ Goals in Cohesive Hybrid Intelligent Multi-Agent System,” in: Proceedings of the 8th International conference Fuzzy Systems, Soft Computing and Intelligent Technologies (FSSCIT-2020). Smolensk: CEUR Workshop Proceedings, 2020. pp. 180-185.

[43] S.V. Listopad, and I.A. Kirikov “Similarity measure of agents’ ontologies in cohesive hybrid intelligent multi-agent system,” Journal of Physics: Conference Series, vol. 1679 (3), pp. 032061, 2020.

[44] I.V. Levenshtein, “Binary Codes capable of correcting deletions, insertions, and reversals,” Cybernetics and Control Theory, vol. 10(8), pp. 707–710, 1966.

[45] A. Maedche, and V. Zacharias, “Clustering Ontology-Based Metadata in the Semantic Web,” Lecture Notes in Artificial Intelligence, vol. 2431, pp. 348-360, 2002.

[46] J. Pei, L. Wen, X. Ye, and A. Kumar, “Efficient Transition Adjacency Relation Computation for Process Model Similarity,” IEEE Transactions on Services Computing, pp. 1-1, 2020.

[47] H. Zha, J. Wang, L. Wen, C. Wang, and J. Sun, “A workflow net similarity measure based on transition adjacency relations,” Computers in Industry, vol. 61(5), pp. 463–471, 2010.

[48] S. Listopad, “Hybrid Intelligent Multi-agent System for Power Restoration,” in: Open Semantic Technologies for Intelligent System. OSTIS 2020, Golenkov V., Krasnoproshin V., Golovko V., Azarov E. (eds), Communications in Computer and Information Science, vol 1282. Springer, Cham, 2020. pp. 245-260.

Метод оценки сплоченности гибридной интеллектуальной многоагентной системы: согласованность протокола решения проблемы

Кириков И.А., Листопад С.В.

Работа направлена на построение нового класса систем распределенного искусственного интеллекта, а именно сплоченных гибридных интеллектуальных многоагентных систем. Концепция таких систем была предложена, чтобы упростить процессы интеграции разнородных интеллектуальных агентов, созданных различными коллективами разработчиков. Предполагается, что агенты таких систем должны быть способны самостоятельно согласовывать свои цели, модели предметной области и вырабатывать протокол решения поставленной проблемы. В настоящей статье представлен один из элементов метода оценки сплоченности агентов системы, а именно оценка согласованности протокола решения проблемы.

Received 16.05.2021

Semantization of information technology development: concepts, models and methods

Julia Rogushina
Institute of Software Systems
National Academy of Sciences of Ukraine
Kyiv, Ukraine
ladamandraka2010@gmail.com

***Abstract*—The paper discusses the conceptual aspects of the modern understanding of semantic technologies, semantic resources and semantic applications in artificial intelligence. The basic terminology used in this field is analysed. On base of this analysis definitions of the main components of semantic technologies are proposed. Main attention is paid to differentiation of elements based on semantics from other entities of intelligent information systems. Proposed approach is approved on estimation of examples of semantic software.**

***Keywords*—ontology, semantic technologies, Semantic Web**

I. Introduction

Semanticization as a direction of information technology development: concepts, models and methods The modern world exists in the information space, where a huge number of intelligent information systems (IIS) interact for solving a variety of problems. Ontology-based semantic technologies (STs) allow to build powerful applied IISs aimed on analysis and modeling of complex objects and processes of different nature. The development of such IISs is based on the results of knowledge structuring in order to construct the schemes of knowledge bases (KBs) and to define main subjects and objects of IIS functional support. The IIS intelligence depends on level of problem solving automation that is based on general and specialized knowledge about user, user current problem and adapting the problem solutions to the current state of the information environment. Such automation needs in integration of IISs knowledge developed independently for various purposes.

From the point of view of the ontological approach, the integration of different ISSs can be realized by mapping and alignment of ontologies of individual ISSs. Such operations can use an approach that combines different ontologies through a top-level ontology, or relationships can be established directly between elements of individual ontologies.

The quality of IIS integration depends on the proximity of their ontologies and quality of their mapping that are determined by the presence or absence of links, primarily information ones, between ontological classes and instances. Establishing such links is a separate complex problem based on the use of background knowledge acquired from various external information sources and from domain experts, and on evaluation and normalization of the obtained results.

Today, the semantization of information technologies (IT) is one of the basic directions of their development that has both a broad theoretical basis and significant practical results related to the development of intelligent applications in various fields. One of the most well-known projects in this field is the Semantic Web that provides a large number of standards and tools for representation and processing of information at the meaning level. Unfortunately, the popularity of this area causes a rather incorrect use of terminology: such concepts as "artificial intelligence", "knowledge processing", "semantics" are applied to various types of software and information processing methods. This determines the need to determine more formally what models of data representation and methods of information processing should be considered as semantic ones, and what information technologies that implement them are semantic, and what subset of intelligent applications use semantics. Now much attention is paid to this issue in scientific research [1] but at present a single generally accepted point of view on the basic concepts of semantic information processing is not defined.

II. Task definition

Development of semantic applications oriented on the Web needs in more accurately defined terminology that provide possibility of comparison and integration of various methods, resources and services. On base of analysis of existing approaches to definition of main components used in semantic technologies and on experience of development ontology-based applications we try to define such concepts as semantic information resources, semantic applications and semantic computing. These definitions are aimed on differentiation of semantic-based elements of IIS from the all spectrum of intelligent software and methods of intellectualizing for considering of their specifics.

III. Semantics of information systems

Semantics is a part of several scientific disciplines such as Knowledge Representation, Information Extraction, Information Retrieval, Computational Linguistics, Artificial Intelligence and Knowledge Management. Such different views imply very different views of cognition, of concepts, and of meaning. Usually, efforts related to formal semantics have involved limiting expressiveness to allow for acceptable computational characteristics.

Various syntactic structures are used in IRs for knowledge representation where these structures define semantic interpretations associated with them.

Semantics can be classified into formal (explicit) and informal (implicit) semantics. Formal semantics operate on formal notation; informal described and transmitted in natural languages (NL). The result of cognition of an abstract object is semantics; and the consequence of behavior is the semantics of execution. Implicit semantics is either present in most IRs on the Web or can be extracted from them with the help of Data Mining and Machine Learning that provide acquisition of structured knowledge or enrichment of existing structured formal representations. Such semantics can be transformed into formal one with human involvement. Formal semantics definite meaningful interpretation of data for their machine processing [2].

Semantics is a section of logic devoted to the study of the meaning of concepts and statements, as well as their formal analogues — expressions (terms and formulas) of different calculuses (formal systems) [3]. The tasks of semantics, first of all, include the clarification of the most important general concepts such as "meaning", "truth", "interpretation", "model", etc. — up to general concepts such as "set", "subject", "correspondence". A number of important semantic problems are grouped around the difference between the meanings and denotations of concepts and between the meaning and significance (truth) of statements. Properties related to the meaning of concepts and statements are called intensional, and properties related to the scope of concepts and the truth values of statements are called extensional.

We can see that a lot of intelligent problems are not semanic — for example, various logical games such as chess, mahjong or kakuro. These tasks are solved by fixed rules and don‘t use external information about context of situation. Their solving don‘t need in some knowledge about world and is based only on logical inference. From other hand, other simple intelligent games such as crosswords are semantic because they need in matching of NL definitions of pictures with words that satisfy some restrictions. For more complex tasks these distinction are less evident and for software applications can depend on realization.

In [4] semantic applications are defined as software tools that explicitly or implicitly use domain semantics to improve usability validity, and completeness. An example of a semantic application is semantic information retrieval systems that use synonyms, superclasses and subclasses of domain terms from the relevant ontology to enrich the search results by keywords. But it should be noted that almost all software uses the knowledge of developers about domain, but these knowledge can be not formalized and represented explicitly. The main advantage of knowledge-based systems is the separation of knowledge from means of their processing. The external knowledge for the Web-oriented applications is usually represented by ontologies.

IV. Ontological analysis as an instrument of semantization

Most authors associate semantic technologies with creation, use and processing of ontologies in semantic applications (SAs) [4]. The knowledge contained into the relevant ontologies should ensure the processing of natural language (NL) information, data integration and semantic search. Ontologies that can be used for these purposes differ significantly in expressiveness, volume, level of abstraction and means of representation.

Ontologies include the set of classes (concepts) and descriptions of the various relations between them, as well as the set of class individuals. Semantics of data is defined by connection with element of description of domain knowledge and meaning of this connection

Ontologies are usually divided into: 1) top-level ontologies that represent knowledge common to many domain; 2) domain ontologies that describe the features of some subject area; 3) ontologies of tasks that contain the knowledge required to run a particular software. But such division is quite conditional because domain scope depends on the problem specifics and application usually are based on some complex hierarchy of ontologies.

Ontologies are used to integrate data and knowledge, and can also become the basis for a more intelligent user interface – for example, the user communicates with the program through NL queries interpreted with the help of of ontology knowledge. However, the NL interface is not a mandatory feature of the SAs.

Source of ontologies for semantic application is one of important questions of semantic technologies. Retrieval or generation of pertinent ontology is a task for domain experts but technical features of such ontology are defined by ISS developers. For some important domains such as medicine, domain ontologies are recognized universally, but for specialized business and science sectors fitting ontologies often do not yet exist and need to be created. More often we have situation that some ontology is developed for sufficiently near domain but it needs in some reduction or expansion. An important issue in the application of semantic technologies is finding the pertinent domain ontology [5]. Although a number of ontology engineering methodologies developed in an academic context are widely used for various practical problems and are tested in the context of real-world or corporate applications.

Thus, there is a problem of matching the user task solved by IIS and a set of existing ontologies or information resources (IRs) that can be used for generation of domain ontologies (for example, semantically marked IRs).

A. IRs used in semantic technologies

Effectiveness of STs depends of quality, relevance and actuality of information that is processed by SAs. Important factors of IR selection are their structuring and semantization, use of generally accepted standards and languages, etc.

Now the main part of information for IISs is provided by the Web. The current Web is primarily a very large number of hyperlinked documents. Part of them designed for human reading is represented in HTML or more controlled XHTML formats. But much of information relating to real-world or abstract notions and the relationships between them is stored in relational and quasi-relational SQL databases. Information processed by IIS can be trusted, dynamic, transparent, user-friendly [6].

Semantic information resource (SIR) is a term that needs in more detail definition. Every IR is a set of one ore more documents that can be stored and used by ISs. IR contains some information (at least its name and size) that can be interpreted into some meaning. But in sphere of semantic technologies SRs correspond to some non-empty subset of all available data and IRs.

SIRs can use various terminological and lexical resources, KBs, ontologies and other SRs. Such SRs include some formal semantic components that define relations between content and formal semantic representation. The paper [7] analyzes the approach to information processing at the semantic level oriented on processing of NL. Various researchers combine term “semantic resources” with lexics, annotations, thesauri, etc. Researchers analyse semantics of NL entities on base of NL documents and single out such three components of ST: 1) ontologies; 2) models of NL entities; and 3) semantic IRs.

In our research we have to process different types of information objects (IOs) (their composition is defined by problem) of different levels of complexity, where NL-entities represent only one of the elements along with multimedia IOs and structured IOs. Information about individuals of IOs them can be contained into heterogeneous IRs – NL texts, multimedia (video, audio, images), structured and semi-structured components and links between them. Knowledge about IO structure and relations can be represented by ontologies too but we can consider other representations (for example, rules, decision trees or semantic networks).

IRs used by STs in general can be described by combination of three main components:

- KBs (external and internal) that define main concepts, their features and relations;
- IO models that define structure of typical elements processed by application;
- IRs that contain information about these IOs (explicitly or implicitly).

KBs represent the upper level of abstraction of the application knowledge structure. For example, domain ontology define the set of concepts that can be used by application, their possible and illegal relations and parameters. Expressiveness of knowledge representation is defined by selected formal language. IO models represents the structure and elements of those typical elements that are processed by application in terms of selected KB. For example, ontology classes can become the base of corresponding typical IO.

Level of IRs identifies what types of sources can process IS to take information about individuals of IOs. For semantic IRs such information can be extracted automatically. Structures but not semantic IRs need in explicit linking of data fields with IO properties. Non-structured IRs require various specialized means of processing that depend on IR data (for example, methods of image recognition for pictures, speech recognition for audio and text recognition for scan copies of text documents, NL processing for text documents). Semantization of IRs reduces the computational complexity of algorithms used by SAs and provides processing of the task-specific aspects of data only.

If we consider these components on the example of Wiki resource based on Semantic MediaWiki, an ontology provides structure and concepts of domain knowledge for semantic markup terms, IO models are represented by templates of typical IOs, and semantic IRs are individual Wiki pages and arbitrary sets of such Wiki pages. It should be noted that in this case not only NL content is used, but also multimedia IOs and their metadata.

Different types of IRs such as taxonomies, vocabularies, thesauri or ontologies can be used as SRs if they have means of their simultaneous use [8], and their semantic interoperability consists in preventing problems of misunderstandings between users by taking into account the semantics associated to the data, and ensuring exchanged information share the same meaning. SRs can be heterogeneous and their analysis can be implemented at different levels. For NL such levels are syntax (representation format), structure (data organization) and semantics (different points of view).

The Semantic Web project proposes a set of languages for knowledge representation such as Resource Description Framework (RDF) [9] and Web Ontology Language

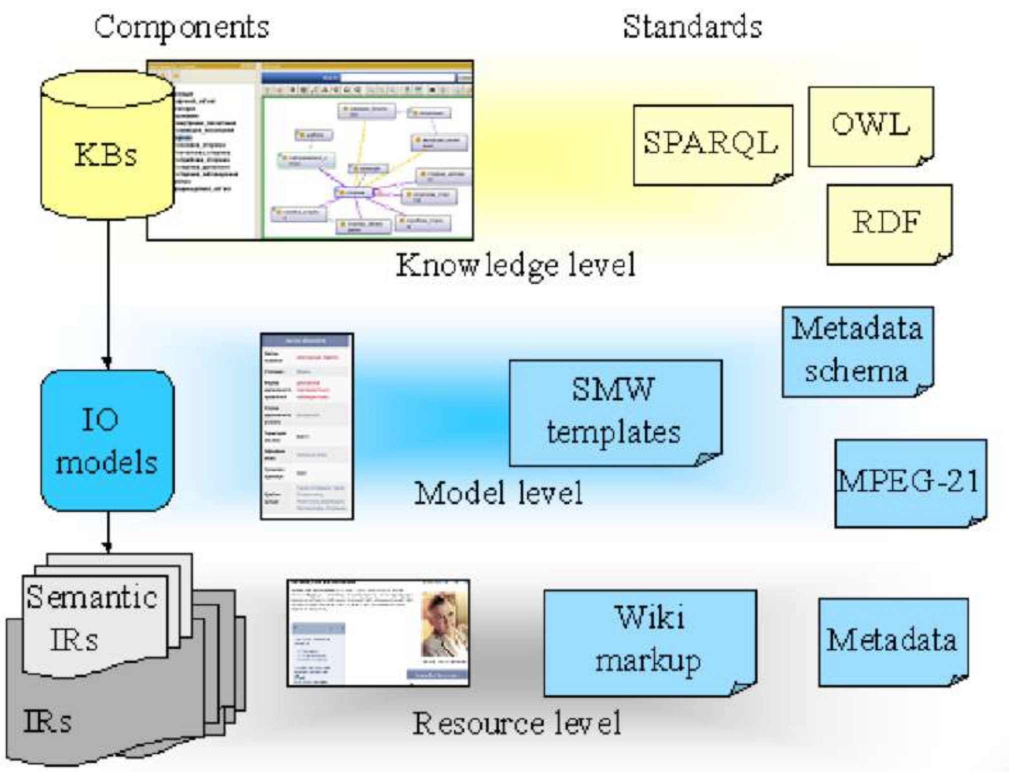


Figure 1. IRs used in semantic technologies.

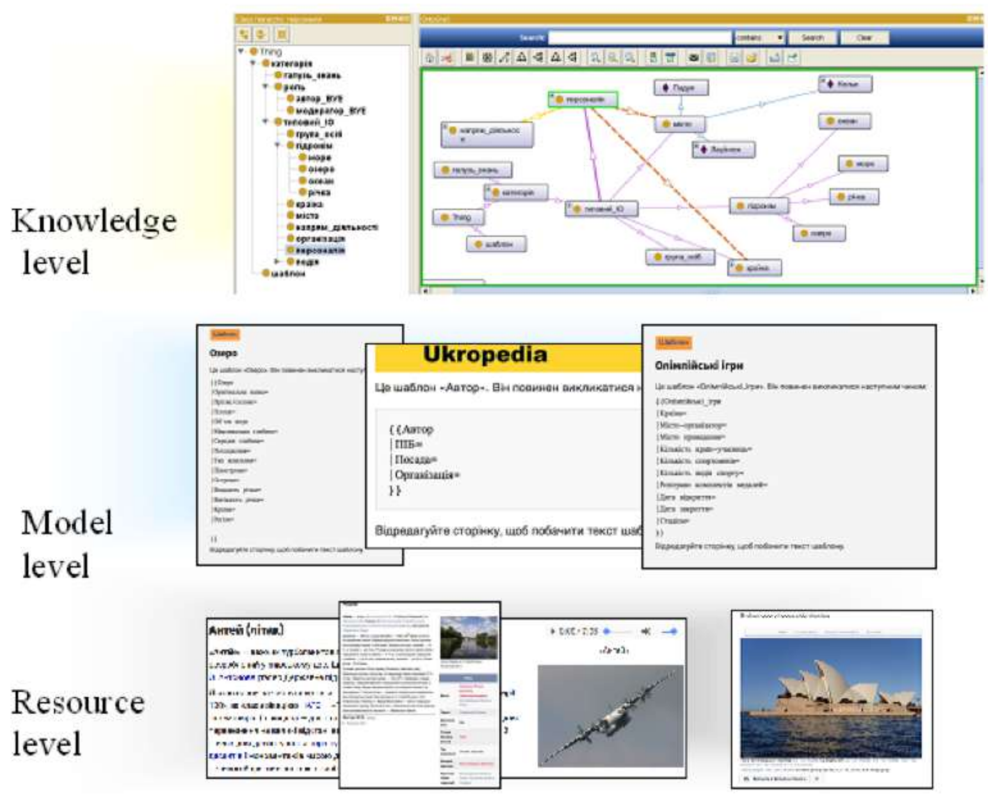


Figure 2. IRs used in semantic technologies.

(OWL) [10]. Every IR specified by means of RDF or OWL is SR.

Some researchers base definition of SR on context [11] where context of IO (i.e. an object, an event or a process) is considered as a collection of semantic situational information that characterizes features of this IO and its relations with other IOs. Contextual information can include metadata of IRs, links with other knowledge descriptions, models of IOs, etc. Examples of SRs that use different types of knowledge are considered in [7][xx] : Wiki and Web resources accessed by information retrieval systems (IPS). But this work doesn't propose definition of SRs.

In this study, we distinguish the following types of IRs that can be used by SA: SRs -– IRs with elements (content and/or metadata) that contain elements of semantic markup for clear connection of the IR fragments with domain concepts formalized by some knowledge representation means; non-semantic IRs -– all other IRs where information is not explicitly related to domain concepts (but such relationships can be determined by various means of analysis such as Text mining).

The main difference between SRs and non-semantic IRs deals with unambiguous interpretation of their semantics with the help of external standards and KBs. SRs have clear references to the formally described knowledge about the semantics of their structuring -– ontology, taxonomy, thesaurus, Examples of SRs are metadata of the Dublin Core standard, semantic Wiki resources based on Semantic MediaWiki, multimedia IRs with metadata in the MPEG-21 standard. etc.

V. Models of semantics

Models of semantics (MSs) depend on IO and IR specifics and on modelling approaches. Various models are oriented on some type of information (NL text, multimedia, images, structured data, etc.). Such models reflect the specific features of information type. Models of semantics of NL-entities in [7] contain models of semantic primitives, texts and their comparison. They have various expressiveness that depends on IIS purposes. For example, NL text can be represented as a semantic net that connect all words for every simple or complex sentence on base of syntax analysis but the same text can be represented as a set of links between sentences.

Other MSs are oriented on some fixed type of IOs (or group of connected IOs). Examples of such MSs are user profiles used by learning SAs that can be exported from one IIS to another by matching of corresponding ontologies [12]. Even the most typical characteristics of the users can be modelled by user profiles with various terms and categories that causes need in standardization of user profiling.

The examples of these standards are DCMI has several elements for defining different categories (roles) of users with respect to a document(Dublin Core Metadata Initiative and FOAF (Friend of a Friend) -– an RDF-based general-purpose model for description of users on the Web. Many of them use ontological representation of knowledge. For example, FOAF is a lightweight ontology aimed at creating an annotated network of homepages for people, groups, companies, etc. It is implemented in RDF Schema and contains such basic classes as agent, person, organization, group, project, document, image, as well as some basic properties of instances of these classes.

Now a lot of standards are developed for representation of metadata for various types of IRs. They can be considered as SRs too. For example, Dublin Core is an example of a lightweight ontology that is used to specify the characteristics of electronic documents. Now this ontology is most widely applied for metadata semantization.

An other example of universal MSs are templates from Semantic MediaWiki that formalize the structure, categories and semantic properties of Wiki page with arbitrary content. Use of ontologies in creation of MSs makes them more interoperable and simplifies integration of different modelling approaches.

To model IOs with complex structure we can use models of simple and complex IOs (CIO), as well as methods of CIO. CIOs are formed by the meaningful relations between simpler IOs. For example, CIO "Family" can be formed by relations "child-parent" and "married" applied to CIOs "Person" and by relation "address" for CIO "place of residence".

On base of above analysis we propose following definition: SIR as IR where elements of content or IR in whole are connected explicitly with elements of some formalized representation of knowledge with unambiguous semantic interpretation. Examples of such SIRs are: 1) IRs with semantic markup where ontology classes are used as markup tags; 2) documents with metadata based on standards defined by ontologies.

VI. Algorithms used in semantic technologies

Structuring of information is one of the ST tasks that helps in selection of important for user attributes. Such attributes can be considered as information factors acquired from data by various machine learning (ML) algorithms, means of Data Mining and other elements of artificial intelligence (AI) such as pattern recognition and logical inference.

Instruments that are used for analysis of SR content from the point of view of Data mining depend on specifics of these resources, means of their semantization and goals of analysis. For example, NL documents can be processed by Text Mining methods.

VII. Semantic computing and semantic applications

Semantic computing (SC) is a term used to describe a set of methods, algorithms, and software used to process data based on their semantics, which are uniquely defined and interpreted. This research direction is based on success in three areas:

- methods and means of practical engineering of ontologies as structures for integration and representation of heterogeneous distributed knowledge and data that make knowledge and data equally accessible to both man and computer;
- use and retrieval of the Web IRs with varying degrees of structuring as a universal source of knowledge about the meaning of concepts, words and other entities;
- methods, algorithms and tools for processing and analysis of large amounts of data [13].

SC combines different disciplines — multimedia computing, Semantic Web, soft computing [14], Cognitive calculations, computational intelligence, computational linguistics, etc. The use of SC allows to link informal intentions of people with content that may contain structured and semi-structured data, multimedia, natural language text, programs, services, and so on. Use of SC helps to disseminate traditional IT of character processing and content syntax in the direction of knowledge processing.

SAs are those software tools where the components of formal models at the conceptual level are described by formal knowledge representation, for example, by set of concepts and relations of pertinent ontology, and data transformation processes are performed with use of semantic computing. Some authors [xx] associate SC only with NL processing, for example, in semantic clustering and classification problems. However, although methods for establishing the semantics of text entities, their comparison with the use of semantic proximity measures are used in many SAs, but there is a large number of other SAs that process structured information, multimedia, etc.

In [7] semantic applications are defined as software that explicitly or implicitly use the domain semantics. But such definition does not make it possible to divide arbitrary software into semantic and non-semantic because implicit domain semantics is used in some way by almost any software. The main criterion for identifying software as a SA as a separation of KB from the means of knowledge processing, i.e. clear representation of both semantics.Therefore, we propose to use the term "semantic application" only for software where domain knowledge is separated from the built-in knowledge of IIS and can be changed independently of the IIS itself. Applications designed to solve problems that traditionally belong to the field of artificial intelligence (AI) but with knowledge that fully integrated into IIS and can not be changed without software transformation, can be considered intelligent or intellectualized, but are beyond the scope of this research and are not considered as semantic ones. The central component of any SA is KB that contains knowledge about task. Interpretation of this knowledge allows to obtain results that the user needs and that can not be obtained without the use of this knowledge (or which lack requires much more time and calculations).

VIII. Use of the Semantic Web for semantics applications

Recently, developers of distributed IISs exhibit a tendency to transition from the use of relational databases to ontological knowledge bases (KBs). This process causes semantization of IISs and their transformation to SAs with differentiation of interoperable knowledge and formally described means of their processing on base of common standards.

The Semantic Web [15] is a project that aims to transform the Web information space into a distributed KB and to ensure the interoperability of knowledge representation. These goals require the use of generally accepted standards in SAs for the languages of knowledge representation and requests for them. For example, semantic search can use ontologies that characterize the user's area of interest or describe his profile, and such ontologies can be selected (by the user or developers) regardless of the implementation of search and mapping algorithms from any external repositories of ontologies. Main components of the Semantic Web

The Semantic Web conception is based on tree main elements -– ontologies [16] for knowledge representation, Web-services [17]for representation of knowledge processing means and software agents [18] that can activate Web-services for knowledge processing for the benefit of users. In SAs ontologies are used for formal modeling of the system structure, i.e. they define the relevant objects, subjects of domain and relations between them. Now domain ontologies are usually represented by the OWL [10] language developed by the Semantic Web initiative that is an add-on to the RDF language [11]. RDF provides information in the form of an oriented marked graph. The basic elements of RDF are triplets <subject, predicate, object>.

However, SAs are not equivalent to IISs based on the Semantic Web standards: now many IISs use ontologies as KBs but their usage is not a prerequisite for IIS semantization. IIS can be based on other formalisms of knowledge formalization which for some reasons better meet the domain needs or already are accumulated in previously created IISs and KBs.

SC is a computational methodology and computational technology with machined descriptions of content and intentions proposed by [12] that provides communication between the content of IRs and the user based on:

- semantic analysis of content for transformation of arbitrary IRs into SIRs with semantically described content;
- semantic integration of knowledge from content of different SIRs with unified model;
- semantic services (for example, Web-services, semantic search engines) and means or their integration oriented on user tasks;
- semantic user interface that process NL requests provide user-friendly representation of processing results (visualization, structuring).

SC models and implements computational structures and behavior at the level of semantics or knowledge that exceeds the level of symbolic data with use of such categories as “to be”, “to have” and “to do”.

The semantics of "being" defines the meaning of the equivalent relationship between unknown and known entities or concepts. The semantics of "mother" gives meaning to the structure or compound essence. The semantics of "do" gives meaning to the action or behavior of a system or person.

Formalization of semantic computations is considered in [20] Is defined in terms of objects, their attributes, relationships and content.

This ontology represents main components of semantic technologies and relations between these components. Ontology can be expanded by links to external ontologies that define more precisely some particular aspects of this research sphere. For example, “inductive inference” can be specified by ontology of inductive modeling algorithms that by-turn can be supplemented with ontology of by The Group Method of Data Handling (GMDH) linked with individuals of methods, their properties and software realizations [21].

IX. Conclusion and future work

To test the correctness of proposed definitions we apply them to various ISSs and IRs that we develop. Such decision is explained by reason that we don‘t know exactly all characteristics of IRs and ISS of other developers and don‘t sure of meanings of concepts in their manuals.

We consider three IISs: e-learning system M(e)L for distant control of student skills by formal model of domain knowledge [22], information retrieval system MAIPS [23] oriented on personified user needs and advisory system AdvisOnt [24]. All these IISs use external ontologies (in OWL) that can be changed by any others according to changes of user needs without any changes of software and apply ontological knowledge for processing of user requests. Therefore we can designate them SAs. We also consider some IRs on base of Semantic MediaWiki (for example, portal version of Great Ukrainian Encyclopedia e-VUE) where we take part in development of the KB structure [25]. Markup of Wiki-pages is based on terms from pertinent ontology, and content elements are connected explicitly with ontology classes by semantic properties connected with ontology relations. Therefore we can consider this IRs as SIR.

References

[1] O. Lassila, D. McGuinness, D. "The role of frame-based representation on the semantic web", In Linköping Electronic Articles in Computer and Information Science, 6 (5), 2001.

[2] A. Sheth, C. Ramakrishnan, C. Thomas, C. "Semantics for the Semantic Web: The implicit, the formal and the powerful", In International Journal on Semantic Web and Information Systems (IJSWIS), 2005, 1(1), 1-18.

[3] o.V. Palagin, M.G. Petrenko "Explanatory ontographic dictionary on knowledge engineering", Kyiv: LLC NVP Interservice, 2017, 478 p (in Ukrainian)

[4] W. Bartussek, H. Bense, T. Hoppe, B.G. Humm, A. Reibold, U. Schade, M. Siegel, P. Walsh "Introduction to Semantic Applications", In Semantic Applications. Methodology, Technology, Corporate Use. Springer-Verlag GmbH Germany, part of Springer Nature 2018.

[5] T. Hoppe, B. G. Humm, A. Reibold "Semantic Applications - Methodology, Technology, Corporate Use", Springer Verlag, Berlin, 2018, pp. 1-12.

[6] A. Sheth, M.D. Lytras "Semantic Web-Based Information Systems: State-of-the-Art", 2006, IGI Global.

[7] V.I.Gorodetsky, O.N. Tushkanova "Semantic technologies for semantic applications. Part 1. Basic components of semantic technologies", In Artificial Intelligence and Decision Making, 2018, N 4, pp.61-71.

[8] C. F.Da Silva, L. Médini,S.A. Ghafour, P. Hoffmann, P. Ghodous, C. Lima, "Semantic interoperability of heterogeneous semantic resources", In Electronic Notes in Theoretical Computer Science, 2006, 150(2), 71-85.

[9] O. Lassila, R. Swick "Resource Description Framework (RDF) Model and Syntax Specification", W3C Recommendation. URL: http://www.w3.org/TR/REC-rdf-syntax.

[10] "OWL Web Ontology Language. Overview", W3C Recommendation: W3C, 2009. http://www.w3.org/TR/owl-features/.

[11] W. Huang, D. Webster "Enabling context-aware agents to understand semantic resources on the WWW and the Semantic Web", In IEEE/WIC/ACM International Conference on Web Intelligence (WI’04), 2004, pp. 138-144.

[12] J. Rogushina "User Profile Ontological Modelling by Wiki-Based Semantic Similarity", In International Conference on Open Semantic Technologies for Intelligent Systems OSTIS, 2020, pp 198-211.

[13] Y. Wang "On formal and cognitive semantics for semantic computing", In International Journal of Semantic Computing, 2010, 4 (02), pp.203-237.

[14] L.A. Zadeh "Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information", In intelligent systems. Soft computing, 1998, 2 (1), pp.23-25.

[15] Davies J., Fensel D., van Harmelen F. Towards the Semantic Web: Ontology-driven knowledge management. John Wiley Sons Ltd. England. 2002. 288 p.

[16] L. Obrst, W.Ceusters, I. Mani, S. Ray, B. Smith "The evaluation of ontologies Semantic Web", Springer US, 2007. 139–158.URL: http://philpapers.org/archive/OBRTEO-6.pdf/

[17] R. Studer, S. Grimm, A. Abecker "Semantic web services", 2007, Springer.

[18] J. Hendler "Agents and the semantic web", In IEEE Intelligent systems, 2001, 16(2), pp.30-37.

[19] P. C. Y Sheu "Semantic computing", In Semantic computing, 2010, pp.1-9.

[20] Y. Wang "Software engineering foundations: A software science perspective", 2007, CRC Press.

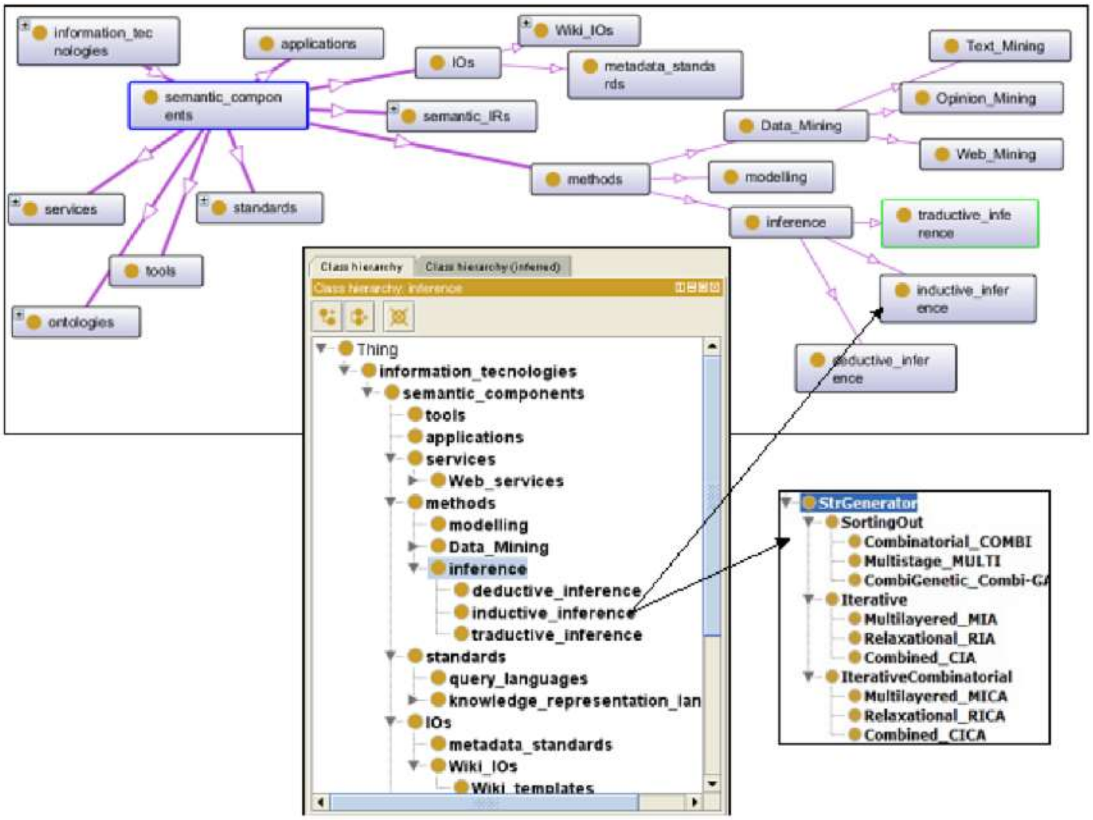


Figure 3. IRs used in semantic technologies.

[21] H. Pidnebesna, V. Stepashko "On construction of inductive modeling ontology as a metamodel of the subject field", 2018.

[22] A.Gladun, J. Rogushina "Distant control of student skills by formal model of domain knowledge", International Journal of Innovation and Learning, 7(4), pp.394-411.

[23] A.Gladun, J. Rogushina "Use of Semantic Web technologies in design of informational retrieval systems", In Building and Environment, pp.89-103.

[24] S. M. Pryima, O. V. Strokan, J. V. Rogushina, A. Y. Gladun, D. V. Lubko, V. M. Malkina "Ontological Analysis of Outcomes of Non-formal and Informal Learning for Agro-Advisory System AdvisOnt", In International Conference on Technologies and Innovation, 2020, pp. 3-17, Springer, Cham.

[25] J. Rogushina "Semantic Wiki resources and their use for the construction of personalized ontologies", In CEUR Workshop Proceedings 1631, 2016, pp.188-195.

Семантизация развития информационных технологий: понятия, модели и методы

Рогушина Ю.В.

Рассматриваются концептуальные аспекты анализа современных семантических технологий, семантических ресурсов и семантических приложений. Предлагаются определения основных компонентов, используемых в семантических технологиях, и анализируются связи между ними. Основное внимание уделяется отличиям между елементами информационных технологий, базируюзихся на семантике, от других элементов интеллектуальных систем.Предлагается онтологическая модель компонентнов семантических технологий.

Онтологический анализ рассматривается как один из наиболее широко используемых инструментов семантизации. Анализируются проблемы использования онтологий в различных интеллектуальных системах и специфика их применения в семантических приложениях. Рассматриваются примеры семантических ресурсов Веб и их связь с онтологиями (в частности, семантические Вики-ресурсы).

Received 29.05.2021

Development of advisory system based on semantic technologies

Julia Rogushina
Institute of Software Systems
National Academy of Sciences of Ukraine
Kyiv, Ukraine
ladamandraka2010@gmail.com

Anatoly Gladun
International Research
and Training Center
of ITS of NASU and MESU
Kyiv, Ukraine
glanat@yahoo.com

Serhii Pryima, Oksana Strokan
Dmytro Motorny Tavriya State
Agrotechnological University
Melitopol, Ukraine
pryima.serhii@gmail.com,
oksana.strokan@tsatu.edu.ua

***Abstract*—We analyze main characteristics of modern semantic technologies and consider specifics of their components used for development of the Web-oriented advisory systems. Ontological analysis is used for representation of distributed background knowledge about users, their professions, competencies, lifelong learning ouecomes, etc., and standards of the Semantic Web project provide the technological foundation for creation of intelligent advisory applications.Validation of learning outcomes of informal and non-formal learning needs in semantic matching of user profiles with classifications of professions and qualifications.**

On base of this research we propose models and means for representation, acquisition and processing of knowledge that make such systems more efficient and dynamic. Methods for comparison of different advisory information objects are based on atomic competences processing are developed. Elements of artificial intelligence and machine learning are used for construction of information objects that are processed by advisory systems.

Advantages of proposed approach are demonstrated on example of applied information system AdvisOnt developed to combine the market of educational services with the labor market that can use validation of the informal and nonformal learning outcomes.

***Keywords*—ontology, semantic technologies, advizory system**

I. Introduction

The modern world exists in the information space, where a huge number of intelligent information systems (ISs) interact for solving a variety of problems. Over the last years the Semantic Web and associated semantic technologies have moved from the domains of research and standards committees to the mainstream industry. Development of the Web-oriented ISs shows a general trend caused by growth of data volumes: transition from the processing and storage of large amounts of data to the processing and storage of more compact knowledge with a much more complex structure based on various methods, paradigms and technological solutions (such research directions as Big Data, Internet of Things, Web Mining). All of them apply semantization for different steps of information processing and try to take into account meaning of data and relations with domain knowledge.

Ontology-based semantic technologies allow to build powerful applied intelligent applications aimed on analysis and modeling of complex objects and processes of different nature. Development of such intelligent ISs (IISs) is based on the results of knowledge structuring in order to construct the schemas of knowledge bases and to define main subjects and objects of IIS functional support. The intelligence of such ISs is related to automation level of problem solving by use of general and specialized knowledge about user, user problem and adapting the solutions to the current state of the information environment.

One of the most well-known projects of semantic technologies is the Semantic Web [1] aimed at is to transform the Web into global knowledge base. It provides a large number of standards and tools for representation and processing information at the knowledge level. Semantic Web is a powerful instrument for improving the efficiency of distributed and shared access to information and its use by intelligent applications [2]. The main components of the Semantic Web are ontologies [3] for representation of knowledge, Web services [4] for knowledge processing and software agents to represent individual needs of users [5]. This semantic technology is supported by open standards that allow to formalize the semantics of information resources (IRs) and software tools for their search and processing: metadata description language RDF [6]; ontology representation language OWL [7]; query language SPARQL [8] for RDF and OWL.

Another interesting example of semantic technology is OSTIS (Open Semantic Technology for Intelligent Systems) oriented on design of knowledge-driven IISs [9]. The basis of OSTIS technology is the semantic computer code — the standard of semantic representation of information in the IS memory. OSTIS provides integration of heterogeneous applied artificial intelligence systems based on open semantics and unifies knowledge representation, machine learning, creation of algorithms and methods for data processing. It aims formation of intelligent environments for human life (Internet of Things, smart home, smart highways, Smart City, etc.). This semantic technology can be used in design of various ISs: intelligent control systems, reference (consulting, advisory) systems, interactive textbooks and training systems, processing of the Web of Things, etc.

Use of knowledge representation and processing methods depends on specifics of solved tasks. In addition, domain of task defines sources of domain knowledge and characterics of these sources. Semantic applications [10] are a subset of intelligent ISs. In these work we analyze the specifics of semantic ISs, their components and features and demonstrate advanteges of semantic technologies on example of he Web-oriented advisory system AdvisOnt.

For this purpose we consider tasks dealt with combining of the market of educational services with the labor market that can solve problems caused by informal and non-formal learning. Therefore we take into account specifics of educational domain, its subjects and objects and their relations into the labor market. We consider open sources of information about them and methods of their acquisitions and matching.

II. Problem definition

We analyse main elements of semantic technologies aimed on more efficient development of the Web-oriented intelligent applications. Ontology-based approach provides a lot of advanages in development of applied software but it makes calculations much more complex and cumbersome. Therefore every practical task needs in analysis of expedience of all components of semantic technologies.

This problem we consider on example of advisory system AdvisOnt that provides automated semantic methods for matching of qualifications and competences of various information objects (IOs) – humans, organizations, learning courses, requirements of employer etc. Ontology of competencies and methods for processing of atomic competences become the ground for integration of such objects described by different terms from various qualification systems. Therefore this system needs in external knowledge sources and in non-trivial methods of their processing, and use of the Semantic Web standards supports these possibilities that can‘t be achieved in other ways.

III. Lifelong learning and validation of its results

Lifelong learning is a key factor in personal and professional development of human [11]. Validation of the results achieved in the process of non-formal and informal learning (knowledge, skills, competencies, etc.) with use of open educational resources is necessary for access to the labor market and lifelong learning [12]. Employer organizations, trade unions, chambers of commerce and industry, national bodies involved in the recognition of professional qualifications, employment services, youth organizations, education providers, and civil society organizations are the most interested parties in providing opportunities for the recognition of non-formal and informal learning.

Official recognition of non-formal and informal learning outcomes obtained outside formal learning systems of partial qualifications has to take into account all outcomes obtained by persons in process of lifelong learning. Information about these outcomes can be proposed by person or be acquired from various external sources. The procedure of recognition consists of the following mandatory steps:

1) submission of documents that directly (such as educational declaration) or indirectly (such as reports, articles, descriptions of projects etc.) certify learning outcomes of person;
2) creation of group that can determine the possibility of forms and terms of attestation for recognition of learning outcomes acquired in non-formal and informal education;
3) attestation of these learning outcomes recognition.

After validation and formalization oriented on recognition of the results of previous formal education such outcomes are integrated with results of formal education to transform all competencies of person into some interoperable representation. All stages of this process need in external knowledge about learning domain that helps to define meaningful relations between elements of learning outcomes.

Learning outcomes of some person consist of his/her knowledge, skills, abilities and competencies. Recognition of such outcomes that is achieved through non-formal and informal (spontaneous) learning with the help of various open educational resources is necessary for semantic matching of resumes with vacancies of labor market and propositions of learning organizations.

Rapid expansion of the information technologies, data storage and software for the analysis of Big Data and the Web resources sets fundamentally change the ways of information interchange on the labor market. Subjects of the labor market have possibilities to describe their proposals and requirements through various informal characteristics which are often intangible (e.g. team spirit, social skills, leadership skills). Use of different terms to describe such characteristics actualizes the problem of comparing the semantics of such descriptions.

Validation of learning outcomes is aimed on recognition of learning outcomes that involves confirmation by the competent authority of the fact that such outcomes obtained by a person are evaluated according to certain criteria and meet the requirements of the standard. Validation allows the recognition of learning outcomes obtained outside the institutions of formal learning (in non-formal and informal education) and is necessary for access to the labor market and lifelong learning [13].

The tools used in this process have to take into account changes in the open world and be dynamic, and they need in semantic retrieval components based on ontological models of user and domain [14].

IV. Semantic technologies

Semantic technologies that are aimed at knowledge-level processing of information can solve this problem because they are able to formalize, analyze and process the content of information resources (IRs). Now these technologies based on the analysis of domain knowledge and personal information about users are widely used in development of various distributed applications.

Semantics allows to define explicitly meanings and relations between domain concepts represented by data (words, phrases, symbols, etc.) that depends on context. For the same piece of information, semantics can be defined differently depending on ontology used to formalize the user's view of the world. Ontologies can be used as a formal, explicit specification of conceptualization of terms at a certain level of details.

Semantic processing of information includes: methods and means of integration and unified representation of heterogeneous distributed knowledge and data; retrieval and processing the Web resources as an universal knowledge source of about meaning of information objects.

What technologies and IRs can be named "semantic" at present is an open question that has not common unified definition. Such situation is caused by specifics of domains where various knowledge-based models and methods are applied. Some researchers establish linkage between semantic technologies and processing of natural language, and others – with ontological analysis. In this work we propose to use term "semantic technologies" for such technologies of information processing that distinguish external IRs as sources of knowledge and methods used for automated analysis of this information.

One of the results of such processing is an achievement of semantic compatibility of open educational IRs that allows to use and integrate information about results of non-formal and informal learning from different sources and databases in various ISs.

Semantic technologies in general can be described through a combination of three main components [15]: ontologies; semantic resources; models of semantics of NL-entities.

But this approach is concentrated on processing of natural language (NL) IRs at the semantic level. In our study we take into account more wide classes of IOs with various structure defined by appropriate ontologies. For example, advisory systems analyze people, organizations, vacancies, learning courses, etc. that cam contain NL definitions, multimedia elements and structured data.

Therefore from the point of view of creating semantic applications, these three basic components form a hierarchy where:

- ontologies and other knowledge bases (KBs) are the upper level of abstraction of the knowledge structure;
- IO models represent the intermediary level that allows to distinguish typical IOs and their properties and characteristics;
- semantic IRs are the lower level that provides information about individuals of classes.

Semantic IRs can have links between content elements (IOs of various types and structure) and with elements of IO models (for example, links with other Wiki pages or with data). Meaning of links is provided by means specific for IR representation and markup. For example, Semantic MediaWiki uses semantic properties.

Structure of IO models can contain relations with other IOs (for example, some IO of category "Person" has semantic link with IO of category "Organizations" by relation "Place of work", and such link can be used into page content only if is present into the IO model). IO models can be formalized by various representations such as templates and forms.

Domain ontologies contain classes and individuals of concepts and formalize their properties and characteristics. Hierarchy of components used by semantic applications is demonstrated by Fig. 1.

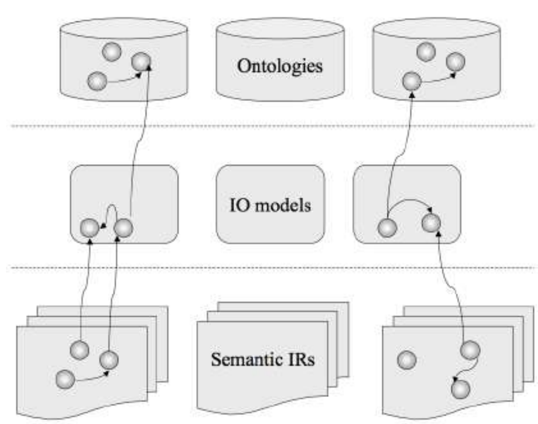


Figure 1. Hierarchy of components of semantic applications.

Semantic IRs explicitly define relations between content elements and formalized knowledge representations. Usually they use some common standards (such as MPEG21 for multimedia) or problem-specific domain ontologies.

Semantic Wiki resources are Wiki IRs with underlying model of the knowledge described in its pages. Widely used semantic markup of the Web resources is realized by various intelligent extensions of the Wiki resources (such as Semantic MediaWiki, OntoWiki, IkeWiki, SemanticXWiki, and KawaWiki). There is a number of Wiki software that provides semantic functionality. Some of them are standalone Wiki applications, and others are realized as extensions or plugging to standard Wiki software. Semantic Wiki-based IRs differ in their degree of formalization. Some of them support integration with external ontologies (RDF and OWL) and can generate local ontologies for group of the Wiki pages. For example, Semantic Media Wiki provides to users such tools of semantic structuring as categories and semantic properties. Categories help to link Wiki pages with more general terms and group them, and semantic properties allow to define various semantic features and their values of concept linked with some page. Categories and semantic properties of the Wiki pages can be used as classes and object properties of domain ontology, and names of Wiki-pages – as individuals of ontology. Such domain ontology can be built automatically by special functions of Semantic Media Wiki or by special algorithms according to personal needs of users. Unfortunately, there is no logical or semantic restrictions on ontology building in Semantic Media Wiki. Therefore ontologies provided such possibilities remain the important content of semantic technologies as a source of domain knowledge

V. Ontologies in semantic technologies

Ontological analysis is widely used now for formal modeling of various domains [16]. Ontology provides a formal explicit description of domain concepts (classes and individuals), their properties, attributes and relations. Moreover, ontology can contain some domain-specific restrictions on use of all these elements and their combinations. Modern intelligent applications use ontologies as interoperable knowledge bases [17]. where information acquired by one ISS can be reused by other one.

The most high-usage languages of ontology representation are dialects of OWL (Ontology Web Language) that differ by their complexity and expressiveness: from OWL Lite to OWL Full. The most popular of them is OWL-DL based on descriptive logics Ontology built on OWL is a sequence of axioms and facts with the addition of references to other ontologies associated with it.

A lot of oriented on the Semantic Web software systems use ontologies as a base of domain knowledge for semantic markup of various documents (NL texts, Wiki resources, other semi-structures and structured texts, multimedia context etc.). Popular representation of information in Wiki resources can be semantizied in this way. Advantages of the use of ontologies for semantic representation for learning domain and competencies ([18], [19]) are analyzed by many researchers.

In order to support the process of validation of learning outcomes in both formal and non-formal and informal learning, the European Commission has developed a free internet portal for multilingual classifier ESCO (European Skills, Competencies, Qualifications and Occupations) [20] that joins the labor markets of the EU member states and allows jobseekers and employers to communicate more effectively with definitions of skills, training and work in all European languages. The main elements of ESCO are professions, skills and qualifications related to the labor, education and training market in the EU. ESCO allows users to determine what knowledge and skills are usually required to work in a particular profession. Each ESCO concept is associated with at least one term in all ESCO languages. Thus, ESCO is a source of information on competencies relevant to the labor market in the international dimension, both for the development of higher education standards and for the review of educational programs in higher education, given that professional standards are currently lacking in many professions. ESCO is published as Linked Open Data, and developers can use RDF format. In this work we consider ESCO as source ontology for semantic application that needs in information about structure of skills and competencies.

VI. IoT and Big Data sources in semantic technologies

Now devices of Internet of Things (IoT) generate a significant amount of structured and unstructured data (Big Data), that can be used in various semantic applications for building of user profiles with the help of based on application of the Semantic Web methodologies and best practices to the IoT data [21].

IoT software implementation for e-learning can be based on wireless (sensor) networks and mobile (portable, portable) computing platforms (smartphones, tablet PCs, embedded computing modules). For example, IoT application for the use of smart cards based on NXP Semiconductors' MAD Standard - MIFARE Application Directory can be used in e-learning systems, Big Data storage, ensuring the security of this data, personal or group user authentication [22].

The use of smart cards has increased the level of validation of non-formal and informal training for home office employees to improve their competencies during the quarantine period

VII. AdvisOnt system

Agricultural advisory systems are widely used now for fast dissemination of agricultural knowledge and information, introduction of modern scientific research and technologies in production, mobility and constant advanced training of agricultural specialists. Their implementation becomes an important factor in competitiveness of rural economy.

Development of the agricultural sector causes the dissemination of modern knowledge among agricultural manufacturers, relevant and efficient training and information support of their employees. AdvisOnt is an agro-advisory system that ensures consulting services for the agricultural sector of economy. It implements an ontological representation of advisory knowledge. AdvisOnt provides formalization and harmonization of semantic models of advisory objects with use of semantic identification and documentation of non-formal and informal learning outcomes and competence-based representation of advisory IOs [23].

We consider this system because in analysis of semantic application we need in information about realization of knowledge base and methods of its processing. Therefore we can propose objective information about compliance to the Semantic Web requirements only for those software where we participate in its design and development.

A. General architecture of AdvisOnt

The general architecture of AdvisOnt defines relations between main subjects of advisory activities (Fig. 2):

- applicant — person needed in some work in agricultural domain and has a set of relevant competencies and skills;
- employer — person or organization needed in employees for execution of some task or work on some position;
- providers of learning services — organizations that propose various (formal, non-formal and informal) learning means for expansion of personal competencies;
- advisors — experts specialized in agricultural domain of fixer region that can use domain knowledge for refinement of mutual interests of employers and applicants and provides advising services if applicant qualification needs in additional learning according to employer demands.

AdvisOnt helps in interaction between expert-advisor and other subjects by e-Extension interface and uses external semantic IRs and knowledge bases: ESCO as a source of structured representation of domains competencies and qualifications; user profile ontology to determine the structure of the applicant's model; domain ontologies containing facts and rules of specific agricultural tasks; expert knowledge and soft skills used for semantic formalization and matching of vacancies and resumes; ontology of open online learning services (such as Massive Open Online Courses (MOOCs) [24].

All classes of ESCO ontology used by AdvisOnt are stored into Turtle file. SPARQL queries and connectors are used for selection of skills and occupations from this RDF repository. The answers of SPARQL queries can be represented as result sets or RDF graphs. In the same way, the results of requests are returned to the RDF repository. Analysis of this ontology is used to define semantic similarity estimates for competence concepts [25].

Domain ontologies are integrated into the RDF repository with use of database of semantic graphs GraphDB. This database complies with W3C standards and links data from various sources, indexes them for semantic search and uses elements of NL analysis. GraphDB connectors provide fast search for keywords and aggregations usually realized by external services with use of synchronization on level of entities defined by URI, properties and property values.

AdvisOnt can use IoT information about user informal competencies and skills acquired from his/her mobile devices to estimate time and quality of learning. Processing of Big Data is not realized now but in future AdvisOnt can import domain rules resulted from Big Data and their metadata by other semantic applications and represented on base of ontologies.

We can consider AdvisOnt as a semantic application because it provides [26]:

- personified interaction for potential employers and job seekers based on use of personal intelligent agents;
- registration of vacancies and resumes with their semantic analysis for formalized representation of used terms (on base of NL texts markup by ESCO ontology concepts);
- comparison of resumes and vacancies at the semantic level with use of semantic similarity of domain terms semantic relations between professions, knowledge, skills, competencies and qualifications defined by ESCO ontology;
- personified search of educational services and training courses based on validation of learning outcomes of formal, informal and non-formal level;
- comparison of training courses and programs with professions on base of their atomic competencies [22].

B. Semantic components of AdvisOnt

From the point of view of relations between components of semantic technology we can AdvisOnt contains:

- external KBs represented by ESCO ontology, MOOC ontology for learning courses, various agriculture domain ontologies and user profile ontologies from other IISs and internal ontology of competencies and qualifications;
- IO models that formalize structure and features of typical advisory IOs: competencies, skills and professions, applicants, vacancies and resumes, etc.;
- IRs that contain semantic markup based on structure of typical IOs and provide additional information about individuals of classes: semantic Wiki resources that can contain NL text and multimedia (Fig. 3).

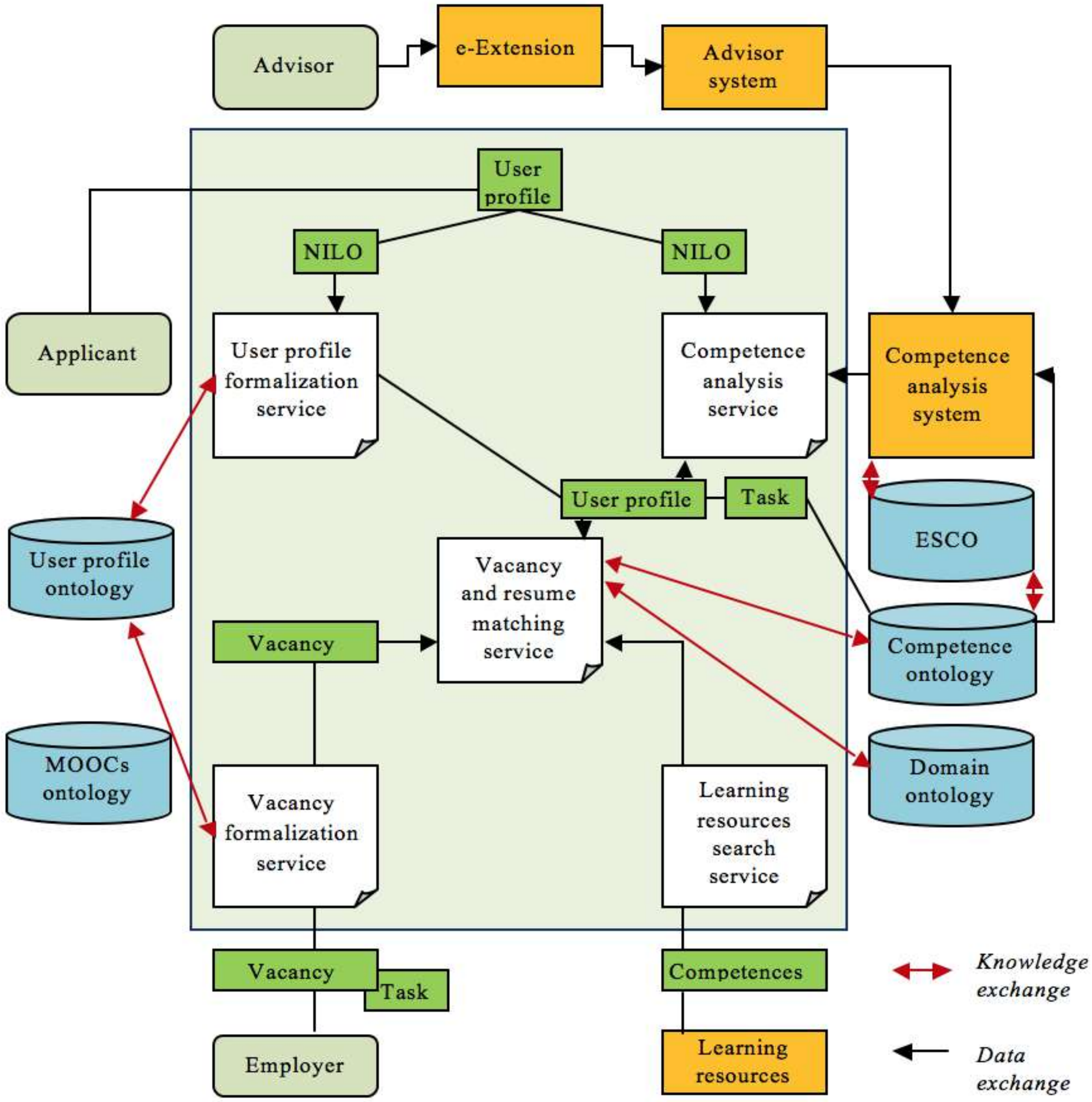


Figure 2. General architecture of AdvisOnt advisory system.

C. AdvisOnt as a Semantic Web application

We can rate AdvisOnt as semantic application of the because this IS conforms to the requirements of the Semantic Web Challenge: Minimum requirements of the Semantic Web application for AdvisOnt can be interpreted like this:

1) data meaning plays a key role in its functioning: AdvisOnt process meaning of vacancies and resumes with use of ontologies to link various terms with concepts and realize original non-trivial approach based on atomic competencies for matching of IOs that cannot be obtained without analysis of their semantics.
2) AdvisOnt uses ontologies and IRs from different owners (ESCO, MOOCs, etc.) that can be hanged by other ones (for example, by ontology of national qualification system or other e-learning platform) without changes of software, these sources are heterogeneous syntactically (ontologies, Wiki IRs, thesauri, etc.) and semantically (use different NL languages and describe various domains), and contain real-world data used by other commercial applications.
3) Search for information is carried out in the real information space of the Web: results of AdvisOnt depend on user requests and actual information retrieved from the Web about vacancies, resumes and learning courses.

AdvisOnt works into the open information space, i.e.

Figure 3. Example of Semantic Wiki resource pages (wiki.isofts.kiev.ua).

recommendations are not absolutely optimal but are based on available data and knowledge of system. This IS is based on processing of IRs that are represented on languages developed by the Semantic Web -– RDF and OWL.

VIII. Conclusion and future work

We consider semantic approach to development of advisory systems that provides methods and means of integration and unified representation of heterogeneous distributed knowledge and data. Analysis of the Web semantic resources as an universal knowledge source of about meaning of information objects helps in selection of external information sources of advisory system.

Analysis of the development of modern ITs based on knowledge shows the feasibility of using the Semantic Web initiative in the development of applied IS for validation of the on formal and non-formal learning outcomes that can to combine the market of educational services with the labor market. Conformity of developed IS with requirements to the Semantic Web application provides more flexible, intelligent and personified IS that is oriented on processing into the open information space and use of interoperable ontological knowledge.

In future we plan to develop more detailed models of advisory IOs and use II methods for acqusition of knowledge about these IOs from various Web IRs (such as Big Data and IoT).

References

[1] Semantic Web. URL: https://www.w3.org/standards/semanticweb/.

[2] M. Sabou, M. d'Aquin, M. Motta. "Exploring the semantic web as background knowledge for ontology matching", In Journal on Data Semantics. 2008.

[3] L. Obrst, W. Ceusters, I. Mani, S. Ray, B. Smith. "The evaluation of ontologies Semantic Web". In Springer US. 2007, pp. 139–158. URL: http://philpapers.org/archive/OBRTEO-6.pdf.

[4] R. Studer, S. Grimm, A. Abecker. "Semantic web services. Springer", 2007.

[5] Hendler J. "Agents and the Semantic Web", In IEEE Intelligent systems. Vol. 16(2). 2001. pp. 30-37.

[6] "Resource Description Framework (RDF) Model and Syntax Specification". W3C Proposed Recommendation. January 1999. "URL: http://www.w3.org/TR/PR-rdf-syntax".

[7] "OWL Web Ontology Language. Overview", W3C Recommendation: W3C. 2009. URL: http://www.w3.org/TR/owl-features/.

[8] "SPARQL Query Language for RDF. W3C Recommendation", 2008. URL: https://www.w3.org/TR/rdf-sparql-query/.

[9] V.Golenkov, N.Guliakina, V.Golovko, V. Krasnoproshin "Artificial Intelligence Standardization is a Key Challenge for the Technologies of the Future". In Open Semantic Technologies for Intelligent System. OSTIS 2020. Communications in Computer and Information Science, vol. 1282. Springer, Cham., pp. 1-21.

[10] T. Hoppe, B. G. Humm, A. Reibold. "Semantic Applications - Methodology, Technology, Corporate Use", Springer Verlag, Berlin. 2018. pp. 1-12.

[11] "A Global Standard for Lifelong Learning and Worker Engagement to Support Advanced Manufacturing", 2019, http://www3.weforum.org/docs.
[12] D.Colardyn, J.Bjornavold "Validation of formal, non-formal and informal learning: Policy and practices in EU member states". In European journal of education, 2004, 39(1), pp. 69-89.
[13] S. Pryima,J. Rogushina, O. Strokan "Use of semantic technologies in the process of recognizing the outcomes of non-formal and informal learning", In Proc. of the 11th International Conference of Programming UkrPROG 2018, pp.226-235.
[14] S. Pryima, O. Strokan, J. Rogushina, A. Gladun, A. Mozgovenko "Ontology-based Methods and Tools for Validation of Non-formal Learning Outcomes", In Proc. of the 12th International Scientific and Practical Conference of Programming (UkrPROG 2020), CEUR Vol-2866, 2020, P. 50-60.
[15] V.I.Gorodetsky, O.N. Tushkanova "Semantic technologies for semantic applications. Part 1. Basic components of semantic technologies", In Artificial Intelligence and Decision Making, 2018, N 4, pp.61-71.
[16] T.R. Gruber "The role of common ontology in achieving shareable, reusable knowledge bases", https://pdfs.semanticscholar.org/0256/614d1dc55962708db9b791f70a29370f1d76.pdf.
[17] S. Staab, R.Studer "Handbook on ontologies", Springer Science and Business Media, 2013.
[18] S. Miranda, F. Orciuoli, V. Loia, D. Sampson "An Ontology-based Model for Competence Management", In Data and Knowledge Engineering,2016, 107, pp.51–66.
[19] K.O. Lundqvist, K.D. Baker, S.A. Williams "An ontological approach to competency management", 2017, http://www.eifel.org/publications/proceedings/ilf07/ Contribution110.doc.pdf (1.10.2017).
[20] ESCO (European Skills, Competences, Qualifications and Occupations). URL: https://ec.europa.eu/esco/portal/home.
[21] A. Gyrard, M. Serrano, G.A. Atemezing "Semantic web methodologies, best practices and ontology engineering applied to Internet of Things", In 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015, pp. 412-417.
[22] M.S. Levin, A. Andrushevich, A. Klapproth "Composition of management system for smart homes", In Information Processes, 2010, №10 (1), pp. 78-86.
[23] J. Rogushina, S.Priyma "Use of competence ontological model for matching of qualifications", In Chemistry: Bulgarian Journal of Science Education, V. 26, N 2, 2017, pp.216-228.
[24] Massive Open Online Courses (MOOCs). www.mooc.org. (last accessed 12.03.2021).
[25] J.Rogushina "Use of Semantic Similarity Estimates for Unstructured Data Analysis", In CEUR Vol-2577, Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security" (ITS 2019), Kyiv, 2019, pp.246-258. http://ceur-ws.org/Vol-2577/paper20.pdf.
[26] S. M. Pryima, O. V. Strokan, J. V. Rogushina, A. Y. Gladun, D. V. Lubko, V. M. Malkina "Ontological Analysis of Outcomes of Non-formal and Informal Learning for Agro-Advisory System AdvisOnt", In International Conference on Technologies and Innovation, 2020, pp. 3-17, Springer, Cham.

Разработка консультирующих систем на основе семантических технологий

Рогушина Ю.В., Гладун А., Прийма С., Строкань О.

Анализируются основные характеристики современных семантических технологий, рассматриваются их компоненты, которые используются для создания Веб-ориентированныъ консультирующих систем. Онтологический анализ используется как основа для формирования фоновых знаний о пользователях, их профессиях и компетенциях.

Стандарты Semantic Web обеспечивают технологический базис создания интеллектуальных приложений, поддеерживающих процесс консультирования. Валидация результатов неформального обучения требует сопоставления профилей пользователей с классификаторами профессий и компетенций. на основе проведенного исследования предлагается модели и методы представления, извлечеия и облаботки знаний, которые обеспечивают еффективность и динамичность разрабатываемой системы. методы сопоставления информационных объектов консультирования базируются на сравнении наборов атомарных компетенций.

Received 24.05.2021

New language for conceptual design of complex systems in the era of post-covid and mass digitalization

Alexandre Kourbatski
Department of Programming Technologies
Belarusian State University
Minsk, Belarus
kurb@unibel.by

Konstantin Mulyarchik
Department of Telecommunications and Information Technologies
Belarusian State University
Minsk, Belarus
k.mulyarchik@gmail.com

***Abstract*—This article focuses on the problems of using traditional languages in the work of the expert community, especially if the field of problem setting and design of complex systems, and on the steps taken nowadays and in the history by the world civilization to resolve them. The authors of the article believe that there has grown up a need in a new language, which, on one hand, would help to overcome the problems of translation, coordination of the conceptual apparatus, raising the effectiveness of communication between experts, and on the other hand, to make a breakthrough in the evolution of software development.**

***Keywords*—conceptual design, desing of complex systems, modelling notations, intelligent symantic systems**

I. Introduction

The rapid development of computer technologies that combine high computing power and low costs has led to the penetration of these technologies into all spheres of human activity and thus led to widespread digitalization. This, in turn, creates a demand for the development of technologies, approaches and methods that allow faster design and development of software systems, with lower costs and with a higher degree of reliability. A number of technologies have been developed for the design of classic software systems in which both the data model and business logic are hard-coded into the system itself. The technologis for design and development of intelligent computer systems, including their semantic compatibility, are becoming more widespread.

With the increasing complexity and scale of the systems being created, more and more people should be involved in these projects and ensure their coordinated activities. This is undoubtedly facilitated to a large extent by the developed technologies for design of solutions, which, however, focus primarily on the system being developed. From the point of view of the actors involved in the process, the process itself as a whole remained unchanged - this is communication in one of the classical languages, discussion of certain issues, coordination, achievement of a common understanding of the problem being solved. And in this sense, this process is filled with difficulties in mutual understanding, establishing contact, communicating a particular point of view so that others can understand it, not to mention the need to translate from one language into another, if there is such.

Many existing technologies for the design of software systems can also be used for standardization and unification in solving a number of other tasks: design of a domain, design of business and organizational systems, design of a knowledge system, etc. Despite this, they are little used by the international expert community for such purposes, probably, because of their focus primarily on computer systems, and secondly, because of their contextual dependence and the need to harmonize basic terms and concepts in natural languages anyway. Amongst such design technologies one can mention business analysis, systems engineering, the theory of inventive problem solving, and others, directly aimed at the consistent analysis of problems and the search for solutions.

It is necessary to pay attention to a number of breakthrough developments in the field of semantically compatible intelligent systems that provide fast and high-quality design of intelligent systems of various scales and complexity. Within the framework of this direction, the world expert community is solving a number of problems: standardization of intelligent systems, unification of the principles of building knowledge bases, problem solvers and interfaces of intelligent systems, semantic compatibility of various types of knowledge, various models for solving problems, intelligent systems among themselves and intelligent systems with their users [1]. On the one hand, these technologies are characterized by a strict formalization of their apparatus and rules of use, which makes it possible to quickly computerize and automate the corresponding models. On the other hand, they are quite close to how people communicate with each other, if an attempt is made to formalize such a communication.

We believe that there is a real need for unification, on the one hand, the design technologies and conceptual design of computer systems, including intelligent systems, and on the other hand, the ways of communication between experts, coordination of tasks to be solved, which may result in the creation of a new language of international communication.

II. Classic technologies for design of computer systems

Over the past 30 years, the world expert community has developed a number of notations, modeling tools and software systems design. The main groups of modeling standards are: descriptive models (IDEF0, OPM, SysML and others), analytical models and simulations (DIS, HLA, UML and others); data exchange standards, transformation models, general modeling standards and other modeling standards for specific areas: software development models, equipment design models, business process models [2]. The languages of system modeling are distinguished separately. The rapid development of cyber-physical systems, in particular such paradigms as the Internet of Things and Industry 4.0, in which technical systems and humans are combined, reveals the limitations in the ability of classical technologies and modeling notations to meet the new emerging requirements of these new paradigms. The reason for this is the universality of these models and their focus on universal applicability, in which a number of details specific to a specific subject area are ignored. This also implies the duration of their update cycle, which does not meet modern technology development requirements.

Thus, one of the topical areas of research in the field of designing computer and information systems is the study of new languages of conceptual modeling and the development of tools for their active and widespread use [3]. It is discussed that the basis of domain-based conceptual modeling is metamodeling, which allows combining BPMN [4], ER [5], EPC [6], UML [7] and Petri Nets [8], [9] within one tool.

III. Traditional language in conceptual design of computer systems

Moving to an earlier stage of the software development life cycle - to the level of conceptual design, problem statement, it should be noted that the traditional language (English, Russian, etc.) is still the means of design and coordination. Concepts are drawn up in the form of text documents in traditional languages, using certain schemes, or elements of notation. Traditional language, carrying both communicative and cognitive functions, has one drawback: all languages that exist today are linear, they are directly connected with time so that the information is revealed sequentially. All this leads to several problems in their use.

A. Problem of harmonizing of conceptual apparatus

First of all, the effective work of the expert group is conditioned by the presence of an agreed conceptual apparatus and context among all its members. This includes not only the terms and meanings that stand behind them, but also the interrelationships of the terms with each other.

Especially when innovative things are discussed in an expert environment, it is necessary to form a conceptual apparatus ab initio, since it simply does not exist. This takes a long time due to the ineffectiveness of natural language, no matter what language to consider.

It should be emphasized that even experts who speak the same language do not always understand each other, especially if they are discussing new problems or innovative approaches. For example, in a discussion there may be situations when they use the same term, but each understands it differently. The worst case scenario is when the concepts of these terms do not intersect at all.

The face of the matter is that at the initial stage of its development, the traditional language was a simple means of communication in everyday life. But over time, languages have become a working tool of the expert layer. The level of complexity of the tasks being solved has increased significantly, but the main tool for solving them – the means of communication – has remained the same, which leads to the slow work of experts, the slow work of the government, and business, especially large ones, contrasted with the extreme dynamics of the modern world [10].

A good example – a group of highly qualified world experts for five years formulated fifteen basic terms in the field of information and cybersecurity. A lot has already changed during this time. In this case, it is obvious that the bottleneck is the language of communication.

B. Problem of translation between traditional languages

Another problem of using traditional languages at an expert level, closely related to the first one, is the need to translate from one language to another. In this case, a partial loss and replacement of meaning inevitably occurs, which also leads to a loss of the effectiveness of the work of experts. Unfortunately, there are only a few qualified translators in the world who not only know both languages but also know the specifics of the subject area, thus able to convey not only the text, but also the specific meaning. At the same time, at the everyday level, the problem of translation from language into language can very soon be solved with the help of automatic translation systems. They have achieved quite acceptable results, are progressing rapidly, and the role of human translators will diminish.

Until recently, during the period of globalization, the world community used English as the language of international communication. However, the coronavirus pandemic that came in 2020 led to some reduction in global international communication and created the preconditions for the emergence of independent regional economic zones with their own civilizational model, relying on norms, ideas and traditions familiar to the most. Within each zone, the dominant national language common to the zone and the national languages of other peoples who have joined the zone will be used [11].

Since the economic and cultural interaction between the zones will be preserved, the need for international communication in the new world will remain. Cultural exchanges, tourist trips, joint scientific research, especially of a fundamental nature, and joint projects will remain. It will also be necessary to solve problems common to human civilization, for example, in the field of ecology.

English, which is now used as a means of international communication, is not suitable in this sense. It seems unlikely that all countries will officially and forever recognize the language of one of the zones as a means of international communication. Therefore, in the world of regional zones, English, like any other national language, is likely to be used as the main international language of communication for only a limited, albeit, possibly quite long, time.

C. Attempts to create an international language

Earlier in world history, several attempts were made to create an artificial (planned) language of international communication. As conceived by their authors, such a language should be more logical, simpler and, accordingly, easier to learn than any ”foreign” language. It should allow one to get rid of the burden of ineffective historical layers inherent in any modern language. They hoped that it would be accepted by representatives of different countries as the second language of educated people. In the future, this will make their language international, and then universal.

The most famous and widespread planned language is Esperanto, created by the Warsaw linguist and optometrist Ludwik Zamenhof in 1887. According to various estimates, today it is spoken by from one hundred thousand to two million people - in the best case, only 0.03% of the world's inhabitants. Nevertheless, only Esperanto can be considered an established planned language. The rest are linguistic projects, mostly within the ownership of small groups of enthusiasts. For more than two thousand years of described human history, linguists have counted about 1000 of such projects.

It is also necessary to pay attention to Latin. The Latin alphabet is the basis of writing in many modern languages. Throughout history, Latin has been used in conversations between diplomats because the parties did not speak their partner's language. Latin is known not only as the language spoken in the Roman Empire, but also the language that has long served as a source for the formation of international socio-political and scientific terminology.

IV. Technologies for desing of intelligent systems

With the development of computer and computing technology, the increase in complexity of the problems being solved, the requirements for design and development technologies in terms of their flexibility and efficiency of modeling began to increase as well. A whole class of information systems has appeared, called semantic systems, in which, unlike classical systems, the data structure is determined by the data itself, and the algorithms for its processing are described by a number of statements, and not rigidly fixed in the program code of the information system. Since such systems are universal in relation to subject areas, their integral part is the presence of a structured description of the subject area - ontology.

Today there is a number of breakthrough developments [1], [12]–[14] in the development of flexible semantically compatible intelligent systems, which have great potential for solving the problem of coordinated communication between experts and the problem of conceptual design. The cornerstone of these studies is the idea of developing a universal language for representing the meaning of knowledge. It is noted that such a language should have the property of nonlinearity [1].

At the applied level, such a universal language describes not only the knowledge itself, but also its meaning through the definition of both information and meta-information. Due to such property it can cover a number of types of information content: specifications and descriptions of various entities, documentation and requirements for systems and their evolution, descriptions of domain areas, definition of tasks and classes of tasks, description of problems, description of solutions to these problems, description of ways to solve various typical problems, description of projects and concepts. If necessary, this language has the ability to expand the description of new types of knowledge, thus allowing it to be widely distributed for solving problems in various applied areas of computer systems. Semantic code SC-code (Semantic Code), presented in [15], refers to such a language.

V. Prerequisites for the emergence of a new language of communication

A. Environment of communication

Linguists note that the language is directly connected with the environment of communication and its distribution. For natural languages such environment is human community; for programming languages - this is a particular software development environment. Unsuccessful attempts to create an artificial language before are associated primarily with the lack of a proper environment for its dissemination, since we had only natural ways to communicate.

Today the Internet is such an environment for the whole world. Now any person has at least something: a smartphone, tablet or computer. The new communication medium is able to very quickly conquer the audience.If a new language could be attached to a computer program, then the entire population of the globe would be immediately covered. As an example, we can cite software from Microsoft, which is imposed all over the world.

We believe that the stage of conceptual design and problem setting in the software development can be used as a reference point. In addition to solving the abovementioned problems of aligning the conceptual apparatus and translation, this language can automatically become a means of the conceptual software design.

When programming was born, the task was set to use separate programming languages as languages understandable by the machine. The implication was that in the future, the computer will learn to understand natural language. And when they say "high-level programming language", they mean that it comes close to a natural language. We haven't made much progress, but the goal remains. It seemed that if a computer began to understand a language close to natural, both the programming process itself and the formulation of the problem would be simplified. But now we have come to understanding that natural language is not very effective. There are many ambiguities that are difficult for a computer to understand. Therefore, we need a new language, which at the first stage would be the language of communication of experts, but at the same time much more comprehensible to a computer. Then, perhaps, it would be possible to make a leap when the computer understands the language of communication, and the level of programming increases. As a process, it will become more natural, because a person does not need to learn a separate programming language, it is enough for him to conduct a dialogue in a new language. Indeed, this language may be closer to the IT environment where its lack is already obvious. Considering that the production of software in the world has become a massive process, the need for such language is increasing. It can help solve many problems in the era of mass digitalization.

VI. Conclusion and further research

On the agenda is the task of creating, in the future, a new language of international communication, which, on the one hand, would be the language of communication for the expert layer, and on the other hand, would allow to effectively solve the problems of conceptual design of systems.

A number of open questions arise that need to be resolved when developing such a language. What well-known technologies, best practices and know-hows can the new language rely on? To what extent should existing artificial intelligence and machine learning technologies be taken into account and used in the development of this language? For example, the previously mentioned open semantic technologies for the design of intelligent systems. Or UML notations, IDEF0, No-code and Low-code technologies mentioned earlier in the article. To what extent the approaches that have shown their viability in the development of programming languages should there be used? Programming languages, keeping their basic corpus unchanged, are developing due to constantly updated libraries.

What alphabet to take as a basis? Use one of the existing languages as a prototype or start from scratch? Perhaps the eastern hieroglyphic writing could be taken into account. For example, Chinese characters may be closer to a new language of communication - they are already inherent in great visualization. Hieroglyphs are more like visual objects that are used in modern programming than words in English or Russian. Should the new language effectively use the elements of visualization?

In addition to applied issues, the conceptual questions arise too. What should be the methodological foundations for building a new language? For example, this could be systematization, systems thinking, the apparatus of models, the theory of business analysis, the theory of inventive problem solving, etc. What should be the contribution of other knowledge areas to the new language?

References

[1] V. Golenkov, V. Golovko, N. Guliakina, V. Krasnoproshin, The standardization of intelligent computer systems as a key challenge

of the current stage of development of artificial intelligence technologies. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2020, pp. 73-88
[2] S.Enin, Y.Gerasimuk, Effektivnoe upravlenie tsifrovizatsiei: arkhitekturnyi podkhod [Effective digital governance: an architectural approach]. *Razvitie informatizatsii i gosudarstvennoi sistemy nauchno-tekhnicheskoi informatsii RINTI-2020 [Development of informatization and the state system of scientific and technical information RINTY-2020]*, 2020, pp.33-40
[3] D.Bork, An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem. *Commun. Assoc. Inf. Syst.,* 2019, pp.673-679.
[4] OMG: The BPMN specification page. Available at: http://www.bpmn.org (accessed 2021, May)
[5] P.Chen, The entity-relationship model—toward a unified view of data. *ACM Trans. Database Syst.*, 1976, pp. 9–36
[6] Software AG: ARIS—the community page. Available at: http://www.ariscommunity.com (accessed 2021, May)
[7] OMG: The UML resource page. Available at: http://www.uml.org (accessed 2021, May)
[8] W.Reisig, Understanding Petri Nets. *Springer, Heidelberg*, 2013.
[9] C.A.Petri, W.Reisig, Petri net. *Scholarpedia*, 2008, p.6477.
[10] A. Kourbatski, Novaya semantika yazyka IT – eto vam ne angliiskii... [The new semantics of the IT language – this is not English...]. *Vesnik suvyazi [Herald of Communication]*, 2020, vol. 6, pp. 30-33
[11] A. Kourbatski, M. Zekov, Yazyk obshcheniya v mire, kotorogo nikto ne zhdal [The language of communication in the world that no one expected]. Mezhdunarodnaya zhizn [International Affairs]. Available at: https://interaffairs.ru/news/show/27236 (accessed 2021, May)
[12] V.Golenkov, D.Shunkevich, I.Davydenko, N.Grakova, Principles of organization and automation of the semantic computer systems development. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2019, pp. 53-90
[13] S.Listopad, Agent Interaction Protocol of Hybrid Intelligent Multi-Agent System of Heterogeneous Thinking. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2020, pp. 51-56
[14] A.Kolesnikov, E.Jasinski, S.Rumovskaya, Predicative representations of relations and links of the intellectual operational-technological control in complex dynamic systems. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2020, pp. 43-50
[15] V.Golenkov, N.Guliakina, Principles of building mass semantic technologycomponent design of intelligent systems. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2011, pp. 21-58

Новый язык международного общения в эпоху постковида и массовой цифровизации

Курбацкий А.Н., Мулярчик К.С.

В данной статье речь идет о проблематике использования традиционных языков в работе экспертного сообщества, в особенности в области постановки задачи и концептуального проектирования сложных систем, и о тех решениях, которые предпринимаются и предпринимались ранее мировой цивилизацией для ее разрешения. Авторы статьи полагают, что в мире назрела необходимость в новом языке, который, с одной стороны, помог бы преодолеть проблемы перевода и согласования понятийного аппарата между экспертами, а с другой – совершить рывок в развитии программирования.

Received 31.05.2021

Specialized KD-Agent for Knowledge Ecosystems

Viktor Krasnoproshin
Faculty of Applied Mathematics and Computer Science
Belarussian State University
Minsk, Belarus
krasnoproshin@bsu.by

Vadim Rodchenko
Faculty of Mathematics and Informatics
Yanka Kupala State University of Grodno
Grodno, Belarus
rovar@grsu.by

Anna Karkanitsa
Faculty of Mathematics and Informatics
Yanka Kupala State University of Grodno
Grodno, Belarus
a.karkanica@grsu.by

***Abstract*—One of the base elements of any knowledge ecosystem is a software agent. The agent receives data about the internal events of the ecosystem, interprets data and executes commands that affect the environment. The paper proposes an option for the implementation of the specialized Knowledge Discovery agent (KD-agent). The input for the agent is the a priori dictionary of features and the training set. As the outcome of the agent activity previously unknown patterns are revealed and can be interpreted within the subject domain. The effectiveness of the proposed approach is demonstrated on the example of model data analysis.**

***Keywords*—knowledge ecosystem, data mining, supervised learning, training set**

I. Introduction

The knowledge ecosystem is a complex adaptive system including a database, a knowledge base and experts [1]. The development and implementation of such systems is one of the priority courses of information technologies growth and usage [2], [3].

The knowledge ecosystem is intended to provide high-quality interaction between objects for the effective implementation of the decision-making process. Typically, it includes technological core, critical interdependencies, knowledge agents and performative actions [1].

Knowledge agents receive and interpret data about internal ecosystem events and execute commands that have impact on the environment. The most important agents' properties are autonomy, social ability, reactivity and pro-activity [4].

The paper describes the process of constructing a specialized intelligent Knowledge Discovery agent (KD-agent). The agent's input data are the a priori dictionary of features and the training set. In automatic mode, the agent performs data analysis on which a set of informative ensembles of features are formed ensuring the separation of classes. The results of the practical usage of KD-agent on the example of model data analysis are described.

II. Knowledge Discovery in Databases

The development of novel and application of existing data mining methods and technologies is a promising avenue of knowledge ecosystems development and use.

The ongoing progress in the development of artificial intelligence technologies is largely due to the wide implementation of machine learning methods based on identifying empirical patterns in datasets [5].

During the learning process an intellectual system is provided with a set of positive and negative examples related by a previously unknown pattern. As a result of learning, a decision rule (algorithm) used to split the presented examples into positive and negative is generated [6].

Thus, machine learning methods traditionally construct practically useful algorithms (decision rules) that implicitly express empirical patterns. For example, the result of the Supervised Learning is a classification algorithm that is a certain practically useful "black box". This result, unfortunately, defies any interpretation within the subject domain.

Knowledge Discovery in Databases (KDD) is a process of discovering in the initial datasets a previously unknown, useful and interpretable patterns, which are further necessary for effective decision-making [7]. However, as it's shown above, machine learning methods do not fully satisfy all the KDD requirements. They do not allow to interpret the discovered patterns.

Formally, the KDD process includes five major stages and can be represented as follows (Fig. 1):

$$DW \xrightarrow{S1} TD \xrightarrow{S2} TS \xrightarrow{S3} DM \xrightarrow{S4} Ps \xrightarrow{S5} K$$

where DW and TD — data warehouse and a target dataset respectively; TS — training set; DM — Data Mining procedure; Ps — resulting set of patterns; K -– knowledge; S1 (Stage 1) — the stage of formulating the goal and objectives of the KDD process and the formation of a target dataset on which the search for patterns will be carried out; S2 (Stage 2) — the stage of data preprocessing and formation of a training set; S3

(Stage 3) — execution of the Data Mining procedure; S4 (Stage 4) -– building class patterns; S5 (Stage 5) -– patterns interpretation in terms of the subject domain.

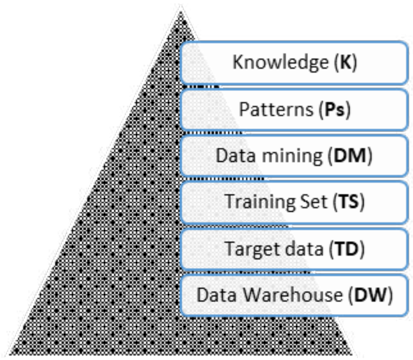


Figure 1. Patterns detection circuit.

Thus, the KDD process begins with the selection of an alphabet of classes, a set of observed features and the construction of an a priori dictionary of features (PDF). Each observed object is then represented as a vector of features from the PDF and, as an outcome, the training set is formed. Further, using the training set, estimates of informative value are calculated for all possible combinations (ensembles) of features from the PDF (in terms of the correct division of the pre-set classes).

After that, domains of classes (class patterns) for each combination of features are constructed. And on the results of the analysis of the mutual placement of patterns, the informativeness of the corresponding ensembles of features are estimated.

Therefore, as a result of the KDD process implementation, we are acquiring knowledge in the form of the informative significance of ensembles of features from the PDF. The knowledge thus acquired can be interpreted in terms of the subject domain, since each feature in any combination carries a specific semantic load.

III. Knowledge Discovery vs Supervised Learning

As noted previously, a classification algorithm (decision rule) is constructed on the basis of the results of the Supervised Learning procedure performance.

Traditionally, the learning process is reduced to the construction of decision rules that deliver the extremum of some functional. Therefore, decision rules families, generally, are selected a priori with accuracy up to parameters. In the learning process, specific values of the parameters which provide the extremum of the pre-set functional are determined.

It is considered that the dictionary of features is used not only for constructing a training set. It also defines a feature space in which the decision surfaces between classes are built.

On the basis of Machine Learning methods, it is possible to solve many applied problems that quite recently were considered non-trivial. In particular, impressive results have been obtained using the technology of artificial neural networks.

At present, neural network technologies allow to provide not only a high-level quality of learning, but also include for a nearly autonomous execution of the Supervised Learning procedure. However, application of artificial neural networks, as well as other Machine Learning methods, is limited to developing a classification algorithm. It turns out that a useful result of the entire resource-intensive process of training set preparing (about 80% of all costs) and processing is precisely the classifier. The classifier in fact is a “black box” that is not possible to further interpretation.

Thus, the main objective of machine learning methods is to build classification algorithms. In fact, this is a weak side of the approach. Although as an outcome it is succeed to learn how to separate class patterns, but at the same time there is no information of any kind about the properties of classes themselves.

An alternative to the data analysis of the training set can be an approach based on the idea of extracting some subsets from the a priori dictionary of features that would provide the separation of classes in a given feature subspace. Actually, features and their various combinations have varying informativity extent characterizing the properties of classes. Suppose the a priori dictionary contains n features. Obviously, 2n-1 of all possible combinations (ensembles) of features can be constructed [8]. If in such a set of combinations there is an ensemble by which the classes are well separating in a given feature subspace, then it can be stated that:

1) previously unknown patterns of classes properties are discovered;
2) these patterns can be interpreted in the subject domain terms;
3) based on the revealed properties, the problem of constructing a classifier becomes trivial.

To detect the described ensembles, it is proposed, initially, to build the domains of classes based on the data of the training set. Thereafter, in the corresponding feature subspace, we can calculate the estimates of their mutual placement.

Essentially, the process described above implements a typical procedure of knowledge discovery in dataset. On its basis, it is proposed to build an intelligent KD-agent that gets as an input a priori dictionary of features and a training set. Such KD-agent will automatically process the data and form the discovered patterns.

IV. Functioning of KD-agent

Within the classical for Machine Learning approach, the following statement of classification problem is adopted:

Let the objects descriptions X and the acceptable answers Y for objects classification are given. Suppose there is an unknown target dependency $y*: X \to Y$*, which values* $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ *are known only for the training set objects.*

It is necessary to construct an algorithm $a: X \to Y$ *that would approximate this target dependency not only on the objects of the finite set, but also on the entire set X* [9].

The solution to this problem is typically carried out in two stages. First, a certain family of algorithms is specified up to parameters. Then, in the learning process, the values of the parameters are determined that provide the extremum of the preselected functional.

The selection of algorithms model (family) $A = \{a : X \to Y\}$ is a non-trivial problem. Such a choice requires the participation of a qualified specialist. It means that learning is only carried out in an automated, but not automatic mode. Another serious disadvantage is that the resulting algorithm $a : X \to Y$ is a "black box" whose outcomes cannot be interpreted.

The application of the learning approach described above (alternative) avoids the mentioned disadvantages. The following modification of the problem statement is proposed:

Let the objects descriptions X and the acceptable answers of objects classification Y are given. There is an unknown target dependency $y*: X \to Y$*, which values* $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ *are only known for the training set objects.*

It is required to find feature subspaces where class patterns do not intersect.

Let the training set $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be formed on the basis of the dictionary of features $F = \{f_1, \ldots, f_n\}$. Let $V = \{v_1, \ldots, v_q\}$ denote the set of all possible combinations (ensembles) of features from F. Obviously, V contains $q = \sum_{i=1}^{n} C_n^i = 2^n - 1$ subsets.

The algorithm of constructing feature subspaces $V^* = \{v_1^*, \ldots, v_k^*\}$, where class patterns do not intersect is as follows:

Step 1. In the set V, n combinations $V^+ = \{v_1^+, \ldots, v_n^+\}$, that contain one feature are being selected. For each individual feature, class patterns are built and their mutual placement is estimated. If the patterns do not intersect, then the feature is included in the resulting set V^*. The combinations that contain this feature are excluded from the set V. If the patterns intersect, then the feature is excluded from V.

Step 2. Let $V^\Delta = \{v_1^\Delta, \ldots, v_p^\Delta\}$ denote by the subset obtained as a result of the set V transformation at the previous step. For each individual combination from V^Δ, class patterns are built and their mutual placement is estimated.

If the patterns do not intersect, then the combination of features is included in the resulting set V^*. And all elements that contain this combination are being excluded from V.

If the patterns intersect, then the combination is excluded from V^Δ. The process is repeated until V^Δ becomes empty.

As a result of the analysis of all elements from $V = \{v_1, \ldots, v_q\}$ (possible combinations of features from the dictionary $F = \{f_1, \ldots, f_n\}$) a set $V^* = \{v_1^*, \ldots, v_t^*\}$ will be constructed, where $0 \le t \le q$.

On the basis of each separate ensemble-combination $v_i^* \in V^*$, we formulate a previously unknown, empirically revealed pattern: **in the feature space of the subset** v_i^* **the classes do not intersect**. It should be noted that within a specific applied problem, each combination of features v_i^* can be interpreted by a subject domain expert.

So, as an input, the KD-agent receives an a priory dictionary of features $F = \{f_1, \ldots, f_n\}$ and the training set $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Based on the above algorithm, agent forms the set $V^* = \{v_1^*, \ldots, v_t^*\}$, where $0 \le t \le q$ (Fig. 2).

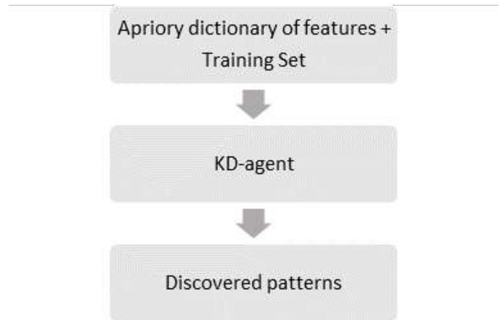


Figure 2. KD-agent workflow.

Let's note that the so constructed KD-agent satisfies all major characteristics subjecting to agents in multi-agent systems (autonomy, local representations, decentralization).

V. Application of the KD-agent

Let's demonstrate the efficiency of the KD-agent by the example of analyzing the training set data aiming to reveal hidden patterns.

Example. Let the given:

- number classes – **even** and **odd**;
- a priori dictionary of features F = {units, tens, hundreds, thousands, tens of thousands, hundreds of thousands, millions};
- training set of seven-bit integers, which contains 2000 even and 2000 odd numbers.

Table 1 shows the results of researching the intersection of class patterns based on the feature **units**, where

$$NE_i = \textit{Number of even}_i$$
$$NO_i = \textit{Number of odd}_i$$

$$a_i = \begin{cases} NM5_i + NNM5_i, NM5_i = 0 \lor NNM5_i = 0 \\ 0, NM5_i > 0 \land NNM5_i > 0 \end{cases}$$

$$Intersection = \frac{20000 - \sum_{i=0}^{9} a_i}{20000} * 100\%$$

Table I shows that even numbers are lack of 1, 3, 5, 7, 9 in the unit's digit, and odd numbers are lack of 0, 2, 4, 6, 8. In addition, the **units** feature provides an absolute separation of the classes **even** and **odd** since the $Intersection = 0\%$.

Table I
Experiment results for the feature **units**

Digit	Number of even	Number of odd
0	405	0
1	0	415
2	408	0
3	0	398
4	373	0
5	0	383
6	423	0
7	0	404
8	391	0
9	0	400

Table II
EXPERIMENT RESULTS FOR THE FEATURE **TENS**

Digit	Number of even	Number of odd
0	204	192
1	201	204
2	205	190
3	190	216
4	203	191
5	183	216
6	200	192
7	216	194
8	197	195
9	201	210

Table II shows the analysis results for the feature **tens**. It could be seen therefore that this feature has no the property of class separation since $Intersection = 100.0\%$.

Table III presents the analysis results for the feature **millions**. The table shows that this feature does not have the property of class separation neither, since $Intersection = 100.0\%$.

Table III
EXPERIMENT RESULTS FOR THE FEATURE **MILLIONS**

Digit	Number of even	Number of odd
0	201	211
1	212	211
2	193	187
3	174	190
4	189	191
5	210	181
6	208	204
7	196	210
8	218	212
9	199	203

Table IV shows the results of the analysis for all features from the a priori dictionary.

Table IV
EXPERIMENT RESULTS FOR ALL FEATURES

Feature name	Intercession (%)
units	0.0
tens	100.0
hundreds	100.0
thousands	100.0
tens of thousands	100.0
hundreds of thousands	100.0
millions	100.0

Let's note that the algorithm running time spent on solving this problem was only 0.09 seconds.

VI. CONCLUSION

The paper presents the implementation variant of the specialized knowledge discovery agent (KD-agent). The input for such an agent is the a priori dictionary of features and the training set. As the outcome of the KD-agent activity previously unknown patterns are revealed and can be interpreted by experts of the corresponding subject domain. It is easy to see that the outcomes of the KD-agent's work can be further used by other agents of the ecosystem.

The effectiveness of the proposed approach is demonstrated on the example of the model data analysis.

REFERENCES

[1] Knowledge ecosystem [Electronic resource]. Available at: https://en.wikipedia.org/wiki/Knowledge_ecosystem (accessed 2021, March).

[2] C.W. Choo, N. Bontis, The Strategic Management of Intellectual Capital and Organizational Knowledge, Oxford University Press, 2002, 748 p.

[3] S. Russel and P. Norving, Iskusstvennyj intellect: sovremennyj podkhod [Artificial intelligence: a modern approach], Moscow: Williams Publishing House, 2006, 1408 p.

[4] AIportal [Electronic resource]. Available at: http://www.aiportal.ru/articles/multiagnt-systems/weak-and-strong-intelligent-agent.html (accessed 2021, March).

[5] P. Flach, Mashinnoe obuchenie. Nauka i iskusstvo postroeniya algoritmov, kotorye izvlekayut znaniya iz dannykh [Machine learning. Science and art of constructing algorithms that extract knowledge from data], Moscow: DMK Press, 2015, 400 p.

[6] MachineLearning [Electronic resource]. Available at: https://en.wikipedia.org/wiki/Machine_learning (accessed 2021, March).

[7] Data mining [Electronic resource]. Available at: https://en.wikipedia.org/wiki/Data_mining (accessed 2021, Feb).

[8] V.G. Rodchenko, Pattern Recognition: Supervised Learning on the Bases of Cluster Structures, XIII International Conference "Pattern Recognition and Information Processing", Minsk, BSU, Communications in Computer and Information Science, Springer, 2017, pp.106-113.

[9] Supervised learning [Electronic resource]. Available at: https://en.wikipedia.org/wiki/Supervised_learning (accessed 2021, March).

Специализированный KD-агент для экосистем знаний

Краснопрошин В.В., Родченко В.Г., Карканица А.В.

Одним из базовых элементов любой экосистемы знаний является программный агент. Находясь в среде экосистемы, агент получает данные о внутренних событиях, интерпретирует их и выполняет команды, которые воздействуют затем на среду. В статье предлагается вариант реализации специализированного knowledge discovery агента (KD-агента). Входными данными для агента являются априорный словарь признаков и обучающая выборка. В результате работы агента выявляются ранее неизвестные закономерности, которые могут быть проинтерпретированы экспертами-специалистами соответствующей предметной области. Эффективность предложенного подхода демонстрируется на примере анализа модельных данных.

Received 23.05.2021

Ontological approach to the development of hybrid problem solvers for intelligent computer systems

Daniil Shunkevich
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: shunkevich@bsuir.by

***Abstract*—The paper considers an ontological approach to the development of problem solvers for intelligent computer systems based on the OSTIS Technology. The formal interpretation of such concepts as action, problem, class of actions, class of problems, method, skill is clarified, which together made it possible to define on their basis the concepts of a problem-solving model and a problem solver. The results obtained will improve efficiency of the component approach to the development of problem solvers and automation tools for the development of problem solvers.**

***Keywords*—OSTIS, problem solver, multiagent system, problem-solving model, ontological approach**

I. Introduction

The problem solver (along with the knowledge base) is a key component of an intelligent system, on which its ability to solve various problems significantly depends. The peculiarity of problem solvers of intelligent systems in comparison with other modern software systems is the need to solve problems in conditions when the information required is not explicitly localized in the knowledge base of the intelligent system and must be found based on any criteria [1].

If at the dawn of the development of artificial intelligence technologies scientists have been trying for a long time to find some universal mechanism that would allow solving any problem, at present, for each specific intelligent system a special problem solver is being developed, the composition of which is determined by a set of classes of problems that the appropriate intelligent system should solve. As a rule, each class of problems corresponds to some *problem-solving model*. Currently, in the field of artificial intelligence a large number of such models have been developed, some of which are considered as traditional (for example, classical algorithms, procedural and object-oriented programs) and some – as intelligent (neural network models, logical models, genetic algorithms).

The expansion of the scope of intelligent systems requires such systems to be able to solve complex problems, the solution of each of which involves the joint usage of many different knowledge representation models and various problem-solving models. In addition, the solution of complex problems implies the usage of common informational resources (in the limiting case – of the entire knowledge base of an intelligent system) by various components of the solver focused on solving various subproblems. Since the solver of complex problems integrates various problem-solving models, we will call it a *hybrid problem solver* [1].

Modern approaches to the construction of hybrid problem solvers, as a rule, involve a combination of heterogeneous problem-solving models without any single basis, for example, using specialized software interfaces between different components of the system, which leads to considerable overhead costs when developing such a system and especially when its modifying, including when adding a new problem-solving model to the system [1].

An approach to the development of hybrid solvers that allows them to be modifiable is proposed within the framework of the OSTIS Technology [2] and is considered in detail in some papers, in particular, in [1].

Within the framework of this approach, the problem solver is interpreted as a hierarchical system of agents (sc-agents) that work on shared semantic memory (sc-memory) and interact by the specification of the actions they perform within this memory. It is assumed that each problem-solving model corresponds to some sc-agent (most often – a non-atomic one that could be decomposed into simpler sc-agents). Thus, it becomes possible to combine different problem-solving models when solving the same complex problem as well as to add new problem-solving models to the solver or exclude them without having to make modifications in its other components.

However, the further development of this approach and, in particular, its usage when developing various applied intelligent systems has shown that the capabilities of the problem solver are also in large part determined by the quality of the knowledge base of the appropriate intelligent system. It may safely be said that the approach to the development of solvers discussed above is connected with the description of the *operational semantics* of the solver, that is, interpreters of the appropriate problem-solving models, while it is obvious that for solving problems it is also necessary to describe the *declarative semantics* of the problem-solving model, that is, the texts of programs itself (not the programs of sc-agents but higher-level programs interpreted by the corresponding set of sc-agents), logical statements, certain configurations of artificial neural networks, etc.

Within the framework of the OSTIS Technology, powerful tools have been developed that allow describing any type of knowledge in a unified form, structuring the knowledge base according to various criteria as well as verifying its quality and editing the knowledge base directly in its use [3]. The basis of the knowledge base created using the OSTIS Technology is a hierarchical system of subject domains and the corresponding ontologies. An ontology is interpreted as a specification of the system of concepts of the corresponding subject domain, while various types of ontologies are distinguished, each of which reflects a certain set of the concept features of the subject domain, for example, *terminological ontology*, *logical ontology*, *set-theoretic ontology*, etc. Speaking about ontologies in the context of this paper, we will have in mind an integrated ontology, which is a combination of ontologies of all types that correspond to a specific subject domain.

II. Proposed approach

Within the framework of this paper, it is proposed to take as a basis the approaches to the development of hybrid problem solvers and hybrid knowledge bases proposed within the context of the OSTIS Technology, to formally clarify and coordinate the interpretation of such concepts as *problem*, *problem-solving model*, *problem solver*, *skill* and others within the appropriate set of ontologies and on the basis of the results obtained to clarify the actual model of the hybrid problem solver, which would allow taking into account the abovementioned aspects.

The systems developed on the basis of the OSTIS Technology are called *ostis-systems*. The *OSTIS Technology* is based on a universal method of semantic representation (encoding) of information in memory of intelligent computer systems called *SC-code*. Texts of the *SC-code* (sc-texts) are unified semantic networks with a basic set-theoretic interpretation. The elements of such semantic networks are called *sc-elements* (*sc-nodes* and *sc-connectors*, which, in turn, can be *sc-arcs* or *sc-edges*, depending on the directivity). The *SC-code Alphabet* consists of five main elements, on the basis of which SC-code constructs of any complexity are built as well as more particular types of sc-elements (for example, new concepts) are introduced. Memory that stores the SC-code constructs is called semantic memory or *sc-memory*.

Within the framework of the technology, several universal variants of visualization of the *SC-code* constructs are also proposed, such as *SCg-code* (graphic version), *SCn-code* (non-linear hypertextual version), *SCs-code* (linear string version).

As it was mentioned earlier, the basis of the knowledge base within the framework of the OSTIS Technology is a hierarchical system of subject domains and ontologies. From there, to solve the problems set within the framework of this paper, it is proposed to develop a complex *Subject domain of actions and problems and the corresponding ontology of problem-solving methods and models*.

Within the framework of this paper, fragments of structured texts in the SCn-code [4] will often be used, which are simultaneously fragments of source texts of the knowledge base, which are understandable both to a human and to a machine. This allows making the text more structured and formalized while maintaining its readability. The symbol ":=" in such texts indicates alternative (synonymous) names of the described entity, which reveal in more detail some of its features.

The development of the specified family of sc-models of subject domains and ontologies will allow:

- explicitly linking the class of problems and the way (method) of its solution;
- this, in turn, will allow accumulating more complex components of solvers and massively simplify their integration, since the appropriate component combined with the group of sc-agents will also include the necessary fragments of the knowledge base, which are a priori squared with the specified group of sc-agents;
- this, in turn, will allow making the automation tools for the development of solvers more intelligent, in particular, it will allow automating the process of selecting solver components based on the specification of classes of problems that the designed intelligent system should be able to solve;
- in the future, this will allow the intelligent system to independently access the library of problem solver components and select components based on new classes of problems that the system has encountered, that is, it will allow the intelligent system to independently learn new *skills*;
- on the other hand, this approach will allow the intelligent system to independently select a combination of problem-solving models for solving problems of a certain class (more exactly, since the solver is based on a multiagent approach, a group of sc-agents that interpret different problem-solving models will be able to determine better, which of the sc-agents and

in what order should work when solving a specific complex problem).

The subject areas and ontologies discribed in this work were developed on the basis of the theory of subject-object influences proposed in the work of V. Martynov and his colleagues [5], [6], [7], [8], [9].

Consider next in more detail fragments of sc-models of specified subject domains and ontologies.

III. Concept of an action and the classification of actions

Before getting to the problem-solving models and the problem solver, it is necessary to formally clarify the concept of a problem and the concept of an action aimed at solving a particular problem or its subproblems.

Within the framework of the *OSTIS Technology*, we will interprete the problem as a formal specification of some action, so it is reasonable at first to clarify the concept of an *action*. Let us consider the specification of the concept *action* in the SCn-code.

action

:= [a purposeful process performed by one or more subjects (cybernetical systems) with the possible usage of certain tools]

:= [a process of influencing some (possibly shared) entity (the subject of influence) on one or several entities (objects of influence – source objects (arguments) or target (created or modified) objects)]

:= [an actio]

:= [an act]

:= [an operation]

:= [a conscious influence]

:= [an active influence]

⊂ *influence*

    := [a process, in which at least one influencing entity (the subject of influence′) and at least one entity that is being influenced (the object of influence′) can be clearly distinguished)]

    ⊂ *process*

:= [a purposeful ("conscious") process performed (managed, implemented) by some subject]

:= [a process of solving some problem]

:= [a purposeful process managed by some subject]

⇒ *decomposition**:

Decomposition of a class of actions in relation to memory of a cybernetical system

= {
- *informational action*
    - ⊃ *action in sc-memory*
- *behavioural action*
    - ⊃ *action in the environment of the ostis-system*
- *effector action*
    - ⊃ *effector action of the ostis-system*
- *receptor action*
    - ⊃ *receptor action of the ostis-system*

}

⊃ *atomic action*

    := [an action, the performance of which does not require its decomposition into a set of sub-actions (particular actions, actions of a lower level)]

    ⇒ *explanation**:

    [An atomic action is performed by a single individual subject and is either an atomic action performed in memory of this subject (an atomic action of its "processor") or an atomic action of one of its effectors.]

⊃ *complex action*

⇒ *subdividing**:

{
- *an action performed by a cybernetical system in its own memory*
- *an action performed by a cybernetical system in its environment*
- *an action performed by a cybernetical system on its physical shell*

}

The result of performing an ***informational action*** is generically a certain new state of information system memory (not necessarily of *sc-memory*) achieved only by transforming the information stored in system memory, that is, either by generating new knowledge based on existing ones or by deleting knowledge that has become unnecessary for whatever reason. It should be noted that if the question is about changing the state of *sc-memory*, then any transformation of information can be reduced to some atomic actions of generating, deleting or changing the incidence of *sc-elements* relative to each other.

In the case of a ***behavioral action***, the result of its performance will be a new state of the environment. It is very important to note that in this case the environment also means the components of the system that are external from the point of view of memory, that is, they are not information structures stored in it. Such components include, for example, various manipulators and other means of influencing the system on the external world, that is, behavioral problems can include changing the state of a mechanical limb of a robot or directly displaying some information on the screen for the user experience.

From the point of view of the problem solving formulated in this paper, the informational actions performed in memory of the ostis-system, that is, *actions in sc-memory*, promote outstanding interest. The classification of *actions in sc-memory* is presented in the knowledge base of the *IMS. ostis Metasystem* that describes the documentation of the current state of the *OSTIS Technology* [4].

On the set of actions a number of relations are set, such as *action subject′* (*performer′*), *customer**, *action object′*, *action context**, *sub-action**, *sequence of actions**, *result** and others [1], [4].

IV. Concept of a problem and the classification of problems

In turn, a *problem* will be interpreted as a specification of some action, within which, depending on the situation, the context of the action performance, the way of its performance, the performer, the customer, the planned result, etc. can be specified in advance using the relations listed above.

Let us consider the specification of the concept *problem* in the SCn-code.

problem
:= [a description of some desirable state or event either in the knowledge base or in the environment]
:= [a problem definition]
:= [a task for performing some action]
:= [a problem description]
:= [a problem situation]
:= [a specification of some action that has sufficient completeness to perform this action]

Each ***problem*** is a specification of an action that either has already been performed, or is currently being performed, or is planned (should) be performed, or can be performed (but not necessarily). Depending on the specific class of problems, both the internal state of the intelligent system itself and the required state of the environment can be described.

Classification of problems can be carried out on a didactic basis within each subject domain, for example, triangle problems, problems on sets of equations, etc.

Each *problem* can include:

- the fact that an *action* belongs to some particular class of *actions* (for example, *action. form a complete semantic neighborhood of the specified entity*), including the state of the *action* from the point of view of the life cycle (initiated, performed, etc.);
- a description of the *purpose** (*result**) of the *action*, if it is exactly known;
- specifying the action *customer**;
- specifying the action *performer** (including a collective one);
- specifying the *action argument(-s)'*;
- specifying a tool or mediator of the *action*;
- a description of the *action decomposition**;
- specifying a *sequence of actions** within the *action decomposition**, i.e., construction of a procedural plan for solving the problem. In other words, the construction of a solution plan is a decomposition of the corresponding *action* into a system of interconnected sub-actions;
- specifying the domain of the *action*;
- specifying the condition for initiating the *action*;
- the moment of the starting and ending the *action*, including the planned and actual ones, the expected and/or actual duration of the performance.

Some problems can be clarified further by the context – additional information about the entities considered in the *problem* definition, i.e., a description of what is given, what is known about these entities.

In addition, a *problem* can include any additional information about the action, for example:

- a list of resources and means that are supposed to be used in solving the problem, for example, a list of available performers, timescales, available funding, etc.;
- the restriction of the domain, in which the *action* is performed, for example, one *sc-construct* must be replaced by another according to some rule but only within some *knowledge base section*;
- the restriction of knowledge that can be used for solving a particular problem, for example, it is necessary to solve an algebra problem using only those statements that are included in the course of the school curriculum up to and including the seventh grade and not using statements studied in high school;
- etc.

As in the case of actions solved by the system, it is possible to classify *informational problems* and *behavioral problems*.

On the other hand, from the point of view of the problem definition, *declarative problem definitions* and *procedural problem definitions* can be distinguished. It should be noted that these classes of problems are not opposed to each other and there may be problem definitions that use both approaches.

problem
⊃ *procedural problem definition*
⊃ *declarative problem definition*
⊃ *question*
⊃ *command*
⊂ *knowledge*
⊃ *initiated problem*
  := [a problem definition to be performed]
⊃ *declarative problem definition*
⊃ *procedural problem definition*
⊃ *declarative-procedural problem definition*
  := [a problem, in the definition of which there are both declarative (target) and procedural aspects]
⊃ *problem solved in memory of a cybernetical system*
  ⊃ *problem solved in memory of an individual cybernetical system*
  ⊃ *problem solved in shared memory of a multiagent system*

:= [an informational problem]
:= [a problem aimed either at generation or search for information that meets the specified requirements or at some transformation of the specified information]
⊃ *mathematical problem*

The *problem* definition may not contain an indication of the context (solution domain) of the *problem* (in this case, the *problem* solution domain is either the entire *knowledge base* or its compliant part) and may also not contain either a description of the underlying situation or a description of the target situation. For example, a description of the target situation for an explicitly specified contradiction found in a *knowledge base* is not required.

Declarative problem definition is a description of the underlying (initial) situation, which is a condition for performing the corresponding action, and the target (final) situation, which is the result of performing this action, that is, a description of the situation (state) that should be achieved as a result of performing the planned action. In other words, such a problem definition includes an explicit or implicit description of:

- what is given – the source data, conditions for performing a specified action;
- what is required – the definition of the purpose and the result of performing the specified action.

In the case of the ***procedural problem definition***, the characteristic of the action specified by this problem is explicitly indicated, namely, for example:

- a subject or subjects that perform this action;
- objects, on which the action is performed – arguments of the action;
- tools that are used to perform the action;
- the moment and, possibly, additional conditions for starting and ending the action;
- a class or classes that each *action* belongs to (including sub-actions) are explicitly specified.

At the same time, it is not explicitly specified what should be the result of performing the corresponding action.

Let us note that, if necessary, the *procedural problem definition* can be reduced to the *declarative problem definition* by translating based on some rule, for example, of the definition of the class of actions through a more general class.

Particular types of problems are a *question* and a *command*.

question
:= [a request]
⊂ *problem solved in memory of a cybernetical system*
:= [a non-procedural problem definition for searching (in the current state of the knowledge base) or for generating knowledge that meets the specified requirements]
⊃ *question – what is it*
⊃ *question – why*
⊃ *question – wherefore*
⊃ *question – how*
:= [a request for a method (way) for solving a given (specified) type of problems or class of problems or a plan for solving a particular specified problem]
:= [a problem aimed at satisfying the information needs of a certain subject-customer]

command
:= [an initiated problem]
:= [a specification of the initiated action]

It should be noted that along with the given extremely general classification of problems, which inherently reflects the classes of problems from the point of view of their definition, there should be a classification of problems from the point of view of their semantics, that is, in terms of the essence of the specified action. This classification can be based on the classification presented in [10].

Within the framework of this paper, as already mentioned, the problems solved in sc-memory promote outstanding interest.

V. Concepts of a Class of Actions and a Class of Problems

From the point of view of the organization of the problem-solving process, the concepts of an *action* and a *problem* are not more important than the concepts of a *class of actions* and a *class of problems*, since it is for them that the appropriate performance algorithms and solution methods are being developed.

Let us define a ***class of actions*** as a maximal set of coincident (similar in a certain way) actions, for which there is (but is not necessarily currently known) at least one **method** (or mean) that provides the performance of any action from the specified set of actions.

class of actions
⇐ *family of subclasses**:
action
:= [a set of similar actions]
⊃ *class of atomic actions*
⊃ *class of easily performable complex actions*

Each distinguished *class of actions* corresponds to at least one common *method* for performing these *actions*. It means that the question is about semantic "clustering" of a set of *actions*, i.e., about the allocation of *classes of actions* on the basis of the semantic similarity (coincidence) of *actions* that are part of the selected *class of actions*.

In this case, first of all, the coincidence (similarity) of *underlying situations* and *target situations* of the *actions* being considered, i.e., the coincidence of *problems* solved as a result of performing the corresponding *actions*, is taken into account. Since one and the same *problem* can be solved as a result of performing several different *actions* that belong to different *classes of actions*, we should talk not only about *classes of actions* (sets of similar actions) but also about ***classes of problems*** (sets of similar problems) solved by these *actions*. For example, the following *relations* are set on the set of *classes of actions*:

- a *relation*, each bunding of which connects two different (disjoint) *classes of actions* that solve one and the same *class of problems*;
- a *relation*, each bunding of which connects two different *classes of actions* that solve different *classes of problems*, one of which is a *superset* of the other.

In addition to the class of actions, the concept of a ***class of atomic actions*** is also distinguished, that is, the set of atomic actions, the indication of belonging to which is a necessary and sufficient condition for performing this action. The set of all possible atomic actions performed by each subject should be divided into classes of atomic actions.

Belonging of some *class of actions* to the set of the ***classes of atomic actions*** fixes the fact that, when all the necessary arguments are specified, belonging of *action* to this class is sufficient for some subject to start performing this action.

At the same time, even if the *class of actions* belongs to the set of the ***class of atomic actions***, it is not forbidden to introduce more particular *classes of actions*, for which, for example, one of the arguments is fixed in advance.

If a specified ***class of atomic actions*** is more particular in relation to *actions in sc-memory*, this indicates that there is at least one *sc-agent* in the current version of the system that is focused on performing actions of this class.

In addition, it is also reasonable to introduce the concept of a *class of easily performable complex actions*, that is, a set of *complex actions*, for which at least one ***method*** is known and available, the interpretation of which allows performing a complete (final, ending with atomic actions) decomposition into sub-actions of each complex action from the above set.

Belonging of some *class of actions* to the set of the ***class of easily performable complex actions*** fixes the fact that, even when specifying all the necessary arguments of belonging the *action* to this class, it is unsufficient for some *subject* to start performing this action, and additional clarifications are required.

In turn, by the ***class of problems*** we will mean the set of problems, for which it is possible to construct a generalized definition of problems that corresponds to the whole set of problems. Each *generalized definition of the problems of the corresponding class* is in fact nothing more than a strict logical definition of the specified class of problems.

class of problems
⇐ *family of subsets**:
 problem

A specific class of actions can be defined in at least two ways.

class of actions
⇒ *subdividing**:
 {• ***class of actions that is precisely defined by the class of problems being solved***
  := [a *class of actions* that provide a solution of the corresponding *class of problems* and at the same time use a wide variety of *methods* for solving problems of this class]
 • ***class of actions that is precisely defined by the used method of solving problems***
 }

Further, let us consider in more detail the formal interpretation of the concept of a *method*.

VI. Concept of a method

By the method we will mean a description of how any or almost any (with explicit exceptions) action that belongs to the corresponding class of actions can be performed.

method
⇐ *second domain**:
 *method**
:= [a method for solving the corresponding class of problems that provides a solution of any or most of the problems of the specified class]
:= [a program for solving problems of the corresponding class, which can be both procedural and declarative (non-procedural) ones]
⊂ *knowledge*
∈ *type of knowledge*
:= [a way]
:= [a knowledge of how it is necessary to solve problems of the corresponding class of problems (a set of equivalent (similar) problems)]
:= [a method (way) for solving a certain (corresponding) class of problems]
:= [an information (knowledge) sufficient to solve any *problem* that belongs to the corresponding *class of problems* using the corresponding *problem-solving model*]

The specification of each *class of problems* includes a description of how to "bind" a *method* to the source data of a specific *problem* that is being solved using this *method*. The description of such a method of "binding" includes:

- a set of variables that are included both in the *method* and in the *generalized definition of problems of the corresponding class* and whose values are the corresponding elements of the source data of each specific problem being solved;
- a part of the *generalized definition of problems* of the class, to which the *method* being considered corresponds, which are a description of the conditions of usage of this *method*.

The very "binding" of the *method* to a specific *problem* being solved using this *method* is carried out by searching in the *knowledge base* for such a fragment that satisfies the conditions for using the specified *method*. One of the results of such a search is to establish a correspondence between the abovementioned variables of the used *method* and the values of these variables within the framework of a specific *problem* being solved.

Another option for establishing the correspondence being considered is an explicit appeal (call) of the corresponding *method* (program) with the explicit transmission of the corresponding parameters. But this is not always possible, because, when performing the process of solving a specific *problem* based on the declarative specification of performing this action, it is not possible to set:

- when it is necessary to initiate the call (use) of the required *method*;
- which specific *method* should be used;
- what parameters that correspond to the specific initiated *problem* should be transmitted for "binding" the used *method* to this *problem*.

The process of "binding" the *method* of solving *problems* to a specific *problem* solved using this *method* can also be represented as a process that consists of the following stages:

- construction of a copy of the used *method*;
- binding the main (key) variables of the used *method* with the main parameters of the specific *problem* being solved.

As a result, on the basis of the considered *method* used as a sample (template), a specification of the process for solving a specific problem – a procedural specification (*plan*) or a declarative one – is built.

Let us note that *methods* can be used even when constructing *plans* for solving specific *problems*, in the case when there is a need for multiple repetition of certain chains of *actions* with an a priori unknown number of such repetitions. It is question about various types of **cycles**, which are the simplest type of procedural *methods* for solving problems that are repeatedly used when implementing *plans* for solving some *problems*.

It is also obvious that several *methods* can correspond to one *class of actions*.

Thus, we assume that the term “method” is with the term “program” synonymous in the generalized sense of this term.

method
:= [a program]
:= [a program for performing actions of a certain class]
⊃ *procedural program*
  := [a generalized plan]
  := [a generalized plan for performing a certain class of actions]
  := [a generalized plan for solving a certain class of problems]
  := [a generalized specification of the decomposition of any action that belongs to a given class of actions]
  ⊂ *algorithm*

Let us consider in more detail the concept of a procedural program (procedural method). Each ***procedural program*** is a generalized plan for performing *actions* that belong to a certain class, that is, it is a *semantic neighborhood; the key sc-element* ′ is a *class of actions*, for the elements of which the process of their performance is additionally detailed.

The input parameters of the *procedural program* in the traditional sense correspond to the arguments that correspond to each *action* from the *class of actions* described by this *procedural program*. When generating a specific *plan* of performing a specific *action* from this class based on this program, these arguments take specific values.

Each *procedural program* is a system of described actions with an additional indication for the action:

- or a *sequence of actions** (transmission of initiation) when the condition for performing (initiating) actions is the performance of one of the specified or all of the specified actions;
- or an event in the knowledge base or the environment that is a condition for its initiation;
- or a situation in the knowledge base or the environment that is a condition for its initiation.

The concept of a method allows determining the relation *problem equivalence** on a set of problems. Problems are equivalent if and only if they can be solved by interpreting one and the same *method* (way) stored in memory of a cybernetical system.

Some *problems* can be solved by different *methods*, one of which, for example, is a generalization of the other. Thus, some relations can also be set on a set of methods.

Let us note that the concept of a *method* allows localizing the domain of solving problems of the corresponding class, that is, limiting the set of knowledge that is sufficient to solve problems of this class in a certain way. This, in turn, allows increasing the efficiency of the system as a whole, eliminating the number of unnecessary actions.

relation defined on a set of methods
∋ *submethod**
:= [a subprogram*]
:= [to be a method that is supposed to be used (accessed) when implementing a given method*]
⇔ *it is important to distinguish**:
*particular method**
:= [to be a method that provides a solution to a class of problems, which is a subclass of problems being solved using a given method*]

In the literature dedicated to the construction of problem solvers, the concept of a ***problem-solving strategy*** is found. Let us define it as a meta-method for solving problems that provides either the search for one relevant known method or the synthesis of a purposeful sequence of actions using various known methods in the general case.

problem-solving strategy
⊂ *method*

It can be said about a universal meta-method (universal strategy) for solving problems that explains all kinds of particular strategies.

In particular, we can talk about several global *strategies for solving informational problems* in knowledge bases. Let us assume that a sign of an initiated action with the definition of the appropriate informational purpose, i.e., a purpose aimed only at changing the state of the knowledge base, has appeared in the knowledge base. And the current state of the knowledge base does not contain a context (source data) sufficient to achieve the above purpose, i.e., such a context, for which there is a method (program) in the available package (set) of methods (programs), the usage of which allows achieving the above purpose. To achieve such a purpose, the context (source data) of which is insufficient, there are three approaches (three strategies):

- decomposition (reduction of the initial purpose to a hierarchical system and/or subpurposes (and/or subproblems) based on the analysis of the current state of the knowledge base and the analysis of what is missing in the knowledge base for using a particular method).
  At the same time, the most attention is paid to methods that require less effort to create conditions for using them. Ultimately, we must reach (at the lowest level of the hierarchy) subpurposes, the context of which is sufficient for the usage of one of the available methods (programs) for solving problems;
- generation of new knowledge in the semantic neighborhood of the definition of the initial purpose using <u>any</u> available methods in the hope of obtaining such a state of the knowledge base that will contain the necessary context (sufficient source data) to achieve the initial purpose using any available method of solving problems;
- combination of the first and second approaches.

Similar strategies exist for finding ways to solve problems being solved in the environment.

VII. Specification of Methods and the Concept of a Skill

Each specific *method* is considered by us not only as an important type of specification of the corresponding class of problems but also as an object that itself needs a specification that provides direct usage of this method. In other words, the method is not only a specification (the specification of the corresponding class of problems) but also an <u>object</u> of the specification. The most important type of such specification is the indication of the *operational semantics of the method*.

operational semantics of the method*
⊂ *specification**
:= [a family of methods that provide interpretation of a given method*]
:= [a formal description of the interpreter of a given method*]
⇒ *second domain**:
operational semantics of the method
⊃ ***complex representation of the operational semantics of the method***
:= [a representation of the *operational semantics of the method* brought (detailed) to the level of all *specifications of atomic actions* performed during the interpretation of the corresponding *method*]

declarative semantics of the method*
⊂ *specification**
:= [a description of the system of concepts that are used within the framework of this method*]

The relation *declarative semantics of the method** connects the *method* and the formal description of the system of concepts (a fragment of the *logical ontology* of the corresponding *subject domain*) that are used (mentioned) within this method. This is necessary for ensuring that one and the same concept is interpreted unambiguously within the framework of the method and the rest of the knowledge base, which is especially important when borrowing a method from a library of reusable components of problem solvers. It is important to note that the fact that any concepts are used within the framework of the method does not mean that the formal record of their definitions is part of this method. For example, a method that allows solving problems for calculating the area of a triangle will include various formulas for calculating the area of a triangle but will not include the definitions of the concepts ‘area", ‘triangle", etc., since if there are a priori correct formulas, these definitions will not be used directly in the process of solving the problem. At the same time, the formal definitions of these concepts will be part of the declarative semantics of this method.

Combining the *method* and its operational semantics, that is, information about how this *method* should be interpreted, we will call a ***skill***.

skill
:= [an ability]
:= [a combination of a *method* with its comprehensive specification – a *complex representation of the operational semantics of the method*]
:= [a method + a method of its interpretation]
:= [an ability to solve the corresponding class of equivalent problems]
:= [a method plus its operational semantics, which describes how this method is interpreted (performed, implemented) and is at the same time the operational semantics of the corresponding problem-solving model]
⇒ *subdividing**:
{• *active skill*
:= [a self-initiating skill]
• *passive skill*
}

Thus, the concept of a *skill* is the most important concept from the point of view of constructing problem solvers, since it combines not only the declarative part of the description of the method of solving a class of problems but also the operational one.

Skills can be *passive skills*, that is, such *skills*, the usage of which must be explicitly initiated by some agent, or *active skills*, which are initiated independently when a corresponding situation occurs in the knowledge base. To do this, in addition to the *method* and its operational semantics, the *sc-agent*, which responds to the appearance of a corresponding situation in the knowledge base and initiates the interpretation of the *method* of this *skill*, is also included in the *active skill*.

This separation allows implementing and combining different approaches for solving problems, in particular, *passive skills* can be considered as a way to implement the concept of a smart software package.

VIII. Concepts of a class of methods and a language for representing methods

Like actions and problems, methods can be classified into different classes. We will define a set of methods, for which it is possible to unify the representation (specification) of these methods, as a ***class of methods***.

class of methods
⇐ *family of subclasses**:
method
:= [a set of methods, for which the representation language of these methods is set]
∋ *procedural method for solving problems*
⊃ *algorithmic method for solving problems*
∋ *logical method for solving problems*
⊃ *productional method for solving problems*
⊃ *functional method for solving problems*
∋ *artificial neutral network*
:= [a class of methods for solving problems based on artificial neural networks]
∋ *genetic "algorithm"*
:= [a set of methods based on a common ontology]
:= [a set of methods represented in the same language]
:= [a set of methods for solving problems, which corresponds to a special language (for example, an sc-language) that provides a representation of methods from this set]
:= [a set of methods that corresponds to a separate problem-solving model]

Each specific *class of methods* mutually identically corresponds to a *language for representing methods* that belong to this (specified) *class of methods*. Thus, the specification of each *class of methods* is reduced to the specification of the corresponding *language for representing methods*, i.e., to the description of its syntactic, denotational and operational semantics.

Examples of *languages for representing methods* are all *programming languages*, which mainly belong to the subclass of *languages for representing methods* – to *languages for representing methods for information processing*. But now the need to create effective formal languages for representing methods for performing actions in the environment of cybernetical systems is becoming increasingly relevant. Complex automation, in particular, in the industrial sphere, is impossible without this.

There can be a whole set of such specialized languages, each of which will correspond to its own model of problem solving (i.e., to its own interpreter).

language for representing methods
:= [a method language]

:= [a language for representing methods that correspond to a specific class of methods]
⊂ *language*
:= [a programming language]
⊃ *language for representing methods for information processing*
  := [a language of representing methods for solving problems in memory of cybernetical systems]
⊃ *language of representing methods for solving problems in the environment of cybernetical systems*
  := [a programming language for external actions of cybernetical systems]

IX. Concept of a problem-solving model

By analogy with the concept of a problem-solving strategy, we introduce the concept of a ***problem-solving model***, which we will interpret as a meta-method for interpreting the corresponding class of methods.

problem-solving model
⊂ *method*
:= [a meta-method]
:= [an abstract machine for interpreting the corresponding class of methods]
:= [a hierarchical system of "microprograms" that provide interpretation of the corresponding class of methods]
⊃ *algorithmic problem-solving model*
⊃ *procedural parallel synchronous problem-solving model*
⊃ *procedural parallel asynchronous problem-solving model*
⊃ *productional problem-solving model*
⊃ *functional problem-solving model*
⊃ *logical problem-solving model*
  ⊃ *coherent logical problem-solving model*
  ⊃ *fuzzy logical problem-solving model*
⊃ *"neural network" problem-solving model*
⊃ *"genetic" problem-solving model*

Each *problem-solving model* is defined by:
- the corresponding class of methods for solving problems, i.e., the language of representing methods of this class;
- the subject domain of this class of methods;
- the ontology of this class of methods (i.e., the denotational semantics of the language of representing these methods);
- the operational semantics of the specified class of methods.

It is important to note that for the interpretation of all problem-solving models, an agent-oriented approach considered in [1] can be used.

specification*
⊃ ***problem-solving model****
  = *narrowing the relation by the first domain(specification*; class of methods)**
  := [a specification of the *class of methods**]
  := [a specification of the *language for representing methods**]

The problem-solving model associates the syntax, denotational and operational semantics of the language for representing methods of the corresponding class with a certain class of methods.

denotational semantics of the language for representing methods of the corresponding class
:= [an ontology of the corresponding class of methods]
:= [the denotational semantics of the corresponding class of methods]
:= [the denotational semantics of a language (an sc-language) that provides a representation of methods of the corresponding class]
:= [the denotational semantics of the corresponding problem-solving model]
⇒ *note**:
  [If the question is about a language that provides an internal representation of the methods of the corresponding class in the ostis-system, the syntax of this language coincides with the syntax of the sc-code**]**
⊂ *ontology*

operational semantics of the language for representing methods of the corresponding class
:= [a meta-method of interpretation of the corresponding class of methods]
:= [a family of agents that provide interpretation (usage) of any method that belongs to the corresponding class of methods]
:= [the operational semantics of the corresponding problem-solving model]

Since each *method* corresponds to a *generalized definition of problems* solved using this *method*, then each *class of methods* must correspond not only to a certain *language of representing methods* that belong to the specified *class of methods* but also to a certain *language of representation of generalized definitions of problems for various classes of problems* that are solved using *methods* that belong to the specified *class of methods*.

X. Concepts of a problem solver and a knowledge processing machine

Taking into account the system of concepts discussed above, we will define the problem solver of an intelligent computer system as a set of *skills* that allow the system to solve problems of a particular class.

problem solver
:= [a problem solver of an intelligent computer system]
:= [a set of all the skills (abilities) acquired by the computer system by now]
⊃ *combined problem solver*
⊃ *hybrid problem solver*

By the ***combined problem solver*** we will mean a solver that provides all the functionality of an intelligent system, that is, the solution of all problems that are related to the direct purpose of the system and ensure the efficiency of its work. Thus, for example, a solver that implements some variant of logical inference cannot be considered as combined, since to use a system that contains such a solver it is necessary to have at least basic information search tools that allow localizing the received answer as well as means that ensure the translation of a question from the user to the system and an answer from the system to the user.

In general, the *combined problem solver*, in contrast to the *problem solver* in a general sense, solves problems related to:

- ensuring the main functionality of the system (solving explicitly defined problems on demand);
- ensuring the correctness and optimization of the system (permanently throughout the system life cycle);
- ensuring the automation of the development of an intelligent system.

By the ***hybrid problem solver*** we will mean a *problem solver*, within which several different *problem-solving models* are used. It is obvious that the *combined problem solver* is predominantly a *hybrid problem solver*, since for the functioning of even a fairly simple intelligent system it is necessary to solve problems of fundamentally different classes discussed above.

In turn, by a *knowledge processing machine* we will mean the set of interpreters of all skills that build some *problem solver*. Taking into account the approach to information processing used within the framework of the OSTIS Technology and discussed in [1], a *knowledge processing machine* is a *sc-agent* (most often – a *non-atomic sc-agent*), which includes simpler sc-agents that provide interpretation of the corresponding set of *methods*.

Thus, we can talk, for example, about a *deductive logical inference machine* or an *information search machine*.

XI. Example of the usage of the developed ontologies

Let us consider the usage of the abovementioned fragments of ontology on the example of the description in the knowledge base of ways to solve a simple problem – the problem of finding roots of a quadratic equation.

As it is known from the school course in mathematics, the problems of this class can be solved in at least two ways – through the discriminant and the Vieta formulas for the quadratic equation.

On the other hand, from the point of view of implementation in the ostis-system, both of these options can also be implemented in two ways:

- in a particular way when an *abstract sc-agent* designed to solve problems of a specific class in a specific way is being developed;
- in a more general way when the corresponding formulas are written in the form of logical rules, which are further interpreted by a group of domain-independent sc-agents. This option is worse in terms of performance but much better in terms of flexibility and extensibility of the system.

Figure 1 shows an example of implementing the solution of problems of the considered class in both ways (through the discriminant and Vieta formulas) in a more particular way in the form of active skills. In this variant, it is assumed that the sc-agent programs are implemented in the SCP language, which is the basic language for processing SC-code texts and whose programs are also written in the SC-code. Based on this, the operational semantics of the methods is an *Abstract scp-machine*, that is, an interpreter of SCP programs.

In turn, figure 2 shows an example of the implementation of the same skills but in a more general way and in the form of passive skills. In this case, the operational semantics of the methods is a *Non-atomic abstract sc-agent of logical inference*, which allows using logical statements and, if necessary, calculating mathematical expressions that are obtained as a result of using a logical statement.

The presented figures also show how the declarative semantics of the corresponding methods are set.

XII. Conclusion

The paper considers an ontological approach to the development of problem solvers for intelligent computer systems based on the OSTIS Technology. The formal interpretation of such concepts as action, problem, class of actions, class of problems, method, skill is clarified, which together made it possible to define on their basis the concepts of a problem-solving model and a problem solver.

Examples of describing skills that allow solving one and the same class of problems in different ways are given.

In the future, the results obtained will increase the efficiency of the component approach to the development of problem solvers and automation tools for the development of problem solvers as well as provide an opportunity not only for the developer but also for the intelligent system to automatically select ways to solve a particular problem.

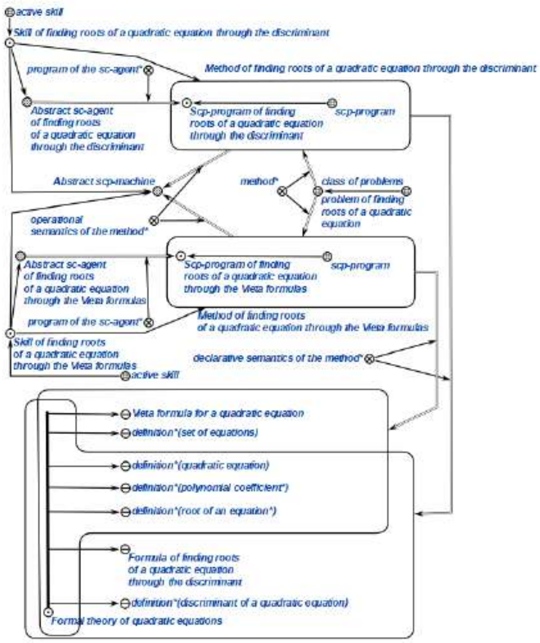


Figure 1. Example of usage of active skills

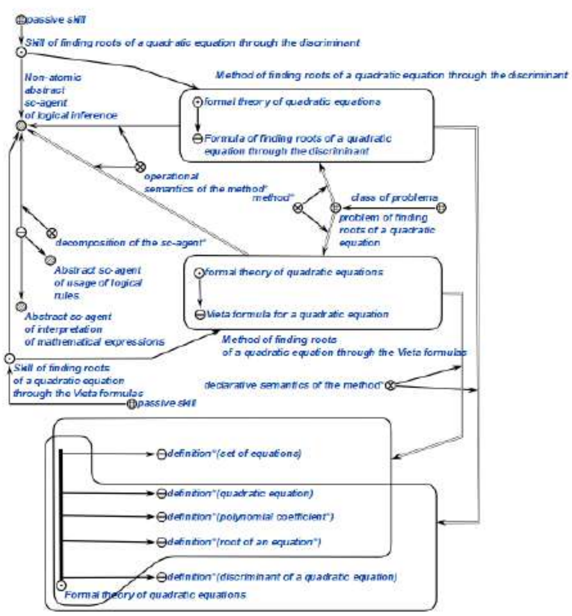


Figure 2. Example of usage of passive skills

Acknowledgment

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help in the work and valuable comments.

The work was carried out with the partial financial support of the BRFFR (BRFFR-RFFR No. F21RM-139).

References

[1] D. Shunkevich, "Agentno-orientirovannye reshateli zadach intellektual'nyh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 119–132.

[2] V. Golenkov, N. Guliakina, I. Davydenko, and A. Eremeev, "Methods and tools for ensuring compatibility of computer systems," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2019, pp. 25–52.

[3] I. Davydenko, "Semantic models, method and tools of knowledge bases coordinated development based on reusable components," in *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2018, pp. 99–118.

[4] (2021, Jun) IMS.ostis Metasystem. [Online]. Available: https://ims.ostis.net

[5] V. Martynov, *Semiologicheskie osnovy informatiki [Semiological Foundations of Informatics]*. Minsk: Nauka i tekhnika [Science and technics], 1974.

[6] ——, *Universal'nyi semanticheskii kod (Grammatika. Slovar'. Teksty) [Universal Semantic Code (Grammar. Dictionary. Texts)*. Minsk: Nauka i tekhnika [Science and technics], 1977.

[7] ——, *Universal'nyi semanticheskii kod: USK-3*.

[8] A. Hardzei, *Theory for Automatic Generation of Knowledge Architecture: TAPAZ-2. Transl. from Rus. I. M. Boyko. Rev. English edn.* Minsk: The Republican Institute of Higher School Publ., 2017.

[9] ——, "Plagiarism problem solving based on combinatory semantics," in *Open Semantic Technologies for Intelligent System*, V. Golenkov, V. Krasnoproshin, V. Golovko, and E. Azarov, Eds. Cham: Springer International Publishing, 2020, pp. 176–197.

[10] A. Fayans and V. Kneller, "About the ontology of task types and methods of their solution," *Ontology of designing*, vol. 10, no. 3, pp. 273–295, Oct. 2020. [Online]. Available: https://doi.org/10.18287/2223-9537-2020-10-3-273-295

Онтологический подход к разработке гибридных решателей задач интеллектуальных компьютерных систем

Шункевич Д.В.

В работе рассмотрен онтологический подход к разработке решателей задач интеллектуальных компьютерных систем на основе Технологии OSTIS. Уточнена формальная трактовка таких понятий как действие, задача, класс действий, класс задач, метод, навык, что в совокупности позволило определить на их основе понятие модели решения задач и решателя задач. Полученные результаты позволят повысить эффективность компонентного подхода к разработке решателей задач и средств автоматизации разработки решателей задач.

Received 01.06.2021

Ontological approach to the development of a software model of a semantic computer based on the traditional computer architecture

Daniil Shunkevich, Denis Koronchik
Belarussian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: shunkevich@bsuir.by, denis.koronchik@gmail.com

***Abstract*—The paper considers an ontological approach to the development of a software model of a platform for interpreting semantic models of intelligent computer systems (a software model of a semantic computer). The architecture of the specified software model and its components are considered in detail, the principles of their implementation and the advantages of the decisions made in comparison with analogues are indicated.**

A distinctive feature of the work is the demonstration of the usage of the ontological approach to the development of software products on the example of the specified software model of a semantic computer.

***Keywords*—ontological approach, OSTIS, semantic computer, crossplatform development, graph database, semantic networks**

I. Introduction

Throughout the development of the field of artificial intelligence, attempts to create specialized hardware solutions designed to interpret a certain class of models, for example, neural network [1] or logical [2] ones, have been repeatedly made. Many of these attempts were unsuccessful, while others led to the industrial production and active usage of such specialized tools [3].

One of the main reasons for failures of such attempts was the absence of an appropriate technology that would allow implementing actively new hardware solutions in the development of intelligent systems that use models interpreted using these hardware tools. That is to say, the absence of a large number of systems, for which the usage of these models would obviously be in demand, as well as the absence of a technology that would allow developing such systems using new hardware solutions within a reasonable time were the obstacle at that time. Confirming this idea, we can introduce the relation between the rapid development of neural network models in recent years and the subsequent development of specialized hardware solutions that are used widely [3].

One of the ways that allow testing, developing and in some cases implementing new models and technologies, regardless of the availability of appropriate hardware tools, is the development of software models of these hardware tools that would be functionally equivalent to these hardware tools but at the same time would be interpreted on the basis of the traditional hardware architecture (in this article we will consider the von Neumann architecture as the dominant one at present). It is obvious that the efficiency of such software models will generally be lower than for the hardware solutions, but in most cases it is sufficient to develop the appropriate technology along with the development of hardware tools and to transfer gradually already working systems from the software model to the hardware one.

These ideas were considered when creating the *OSTIS Technology*, one of the key principles of which is the focus on a fundamentally new hardware basis for the development of intelligent computer systems – ***a semantic computer*** [4]. Currently, along with the work on creating a semantic computer, an active implementation of its software model is underway, which is currently actively used in the development of intelligent computer systems for various purposes.

The work on this software model has been carried out for a long time, and some results were published earlier by the authors [5], [6]. In contrast to these papers, the key emphasis in this paper is made on the ontological approach to the development of various kinds of products on the example of a software model of a semantic computer.

The ontological approach to software development is currently being explored as part of the Ontology Driven Software Development trend [7]. The main advantages of this approach are associated with the ability to automate the processes of analysis, development and evolution of models of software systems without taking into account the peculiarities of their implementation on specific platforms, which in turn increases the flexibility of such systems and the efficiency of their evolution (maintenance).

In this paper the *OSTIS Technology* will also be used as the basis for the ontological approach, which, in turn, illustrates such its property as reflexivity. Therefore, as part of this paper, fragments of structured texts in the SCn-code [8] (one of the *OSTIS Technology* standards) will often be used, which are simultaneously fragments of source texts of the knowledge base that are understandable both to a human and to a machine. This allows making the text more structured and formalized while maintaining its readability.

II. Problem definition

A. Architecture of ostis-systems

The *OSTIS Technology* is based on a universal method of semantic representation (encoding) of information in the memory of intelligent computer systems called *SC-code*. Texts in the *SC-code* (sc-texts) are unified semantic networks with a basic set-theoretic interpretation. The elements of such semantic networks are called *sc-elements* (*sc-nodes* and *sc-connectors*, which, in turn, can be *sc-arcs* or *sc-edges*, depending on the directivity). The *SC-code alphabet* consists of five main elements, on the basis of which SC-code constructs of any complexity are built, as well as more particular types of sc-elements (for example, new concepts) are introduced.

Within the framework of the technology, several universal versions of visualization of *SC-code* constructs are also proposed, such as *SCg-code* (graphic version), *SCn-code* (non-linear hypertextual version), *SCs-code* (linear string version).

Systems built on the basis of the *OSTIS Technology* are called *ostis-systems*. Each *ostis-system* consists of a complete model of this system described by means of the *SC-code* (*sc-model of a computer system*) and a *platform for interpreting sc-models*, which in general can be implemented both in software and in hardware [9]. This ensures full platform independence of *ostis-systems*.

In turn, the *sc-model of a computer system* is conventionally divided into the *sc-model of the knowledge base*, *sc-model of the problem solver* and *sc-model of the computer system interface* (both with user and with the environment and other ostis-systems) as well as the model of abstract semantic memory (*sc-memory*), in which *SC-code* constructs are stored, and, accordingly, all the listed *sc-models* (figure 1).

Due to the availability of the *SC-code Alphabet* and the possibility of a complete description of the system using the SC-code, it becomes possible to make ostis-systems completely platform-independent. Thus, the development of an ostis-system is reduced to the development of its model and is carried out independently not only of the operating system but also of the architecture of the computer, on which the system runs. The platform, in turn, can be implemented both in the software version (in fact, in the form of a virtual machine) and in the hardware version. Therefore, the most attention within

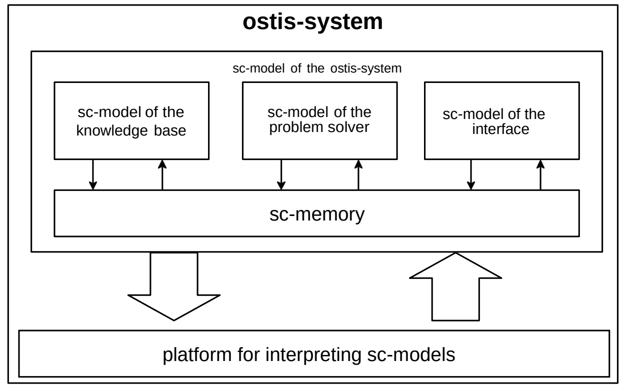


Figure 1. The architecture of the ostis-system

this article is paid to the software version of the platform implementation.

B. Principles which underlie the approach to the development of a software version of the implementation of the platform for interpreting sc-models

Since sc-texts are semantic networks, that is, in fact, graph constructs of a certain type, at the lower level, the problem of developing a software version of the implementation of the platform for interpreting sc-models is reduced to the development of means for storing and processing such graph constructs.

Currently, a large number of the simplest models for representing graph constructs in linear memory have been developed, such as incident matrices, adjacency lists and others [10]. However, when developing complex systems, as a rule, it is usually necessary to use more efficient models, both in terms of the amount of information required for representation and in terms of the efficiency of processing graph constructs stored in one form or another.

The most common software tools focused on storing and processing graph constructs include graph DBMS (Neo4j [11], ArangoDB [12], OrientDB [13], Grakn [14], etc.) as well as so-called rdf-storages (Virtuoso [15], Sesame [16], etc.) designed for storing constructs represented in the RDF model. To access information stored within such tools, both languages implemented within a specific tool (for example, the Cypher language in Neo4j) and languages that are standards for a large number of systems of this class (for example, SPARQL for rdf storages) can be used.

The popularity and development of such tools lead to the fact that at first glance it seems reasonable and effective to implement a *software version of the implementation of the platform for interpreting sc-models* based on one of these tools. However, there are a number of reasons why it was decided to implement a *software version of the implementation of the platform* *for interpreting sc-models* from scratch. These include the following ones:

- to ensure the efficiency of storing and processing information constructs of a certain type (in this case – SC-code constructs, sc-constructs), the specificity of these constructs must be taken into consideration. Particularly, at that time, the experiments described in [5] showed a significant increase in the efficiency of the own solution compared to the existing ones;
- in contrast to classical graph constructs, where an arc or an edge can be incident only to the graph vertex (this is also true for rdf-graphs), for the SC-code, it is quite typical when an sc-connector is incident to another sc-connector or even two sc-connectors. In this regard, the existing means of storing graph constructs do not allow storing sc-constructs (sc-graphs) explicitly. A possible solution to this problem is the transition from the sc-graph to the incident orgraph, an example of which is described in [17], however, this option leads to the multiplication of stored elements by several times and significantly reduces the efficiency of search algorithms due to the need to do a large number of additional iterations;
- the basis of information processing within the OSTIS Technology is a multiagent approach, in which agents of processing information stored in sc-memory (sc-agents) react to events that occur in sc-memory and exchange information by specifying the actions they perform in sc-memory [18]. Therefore, one of the most important problems is to implement within the *software version of the implementation of the platform for interpreting sc-models* the possibility of subscribing to events that occur in the software model of sc-memory, which is currently not practically supported within the modern means of storing and processing graph constructs;
- the SC-code also allows describing external information constructs of any kind (images, text files, audio and video files, etc.), which are formally interpreted as the contents of *sc-elements* that are signs of *external files of the ostis-system*. Thus, a component of the *software version of the implementation of the platform for interpreting sc-models* should be the implementation of file memory, which allows storing these constructs in any generally accepted formats. The implementation of such a component within the modern means of storing and processing graph constructs is also not always possible.

Due to all reasons outlined, it was decided to implement a *software version of the implementation of the platform for interpreting sc-models* "from scratch", taking into account the features of information storing and processing within the OSTIS Technology. Further, the architecture of the platform implementation, the principles of storing sc-constructs in traditional linear memory as well as the implementation of tools for accessing and editing constructs stored in the sc-memory software model will be considered in detail.

III. General architecture of the software version of the implementation of the platform for interpreting sc-models

Let us consider the specification of the concept *software version of the implementation of the platform for interpreting sc-models of computer systems* in the SCn-code. Within this and other fragments, the symbol ":=" indicates alternative (synonymous) names of the described entity, which reveal in more detail some of its features.

software version of the implementation of the platform for interpreting sc-models of computer systems
:= [a software version of the implementation of the basic interpreter of logical-semantic models of computer systems]
:= [a version of the implementation of the basic interpreter of logical-semantic models of computer systems on traditional computers with the von Neumann architecture]
⊃ *web-oriented version of the implementation of the platform for interpreting sc-models of computer systems*
  ⊂ *multiuser version of the implementation of the platform for interpreting sc-models of computer systems*
  ∋ *Software version of the implementation of the platform for interpreting sc-models of computer systems*

Software version of the implementation of the platform for interpreting sc-models of computer systems
⇒ *decomposition of the software system**:
  { • *Software model of sc-memory*
    • *Implementation of the interpreter of sc-models of user interfaces*
  }

The current *Software version of the implementation of the platform for interpreting sc-models of computer systems* is a web-oriented one, that is, in terms of the modern architecture, each ostis-system is a website that is accessible online through a usual browser. Such version of the implementation has an obvious advantage – an access to the system is possible from anywhere in the world where there is an Internet connection, while no specialized software is required to work with the system. On the other hand, this version of the implementation provides the possibility to work with the system for several users parallely.

At the same time, the interaction of the client and server parts is organized in such a way that the web interface can be easily replaced with a desktop or mobile interface, both universal and specialized ones.

This version of the implementation is distributed under an open-source license; for storing source texts, the Github hosting and a collective ostis-dev account are used [19].

The implementation is crossplatform and can be compiled from source texts in various operating systems.

Figure 2 shows the current architecture of the platform for interpreting sc-models of computer systems.

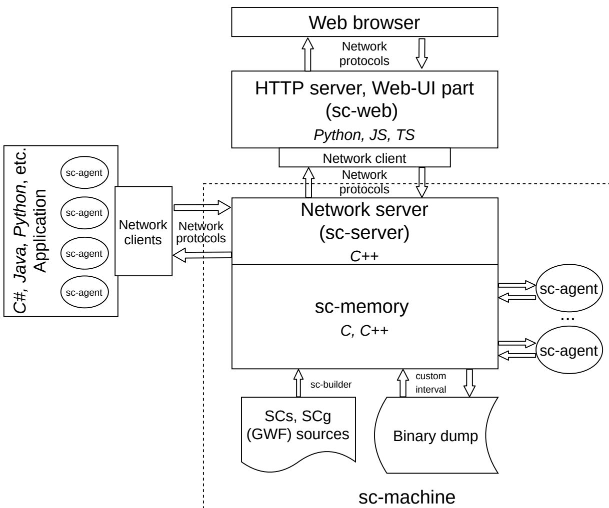


Figure 2. The architecture of the platform for interpreting sc-models of computer systems

The illustration above shows that the core of the platform is a *Software model of sc-memory* (sc-machine), which can simultaneously interact with both the *Implementation of the interpreter of sc-models of user interfaces* (sc-web [20]) and with any third-party applications using the corresponding network protocols. In terms of the general architecture, *Implementation of the interpreter of sc-models of user interfaces* acts as one of the many possible external components that interact with the *Software model of sc-memory* over the network.

Software model of sc-memory
:= [A software model of semantic memory implemented on the basis of traditional linear memory and that include storage facilities for sc-constructs and basic tools for processing these constructs, including ones for the remote access to them via appropriate network protocols]
⇐ *software model*:*
sc-memory
∈ *software model of sc-memory based on linear memory*
⇒ *component of the software system*:*
- *Implementation of the sc-storage and means of access to it*
- *Implementation of a basic set of platform-dependent sc-agents and their common components*
- *Implementation of the subsystem of interaction with the environment using network protocols*
- *Implementation of auxiliary tools for working with sc-memory*
- *Implementation of the scp-interpreter*

Within the current *Software model of sc-memory* [21], an *sc-storage* is understood as a component of the software model that stores sc-constructs and accesses them through a program interface. In general, the *sc-storage* can be implemented in different ways. In addition to the very *sc-storage*, the *Software model of sc-memory* also includes the *Implementation of file memory of the ostis-system* designed to store the contents of *internal files of ostis-systems*. It is worth noting that when switching from the *Software model of sc-memory* to its hardware implementation, it will be reasonable to implement file memory of the ostis-system on the basis of traditional linear memory (at least, at the first stages of the development of a *semantic computer*).

The current version of the *Software model of sc-memory* assumes the possibility of saving the memory state (snapshot) to the hard disk and further loading it from the previously saved state. This ability is necessary for restarting the system, in case of possible failures as well as when working with source texts of the knowledge base when the building from source texts is reduced to creation of a snapshot of the memory state, which is then placed in the *Software model of sc-memory*.

IV. Principles of implementation of the sc-storage

Let us consider the specification of the entity *Implementation of the sc-storage and means of access to it* in the SCn-code:

Implementation of the sc-storage and means of access to it
⇒ *component of the software system*:*
- *Implementation of the sc-storage*
  ∈ *implementation of the sc-storage based on linear memory*
  ⇒ *class of software system objects*:*
  segment of the sc-storage
  := [a page of the sc-storage]
  ⇒ *generalized part*:*
  element of the sc-storage
- *Implementation of file memory of the ostis-system*

Within this implementation of the *sc-storage*, *sc-memory* is modeled as a set of *segments*, each of which is a fixed-sized ordered sequence of *elements of the sc-storage*, each of which corresponds to a specific sc-element. Currently, each segment consists of $2^{16} - 1 = 65535$ *elements of the sc-storage*. The allocation of *segments of the sc-storage* allows, on the one hand, simplifying an address access to *elements of the sc-storage* and, on the other hand, realizing the possibility of unloading a part of sc-memory from RAM to the file system if necessary. In the second case, the sc-storage segment becomes the minimal (atomic) unloaded part of sc-memory. The mechanism for unloading segments is implemented by the existing principles of organizing virtual memory in modern operating systems.

The maximal possible number of segments is limited by the settings of the software implementation of the sc-storage (currently, the default number is $2^{16} - 1 = 65535$ segments, but in general it may be different). Thus, technically, the maximal number of stored sc-elements in the active implementation is about 4.3×10^9 sc-elements.

By default, all segments are physically located in RAM, if there is not enough memory amount, then a mechanism for unloading part of the segments to the hard disk (the virtual memory mechanism) is provided.

Each segment consists of a set of data structures that describe specific *sc-elements* (*elements of the sc-storage*). Regardless of the type of the sc-element being described, each *element of the sc-storage* has a fixed size (currently – 48 bytes), which ensures the convenience of storing them. Thus, the maximal size of the knowledge base in the current software model of sc-memory can reach 223 GB (without taking into account the contents of *internal files of the ostis-system* stored on the external file system).

The described structure of the *Implementation of the sc-storage* is illustrated in figure 3.

Figure 4 shows an example of encoding information in the *Implementation of the sc-storage* written in the SCg-code (a graphical version of the visualization of SC-code texts). For clarity, *labels of the access level* in this example are omitted.

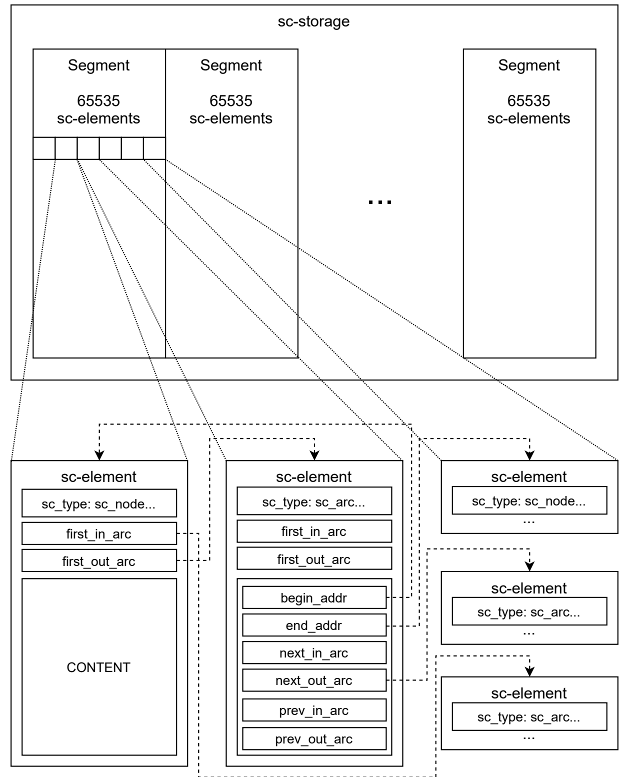


Figure 3. An example of encoding information in the Implementation of the sc-storage

sc-address

:= [an address of the element of the sc-storage that corresponds to the specified sc-element, within the current state of the Implementation of the sc-storage as part of the software model of sc-memory]

⇒ *family of relations that precisely define the structure of a given entity**:
- *segment number of the sc-storage**
- *number of the element of the sc-storage within the segment**

Each element of the sc-storage in the current implementation can be precisely specified by its address (sc-address) that consists of a segment number and a number of the *element of the sc-storage* within the segment. Thus, the *sc-address* serves as the unique coordinates of the *element of the sc-storage* within the *Implementation of the sc-storage*.

The sc-address is not taken into account in any way when processing the knowledge base at the semantic level and is only necessary to provide access to the corresponding data structure stored in linear memory at the level of the *Implementation of the sc-storage*.

In general, the sc-address of the element of the sc-storage that corresponds to the specified sc-element may change, for example, when rebuilding the knowledge base from source texts and then restarting the system. In this case, the sc-address of the element of the sc-storage that corresponds to the specified sc-element cannot change directly during the activity of the system in the current implementation.

For simplicity, we will use the term "sc-address of the sc-element", meaning the *sc-address* of the *element of the sc-storage* that uniquely corresponds to this *sc-element*.

element of the sc-storage

:= [a cell of the sc-storage]

:= [an element of the sc-storage that corresponds to the sc-element]

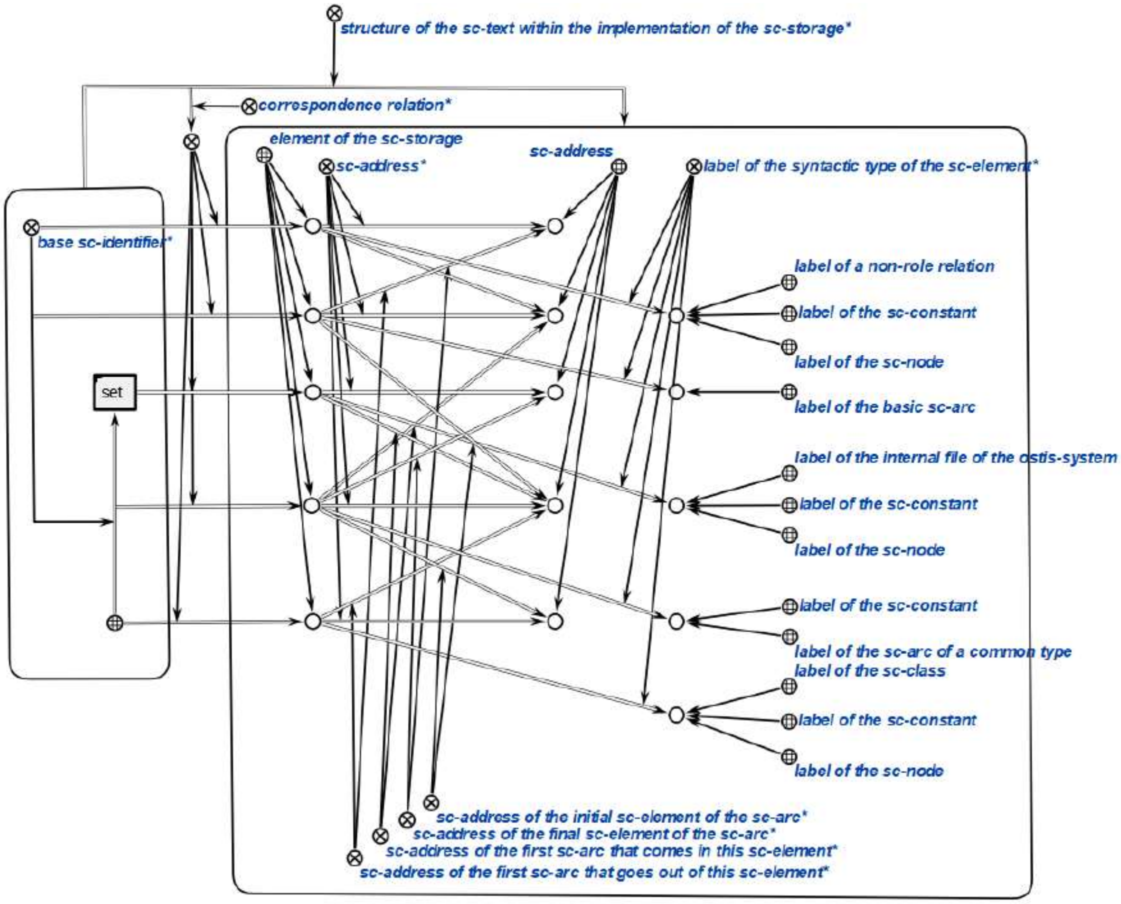


Figure 4. The Structure of the Implementation of the sc-storage

:= [an image of the sc-element within the sc-storage]
:= [a data structure, each instance of which corresponds to one sc-element within the sc-storage]
⇒ *subdividing**:
{ • *element of the sc-storage that corresponds to the sc-node*
• *element of the sc-storage that corresponds to the sc-arc*
}

element of the sc-storage that corresponds to the sc-node
⇒ *family of relations that uniquely define the structure of a given entity**:
{ • *label of the syntactic type of the sc-element**
• *label of the access level of the sc-element**
• *sc-address of the first sc-arc that goes out of this sc-element**
• *sc-address of the first sc-arc that comes in this sc-element**
• *contents of the element of the sc-storage**
⇒ *second domain**:
contents of the element of the sc-storage
:= [the contents of the element of the sc-storage that corresponds to the internal file of the ostis-system]
}

element of the sc-storage that corresponds to the sc-arc
⇒ *family of relations that uniquely define the structure of a given entity**:
{ • *label of the syntactic type of the sc-element**
• *label of the access level of the sc-element**
• *sc-address of the first sc-arc that goes out of this sc-element**
• *sc-address of the first sc-arc that comes in this sc-element**
• *specification of the sc-arc within the sc-storage**
⇒ *second domain**:
specification of the sc-arc within the sc-storage
}

specification of the sc-arc within the sc-storage
⇒ *family of relations that uniquely define the structure of a given entity**:
{• *sc-address of the initial sc-element of the sc-arc**
• *sc-address of the final sc-element of the sc-arc**
• *sc-address of the next sc-arc that goes out of the same sc-element**
• *sc-address of the next sc-arc that comes in the same sc-element**
• *sc-address of the previous sc-arc that goes out of the same sc-element**
• *sc-address of the previous sc-arc that comes in the same sc-element**
}

Each element of the sc-storage that corresponds to a certain sc-element is described by its syntactic type (label), and, regardless of the type, sc-addresses of the first sc-arc that comes in this sc-element and the first sc-arc that goes out of this sc-element (they can be empty if there are no such sc-arcs) are indicated.

The remainder of bytes, depending on the type of the corresponding sc-element (sc-node or sc-arc), can be used either to store the contents of the internal file of the ostis-system (it can be empty if the sc-node is not a file sign) or to store the specification of the sc-arc.

The *sc-address of the first sc-arc that goes out of this sc-element**, the *sc-address of the first sc-arc that comes in this element** and the *contents of the element of the sc-storage* may generally be missing (to be "empty", with value zero), but the size of the element in bytes will remain the same.

Each sc-node in the current implementation can have the contents (it can become an *internal file of the ostis-system*). If the size of the contents of the internal file of the ostis-system does not exceed 48 bytes (the size of the *specification of the sc-arc within the sc-storage*, for example, a small *string sc-identifier*), then this contents is explicitly stored within the element of the sc-storage as a sequence of bytes.

Otherwise, it is placed in a specially organised file memory (for its organisation a separate platform module is responsible, which in general can be arranged differently), and a unique address of the corresponding file is stored within the element of the sc-storage, which allows finding it on the file system quickly.

Currently, sc-edges are stored in the same way as sc-arcs, that is, they have a begin sc-element and an end one, the difference is only in the *label of the syntactic type of the sc-element*. This leads to some inconveniences during processing, but sc-edges are currently used quite rarely.

In terms of the software implementation, the data structure for storing the sc-node and the sc-arc remains the same, but the list of fields (components) changes in it.

In addition, as it can be seen, each element of the sc-storage (including the *element of the sc-storage that corresponds to the sc-arc*) does not store a list of sc-addresses of the sc-elements connected with it but stores sc-addresses of one outgoing and one incoming arcs, each of which, in turn, stores sc-addresses of the next and previous arcs in the list of outgoing and incoming sc-arcs for the corresponding elements.

All of the above allows:

- making the size of such a structure fixed (currently – 48 bytes) and independent of the syntactic type of the stored sc-element;
- providing the ability to work with sc-elements without taking into account their syntactic type in cases where it is necessary (for example, when implementing search requests such as “Which sc-elements are elements of this set”, “Which sc-elements are directly related to this sc-element”, etc.);
- providing the ability to access the *element of the sc-storage* in a constant time;
- providing the ability to place the *element of the sc-storage* in the processor cache, which, in turn, allows speeding up the processing of sc-constructs.

The current *Software model of sc-memory* assumes that all sc-memory is physically located on one computer. To implement a distributed version of the *Software model of sc-memory*, it is proposed to extend the *sc-address* by specifying the address of the physical device where the corresponding *element of the sc-storage* is stored.

Obviously, the type (class, kind) of the sc-element in sc-memory can be set by explicitly specifying that this sc-element belongs to the corresponding class (sc-node, sc-arc, etc.).

However, within the *platform for interpreting sc-models of computer systems*, there must be some set of *labels of the syntactic type of the sc-element* that specify the type of the element at the platform level and do not contain a corresponding sc-arc of belonging (or rather, the basic sc-arc) explicitly stored within sc-memory (its occurrence is implied, but it is not stored explicitly, since this will lead to an infinite increase in the number of sc-elements that need to be stored in sc-memory). At a minimum, there must be a label that corresponds to the *basic sc-arc* class, since an explicit indication of belonging of the sc-arc to this class generates another *basic sc-arc*.

Thus, *basic sc-arcs* that denote the belonging of sc-elements to some known limited set of classes are represented implicitly. This fact must be taken into account in a number of cases, for example, when checking whether an sc-element belongs to a certain class, when searching for all outgoing sc-arcs from a given sc-element, etc.

If necessary, some of these implicitly stored sc-arcs can be represented explicitly, for example, in the case when such an sc-arc must be included in any set, that is, another sc-arc must be drawn into it. In this case, there is a need to synchronize the changes connected with this sc-arc (for example, with deleting it) in its explicit and implicit representation. This mechanism is not implemented in the current *Implementation of the sc-storage*.

Thus, it is impossible to completely stop using *labels of the syntactic type of the sc-element*, however, though an increase in their number raises the efficiency of the platform due to the simplifications of certain operations on the validation of types of the sc-element, but it leads to an increase in the number of situations, in which the explicit and implicit representation of sc-arcs must be taken into consideration, which, in turn, complicates the development of the platform and the development of the program code for processing the stored sc-structures.

label of the syntactic type of the sc-element

:= [a unique numerical identifier that precisely corresponds to the specified type of sc-elements and is attributed to the corresponding element of the sc-storage at the implementation level]

⇐ *second domain**:
*label of the syntactic type of the sc-element**

⊃ *label of the sc-node*
⇒ *numerical expression in the hexadecimal number system**:
[0x1]

⊃ *label of the internal file of the ostis-system*
⇒ *numerical expression in the hexadecimal number system**:
[0x2]

⊃ *label of the sc-edge of a common type*
⇒ *numerical expression in the hexadecimal number system**:
[0x4]

⊃ *label of the sc-arc of a common type*
⇒ *numerical expression in the hexadecimal number system**:
[0x8]

⊃ *label of the sc-arc of belonging*
⇒ *numerical expression in the hexadecimal number system**:
[0x10]

⊃ *label of the sc-constant*
⇒ *numerical expression in the hexadecimal number system**:
[0x20]

⊃ *label of the sc-variable*
⇒ *numerical expression in the hexadecimal number system**:
[0x40]

⊃ *label of the positive sc-arc of belonging*
⇒ *numerical expression in the hexadecimal number system**:
[0x80]

⊃ *label of the negative sc-arc of belonging*
⇒ *numerical expression in a hexadecimal number system**:
[0x100]

⊃ *label of the fuzzy sc-arc of belonging*
⇒ *numerical expression in the hexadecimal number system**:
[0x200]

⊃ *label of the permanent sc-arc*
⇒ *numerical expression in the hexadecimal number system**:
[0x400]

⊃ *label of the temporal sc-arc*
⇒ *numerical expression in the hexadecimal number system**:
[0x800]

⊃ *label of the non-binary sc-connective*
⇒ *numerical expression in the hexadecimal number system**:
[0x80]

⊃ *label of the sc-structure*
⇒ *numerical expression in the hexadecimal number system**:
[0x100]

⊃ *label of a role relation*
⇒ *numerical expression in the hexadecimal number system**:
[0x200]

⊃ *label of a non-role relation*
⇒ *numerical expression in the hexadecimal number system**:
[0x400]

⊃ *label of the sc-class*
⇒ *numerical expression in the hexadecimal number system**:
[0x800]

⊃ *label of an abstract entity*
⇒ *numerical expression in the hexadecimal number system**:
[0x1000]

⊃ *label of a material entity*
⇒ *numerical expression in the hexadecimal number system**:
[0x2000]

⊃ *label of the constant positive permanent sc-arc of belonging*
:= [a label of the basic sc-arc]

:= [a label of the sc-arc of the main type]
⇐ *intersection**:
{• *label of the sc-arc of membership*
• *sc-constant label*
• *label of the positive sc-arc of membership*
• *permanent sc-arc label*
}
⊃ *label of the variable positive permanent sc-arc of membership*

Labels of syntactic types of sc-elements can be combined to obtain more specific label classes. In terms of software implementation, such a combination is expressed by the operation of bitwise addition of the values of the corresponding labels.

Numerical expressions of some label classes may be the same. This is done to reduce the size of the element of the sc-storage by reducing the maximal size of the label. There is no conflict in this case, since such label classes, for example, the *label of a role relation* and the *label of a fuzzy sc-arc of belonging*, cannot be combined.

It is important to note that each of the allocated label classes (except the classes obtained by combining other classes) uniquely corresponds to the ordinal number of a bit in linear memory, which can be seen by looking at the corresponding numerical expressions of the label classes. It means that the label classes are not included in each other, for example, specifying the *label of a positive sc-arc of belonging* does not automatically indicate the *label of the sc-arc of belonging*. This allows making the operations of combining and comparing labels more efficient.

Let us briefly consider the disadvantages of the current implementation of the *labels of syntactic types of sc-elements* and possible ways to eliminate them:

- At the moment, the number of *labels of the syntactic type of the sc-element* is quite large, which leads to arising a sufficiently large number of situations, in which it is necessary to take into account the explicit and implicit storage of sc-arcs of belonging to the corresponding classes. On the other hand, changing the set of labels for any purpose in the current implementation is a rather time-consuming problem (in terms of the extent of changes in the platform code and sc-agents implemented at the platform level), and an extension of the set of labels without increasing the size of the element of the sc-storage in bytes turns out to be quite impossible.
  The solution to this problem is to minimize the number of labels as much as possible, for example, to the number of labels that correspond to the *SC-code Alphabet*. In this case, the belonging of sc-elements to any other classes will be written explicitly, and the number of situations, in which it will be necessary to take into account the implicit storage of sc-arcs, will be minimal;
- Some labels from the current set of *labels of the syntactic type of the sc-element* are used quite rarely (for example, the *label of the sc-edge of a common type* or the *label of a negative sc-arc of belonging*), in turn, in sc-memory may exist classes that have quite a lot of elements (for example, a *binary relation* or a *number*). This fact does not allow using the efficiency of the occurrence of labels in full.
  The solution to this problem is to stop using the set of labels known in advance and switch to a dynamic set of labels (while their number can remain fixed). In this case, a set of classes expressed as labels will be formed on the basis of some criteria, for example, the number of elements of this class or the frequency of references to it.

label of the access level of the sc-element
⇐ *second domain**:
*label of the access level of the sc-element**
⇒ *generalized structure**:
{• *label of the reading access level of the sc-element*
• *label of the writing access level of the sc-element*
}

In the current *Implementation of the sc-storage*, *labels of the access level* are used to provide the ability to restrict access of some processes in sc-memory to some sc-elements stored in sc-memory.

Each element of the sc-storage corresponds to a *label of the reading access level of the sc-element* and a *label of the writing access level of the sc-element*, each of which is expressed as a number from 0 to 255.

In turn, each process (most often one that corresponds to some sc-agent) that tries to gain access to this element of the sc-storage (read or change it) corresponds to the reading and writing access level expressed in the same numerical range. The specified access level for the process is a part of the *context of a process*. Reading or writing access to the element of the sc-storage is not allowed if the process correspondingly has a lower reading or writing access level than the element of the sc-storage that is being accessed.

Thus, value zero of the *label of the reading access level of the sc-element* and the *label of the writing access level of the sc-element* means that any process can get unlimited access to this element of the sc-storage.

V. Implementation of means for processing constructs stored in semantic memory

Let us consider the currently implemented means of processing (access and editing) constructs stored in semantic memory.

The basic means of access to constructs stored in sc-memory are *sc-iterators*.

sc-iterator
:= [ScIterator]
⇐ *class of components**:
Implementation of the sc-storage
⊃ *three-element sc-iterator*
⇒ *class of sc-constructs**:
three-element sc-construct
⊃ *five-element sc-iterator*
⇒ *class of sc-constructs**:
five-element sc-construct

From a functional point of view, *sc-iterators*, as part of the *Implementation of the sc-storage*, are a basic tool for access to constructs stored in sc-memory, which allows reading (viewing) constructs that are isomorphic to the simplest templates – *three-element sc-constructs* and *five-element sc-constructs* [8].

From the point of view of implementation, the *sc-iterator* is a data structure that corresponds to a certain additionally precised class of sc-constructs and allows using the appropriate set of functions to consistently view all sc-constructs of this class represented in the current state of sc-memory (to iterate over sc-constructs).

Each class of *sc-iterators* corresponds to some known class (template, pattern) of sc-constructs. When developing an sc-iterator, this template is precised, that is, some (at least one) elements of the template are associated with a specific *sc-element* known in advance (the starting point for the search), and other elements of the template (those that need to be found) are associated with some type of the sc-element from the types that correspond to *labels of the syntactic type of the sc-element*.

Then, by calling the corresponding function (or a class method in OOP), all sc-constructs that correspond to the received template are sequentially viewed (taking into consideration the specified types of sc-elements and sc-elements known in advance), that is, the *sc-iterator* sequentially "switches" from one construct to another as long as such constructs exist. The existence of the following construct is checked immediately before switching. In the general case, there may not be constructs that correspond to the specified template, in this case, iteration will not occur (there will be 0 iterations).

At each iteration, sc-addresses of sc-elements included in the corresponding sc-construct are written to the sc-iterator, so the found elements can be processed appropriately, depending on the problem.

Currently, a *five-element sc-iterator* is implemented on the basis of *three-element sc-iterators* and in this sense is not atomic. However, the introduction of *five-element sc-iterators* is reasonable in terms of the convenience for the developer of programs for processing sc-constructs.

sc-template
:= [ScTemplate]
⇐ *class of components**:
Implementation of the sc-storage
:= [a data structure in linear memory that describes a generalized sc-structure, which, in turn, can either be explicitly represented in sc-memory or not represented in its current state but can be represented if necessary]

Sc-iterators allow searching only for sc-constructs of the simplest configuration. To implement the search for sc-constructs of a more complex configuration as well as the generation of complex sc-constructs, *sc-templates* are used, on the basis of which the search or generation of constructs are then carried out. The *sc-template* is a data structure that corresponds to some *generalized sc-structure*, i.e., a *sc-structure* that contains *sc-variables*. Using the appropriate set of functions, the following actions can be carried out:

- search in the current state of sc-memory for <u>all</u> sc-constructs that are isomorphic to the specified template. As search parameters, the values for any of the sc-variables in the template can be specified. After the search, a set of search results will be developed, each of which is a set of pairs in the form “an sc-variable from the template – the corresponding sc-constant”. This set can be empty (in the current state of sc-memory there are no constructs that are isomorphic to the given template) or contain one or more elements. Substitution of values of sc-variables can be carried out both by the sc-address and by the system sc-identifier;
- generation of an sc-construct that is isomorphic to a given template. The parameters and results of generation are developed in the same way as in the case of search, except that in the case of generation, the result is always single and a set of results is not developed.

Thus, each *sc-template* factually assumes a set of templates developed by specifying values for the sc-variables included in the initial template.

It is important to note that the *sc-template* is a data structure in linear memory that corresponds to some *generalized sc-structure* in sc-memory but is not this *generalized sc-structure* itself. It means that the sc-template can be automatically developed on the basis of the *generalized sc-structure* explicitly represented in sc-memory and also developed at the level of the program code by calling the corresponding functions (methods). In the second case, the *sc-template* will exist only in linear memory, while the corresponding *generalized sc-structure* will not be explicitly represented in sc-memory. In this case, the substitution of the values of sc-variables will be possible only by the system sc-identifier, since the corresponding template elements will not contain sc-addresses.

When searching for sc-constructs that are isomorphic to a given template, from the point of view of efficiency, it is extremely important to consider, from which sc-element the search should be initiated. As it is known, in general, the search problem in the graph is an NP-complete problem, but the search in the sc-graph allows taking into consideration the semantics of the processed information, which, in turn, allows reducing the search time significantly.

One of the possible options for optimizing the search algorithm implemented at the moment is the ranking of three-element sc-constructs that are part of the sc-template, according to the order of search for these sc-constructs upon criterion of reducing the number of possible search options that are generated by one or another three-element sc-construct that contains sc-variables. Thus, at first, when searching, those three-element sc-constructs that initially contain two sc-constants are selected, and then those that initially contain one sc-constant. After performing the search step, the priority of sc-constructs changes, taking into account the results obtained in the previous step.

Another optimization option is based on that specific feature of formalization in the SC-code, that, in general, the number of sc-arcs that come in a certain sc-element, as a rule, is significantly lower than the number of sc-arcs that go out it. Thus, it is reasonable to initiate the search through the incoming sc-arcs.

It can be assumed that the features provided by *sc-templates* allow eliminating the usage of *sc-iterators* completely. However, this is not quite true for the following reasons:

- search and generation by a template are implemented on the basis of sc-iterators as a basic means of searching for sc-constructs within the *Implementation of the sc-storage*;
- *sc-iterators* make it possible to organize the search process more flexibly, taking into account the semantics of specific sc-elements involved in the search. For example, we can consider the fact that for some sc-elements the number of incoming sc-arcs is significantly lower than the number of outgoing ones (or vice versa). Thus, when searching for constructs that contain such sc-elements, it is more efficient to initiate a search from those sections where there are potentially fewer arcs.

context of a process within the software model of sc-memory
:= [ScContext]
:= [a context of a process run at the level of the software model of sc-memory]
:= [a meta description of a process in sc-memory run at the level of the software model of sc-memory]
:= [a data structure that contains meta information about a process run in sc-memory at the platform level]
⇐ *class of components**:
Implementation of the sc-storage

Each process that is run in sc-memory at the level of the *platform for interpreting sc-models of computer systems* (that most often corresponds to some *sc-agent* implemented at the platform level) associates with the *context of a process*, which is a data structure that describes metainformation about this process. Currently, the context of a process contains information about the reading and writing access level for this process (See the *label of the access level of the sc-element*).

When calling any functions (methods) connected with access to constructs stored in sc-memory within the process, one of the parameters is necessarily the *context of a process*.

subscription to an event in sc-memory within the software model of sc-memory
:= [ScEvent]
:= [a data structure that describes within the sc-memory software model the correspondence between the class of events in sc-memory and actions that should be performed when events of this class occur in sc-memory]
⇐ *class of components**:
Implementation of the sc-storage

To make it possible to develop sc-agents within the *platform for interpreting sc-models of computer systems*, the possibility to subscribe to an event that belongs to one of the *classes of atomic events in sc-memory** [8] is implemented, specifying the sc-element which should be associated with the event of this class (for example, the sc-element, for which an incoming or outgoing sc-arc should be shown up). A subscription to an event is a data structure that describes a class of expected events and a function in the program code that should be called when this event occurs.

All subscriptions to events are registered within the event table. With any change in sc-memory, this table is viewed and the functions that correspond to the event that occurred are run.

In the current implementation, each event is processed in a separate operating system thread, while at the implementation level a parameter is set that describes the number of maximal threads that can be run in parallel.

Thus, it is possible to implement sc-agents that respond to events in sc-memory as well as to suspend its work when running a certain process in sc-memory and wait for some event to occur (for example, create a subproblem for some group of sc-agents and wait for its solution).

To store the contents of internal files of ostis-systems, the size of which exceeds 48 bytes, files that are explicitly stored on the file system are used, which is accessed by means of the operating system, on which the *Software version of the implementation of the platform for interpreting sc-models of computer systems* runs. The implementation of file memory is described in more detail in [5].

In addition, to implement a quick search for sc-elements by their string sc-identifiers or their fragments (substrings), an additional key-value storage is used, which corresponds to the *string sc-identifier* an *sc-address* of the *sc-element*, whose identifier is this string (in the case of the basic and system sc-identifier) or the *sc-element*, which is a sign of the *internal file of the ostis-system* (in the case of a nonbasic sc-identifier).

VI. Implementation of the basic set of platform-dependent sc-agents and their common components

Part of the *Software model of sc-memory* is the *Implementation of a basic set of platform-dependent sc-agents and their common components*, which allows the user to navigate through the knowledge base via user interface commands. This, in turn, allows beginning work with the ostis-system immediately after installing the platform and downloading the knowledge base without the need to connect any additional modules.

Implementation of a basic set of platform-dependent sc-agents and their common components
⇒ *component of the software system**:
- *Implementation of the basic set of search sc-agents*
- *Implementation of the basic mechanism for collecting junk information*
- *Implementation of the basic set of front end sc-agents*

The current implementation of the mechanism for collecting junk information contains an sc-agent that responds to the explicit addition of any sc-element to the set "junk information" and carries out the physical deleting of this sc-element from sc-memory

Implementation of the basic set of search sc-agents
⇒ *component of the software system**:
- *Implementation of an Abstract sc-agent for searching for the semantic neighborhood of a given entity*
- *Implementation of an Abstract sc-agent for searching for all entities that are particular towards a given one*
- *Implementation of an Abstract sc-agent for searching for all entities that are common towards a given one*
- *Implementation of an Abstract sc-agent for searching for all sc-identifiers that correspond to a given entity*
- *Implementation of an Abstract sc-agent for searching for basic sc-arcs that are incident to a given sc-element*

Implementation of the basic set of front end sc-agents
⇒ *component of the software system**:
- *Implementation of an Abstract sc-agent for processing user interface commands*
- *Implementation of an Abstract sc-agent for translating from an internal knowledge representation to an intermediate transport format*
  ⇒ *note**:
  [currently, an approach is used, in which, regardless of the form of external representation of information, the information stored in sc-memory is firstly translated into an intermediate transport format based on JSON, which is then processed by sc-agents of the user interface, which are part of *Implementation of the interpreter of sc-models of user interfaces*]

VII. Principles of interaction of the software model of sc-memory with external resources

The interaction of the software model of sc-memory with external resources can be carried out through a specialized program interface (API), but this option is inconvenient in most cases, because:
- it is supported only for a very limited set of programming languages (C, C++, Python);
- it requires that the client application that accesses the software model of sc-memory factually forms a whole unit with it, thus eliminating the possibility of building a distributed group of ostis-systems;
- as a consequence of the previous item, the possibility of parallel work with sc-memory of several client applications is excluded.

To make it possible to access sc-memory remotely, without taking into account the programming languages, with which a specific client application is implemented, it was decided to implement the possibility of accessing sc-memory using universal protocols that do not depend on the means of implementing a particular component or system. The binary protocol SCTP [22] and the text protocol based on JSON [23] were developed as such protocols. Further, the protocols themselves will be considered in more detail as well as the means that allow interaction on the basis of these protocols.

Implementation of the subsystem of interaction with the environment using network protocols

⇒ *component of the software system**:
- *Implementation of the subsystem of interaction with the environment using the SCTP protocol*
- *Implementation of the subsystem of interaction with the environment using protocols based on the JSON format*

SCTP is a *binary protocol* that allows performing such operations as reading (search) and editing constructs stored in sc-memory as well as track events that occur in sc-memory.

The interaction between the client and the server on the SCTP protocol is carried out by exchanging *sctp-commands*, each of which is a set of bytes intended for machine processing (but not for human perception).

SCTP

:= [The Semantic Code Transfer Protocol]

⇔ *analogy**:
HTTP

⇒ *generalized implementation**:
- *sctp-server*
- *sctp-client*

should be distinguished*

∋ {• *SCTP*
  := [The Semantic Code Transfer Protocol]
• *Stream Control Transmission Protocol*
  := [The Stream Control Transmission Protocol]
  ⇒ *note**:
    [A transport layer protocol in computer networks developed in 2000.]
}

The stp-server processes stp-commands that come from different sctp-clients and provides their interpretation in sc-memory.

In general, stp-clients can be implemented in different programming languages and have a different programming interface. In fact, the problem of the stp-client is to convert high-level commands presented in a convenient for the programmer form into one or more low-level sctp-commands, send them to the server, wait for the sctp-result and interpret it.

sctp-command

⇒ *generalized decomposition**:
{• *sctp-command header*
  := [a part of the sctp-command that specifies its type and some additional information about it]
• *arguments of the sctp-command*
  := [a part of the sctp-command that contains its arguments and the dimension of which may vary, depending on the type of the command.]
}

⇒ *inclusion**: *example**:
- *sctp-command for deleting an sc-element with the specified sc-address*
- *sctp-command for creating a new sc-node of the specified type*
- *sctp-command for getting the initial and final elements of the sc-arc*

⇒ *note**:
[Running of each sctp-command assumes the occurrence of an sctp-result that uniquely corresponds to this command.]

The SCTP protocol has a number of advantages:
- The SCTP protocol is crossplatform;
- The SCTP protocol can be implemented quite simply in almost any programming language.

However, the SCTP protocol can be considered outdated at the moment, since it has a number of significant disadvantages:
- SCTP protocol commands are low-level (focused on work with single sc-elements or the simplest sc-constructs of 3 or 5 elements). This leads to the fact that performing even a simple transformation in the knowledge base or a content-addressable retrieval through a set of interrelated constructs is expressed in the form of a fairly large set of sctp-commands. Because for each command there is an sctp-result that is also sent over the network, this unnecessarily loads the network and greatly decreases the efficiency of the system as a whole. In addition, the efficiency of the system begins to depend heavily on the network bandwidth;
- The SCTP protocol is not designed for human perception.

Implementation of the subsystem of interaction with the environment using the SCTP protocol

⇒ *component of the software system**:
- *Implementation of the sctp-server*
- *Implementation of the sctp-client*

The *Implementation of the subsystem of interaction with the environment using the SCTP protocol* includes the *Implementation of the sctp-client* in C++, at the same time, there are other implementations of *sctp-clients* within the same software implementation of the platform, for example, within the *Implementation of the interpreter of sc-models of user interfaces*.

Due to the large number of disadvantages of the SCTP protocol, it was decided to develop another protocol based on some universally accepted text transport format. The JSON format was chosen as such one. Currently, the existing components of the platform and specific ostis-systems are being transferred from using the SCTP protocol to the text protocol based on the JSON format. This protocol does not have its name yet.

Within the *Protocol for interaction with sc-memory based on JSON*, each command is a json-object, in which the command identifier, the type of the command and its arguments are specified. In turn, the response to the command is also a json-object, in which the command ID, its status (run successfully/unsuccessfully) and the results are specified. The structure of arguments and results of a command is determined by the type of command.

The protocol based on the JSON format has a number of advantages:

- JSON is a universally accepted open format, for working with which there are a large number of libraries for popular programming languages. This, in turn, simplifies the implementation of the client and server for the protocol built on the basis of JSON;
- The implementation of the JSON-based protocol does not apply fundamental restrictions on the dimension (length) of each command, unlike the binary protocol does. Thus, it becomes possible to use non-atomic commands that allow, for example, creating several sc-elements in one act of transferring such a command over the network. Important examples of such commands are the *Random template generation command* and the *Random template search command*;
- It can be said that the JSON-based protocol is the next step towards creating a powerful and universal request language that is similar to the SQL language for relational databases and that is designed to work with sc-memory. The next step will be the implementation of such a protocol based on one of the standards for the external display of sc-constructs, for example, *SCs-code*, which, in turn, will allow transferring entire programs for processing sc-constructs as commands, for example, in the SCP language.

VIII. Implementation of accessory tools for working with sc-memory

The *Implementation of accessory tools for working with sc-memory* is currently represented only by the *Implementation of the collector of the knowledge base from source texts written in the SCs-code*.

The collector of the knowledge base from source texts allows building a knowledge base from a set of source texts written in a limited SCs-code into a binary format perceived by the *Software model of sc-memory*. In this case, the building is possible both "from scratch" (with the destruction of a previously created memory snapshot) and an additive building, when the information contained in a given set of files is added to an already existing memory state snapshot.

In the current implementation, the collector performs "pasting together" ("merging") of sc-elements that have the same *system sc-identifiers* at the source level.

IX. Implementation of the interpreter of sc-models of user interfaces

Along with the implementation of the *Software model of sc-memory*, an important part of the *Software version of the implementation of the platform for interpreting sc-models of computer systems* is the *Implementation of the interpreter of sc-models of user interfaces*, which provides basic means for viewing and editing the knowledge base by the user, means for navigating through the knowledge base (asking questions to the knowledge base) and can be supplemented with new components, depending on the problems solved by each specific ostis-system.

Implementation of the interpreter of sc-models of user interfaces
:= [sc-web]
⇒ *programming language used**:
- *JavaScript*
- *TypeScript*
- *Python*

Figure 5 shows the Architecture of the *Implementation of the interpreter of sc-models of user interfaces*. This artwork shows the planned version of the architecture of the *Implementation of the interpreter of sc-models of user interfaces*, an important principle of which is the simplicity and uniformity of connecting with any components of the user interface (editors, visualizers, switches, menu commands, etc.). For this purpose, the Sandbox software middleware is implemented, within which low-level operations of interaction with the server part are implemented and which provides a more convenient programming interface for developers of components.

The current *Implementation of the interpreter of sc-models of user interfaces* has a number of disadvantages:

- The lack of a single unified mechanism for client-server interaction. Some components (a visualizer of sc-texts in the SCn-code, menu commands, etc.) run over the HTTP protocol, some – over the SCTP protocol using the WebSocket technology, which leads to significant difficulties in the development of the platform;
- The HTTP protocol assumes a clear separation of the active client and the passive server that responds to client requests. Thus, in practice, the server (in this case, sc-memory) cannot send a message to the client

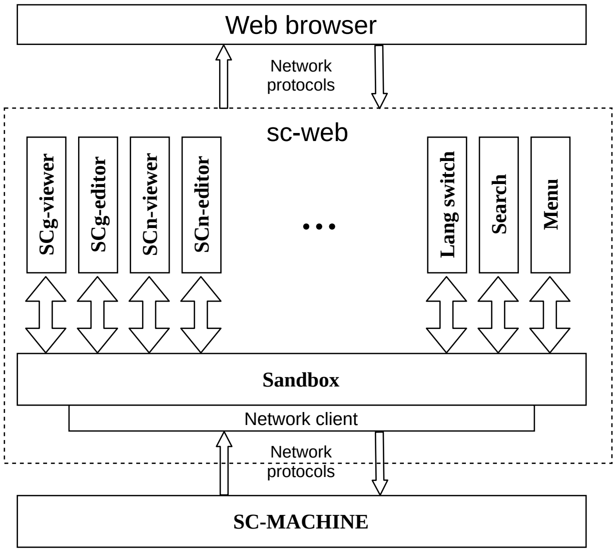


Figure 5. Architecture of the Implementation of the interpreter of sc-models of user interfaces

in for convenience, which increases the security of the system but significantly reduces its interactivity. In addition, this variant of the implementation makes it difficult to implement the multiagent approach adopted in the OSTIS Technology, in particular, it makes it difficult to implement sc-agents on the front end. These problems can be solved by constant monitoring of certain events on the front end, however, this option is ineffective;
In addition, a part of the interface factually works directly with sc-memory using the WebSocket technology, and the other one – through a middleware based on the tornado library for the Python programming language, which leads to additional dependencies on third-party libraries;

- Some of the components (for example, the search box by ID) are implemented by third-party means and are not connected with sc-memory in practice. It makes it difficult to develop the platform;
- The current *Implementation of the interpreter of sc-models of user interfaces* is focused only on conducting a dialogue with the user (in the "user question – system answer" style). Obviously necessary situations are not supported, such as running a command that does not assume an answer; an error or the absence of an answer; the need of a system to ask a question to the user, etc;
- The ability for the user to interact with the system without using special control components is limited. For example, you can ask a question to the system by drawing it in the SCg-code, but the user will not see the answer, although it will be created in memory by the corresponding agent;
- Most of the technologies used in the implementation of the platform are now outdated, which makes it difficult to develop the platform;
- The idea of platform independence of the user interface (building an sc-model of the user interface) is not implemented to the full. Currently, it is likely to be difficult to describe the sc-model of the user interface (including the exact placement, dimensions, design of components, their behavior, etc.) completely due to performance limitations, however, it is quite possible to implement the ability to ask questions to all interface components, change their placement, etc., however, these capabilities cannot be implemented in the current version of the platform implementation;
- The interface part works slowly due to the disadvantages of the SCTP protocol and some disadvantages of the implementation of the server part in Python;
- The inheritance mechanism is not implemented when adding new external languages. For example, adding a new language, even very close to the SCg-code, requires physically copying the component code and making appropriate changes, thus two components that are not connected in any way are obtained, which begin to develop independently;
- A low level of documentation of the current *Implementation of the interpreter of sc-models of user interfaces*.

Based on this, the requirements for the future (new) version of the *Implementation of the interpreter of sc-models of user interfaces*, which is currently being developed, were formulated:

- Unify the principles of interaction of all interface components with the *Software model of sc-memory*, regardless of what type the component belongs to. For example, the list of menu commands should be formed through the same mechanism as the response to the user request, and the editing command generated by the user, and the command for adding a new fragment to the knowledge base, etc;
- Unify the principles of user interaction with the system, regardless of the method of interaction and the external language. For example, it should be possible to ask questions and run other commands directly through the SCg/SCn interface. At the same time, it is necessary to take into account the principles of editing the knowledge base, so that the user cannot add new information into the compliant part of the knowledge base under the guise of asking a question;
- Unify the principles of processing events that occur when the user interacts with interface components – the behavior of buttons and other interactive components should not be set statically by third-party means but implemented as an agent, which, nevertheless, can be implemented randomly (not

necessarily at a platform-independent level). Any action performed by the user at the logical level should be interpreted and processed as the initiation of the agent;
- Provide the ability to run commands (in particular, ask questions) with a random number of arguments, including those without arguments;
- Provide the ability to display the answer to the question partially if the answer is very large and it takes a long time for it to be displayed;
- Each displayed interface component should be interpreted as an image of some sc-node described in the knowledge base. Thus, the user should be able to ask random questions to any components of the interface;
- Simplify as much as possible and document the mechanism for adding new components;
- Provide the ability to add new components based on existing ones without creating independent copies. For example, it should be possible to create a component for a language that extends the SCg language with new primitives, redefine the principles of placement of sc-texts, etc;
- Minimize dependence on third-party libraries;
- Minimize the usage of the HTTP protocol (a bootstrap loading of the general interface structure), ensure the possibility of equal two-way interaction between the server and client parts;
- Stop using the SCTP protocol completely, switch to the JSON-based one, document it.

It is obvious that the implementation of most of the above requirements is not only connected with the very version of the implementation of the platform but also requires the development of the theory of logical-semantic models of user interfaces and the clarification of the general principles of the organization of user interfaces of ostis-systems within it. However, the possibility of implementing such models in principle should be taken into account within the platform implementation.

Next, let us consider the current set of components that are included by default in the *Implementation of the interpreter of sc-models of user interfaces*. As mentioned earlier, this set can be extended by other components, depending on the problems solved by a specific ostis-system.

Implementation of the interpreter of sc-models of user interfaces
⇒ *component of the software system**:
- *User interface command menu panel*
- *Component for switching the language of identification of displayed sc-elements*
- *Component for switching the external language of knowledge visualization*
- *Search box of sc-elements by ID*
- *Panel for displaying the user dialog with the ostis-system*
- *Panel for visualization and editing knowledge*
  ⇒ *component of the software system**:
  - *Visualizer of sc.n-texts*
  - *Visualizer and editor of sc.g-texts*

User interface command menu panel contains displays of command classes (both atomic and non-atomic) that are currently available in the knowledge base and are included in the decomposition of the *Main menu of the user interface* (meaning a complete decomposition, which in general can include several levels of non-atomic command classes).

Interaction with the display of a non-atomic command class initiates a command for displaying classes of commands included in the decomposition of this non-atomic command class.

Interaction with the display of an atomic command class initiates the generation of a command of this class with previously selected arguments based on the corresponding *generalized definition of the command class* (command class template).

The *Component for switching the language of identification of displayed sc-elements* is a display of the set of natural languages available in the system. The interaction of the user with this component switches the user interface to the mode of communication with a particular user using the *base sc-identifiers* that belong to this *natural language*. It means that when displaying sc-identifiers of sc-elements in any language, for example, in the SCg- or SCn-code, the *base sc-identifiers* that belong to this *natural language* will be used. It is subject both to sc-elements displayed within the *Panel for visualization and editing knowledge* and to any other sc-elements, for example, command classes and even the *natural languages* themselves displayed within the *Component for switching the language of identification of displayed sc-elements*.

The *Component for switching the external language of knowledge visualization* is used to switch the language of knowledge visualization in the active window displayed on the *Panel for visualization and editing knowledge*. In the current implementation, SCg- and SCn-codes are supported as such languages by default as well as any other languages included in the set of *external languages of the SC-code visualization*.

The *Search box of sc-elements by ID* allows searching for sc-identifiers that contain a substring input in this box (capitalization is respected). As a result of the search, a list of sc-identifiers that contain the specified substring is displayed, when interacting with which the question "What is it?" is automatically put, the argument of which is either the sc-element itself, which has this sc-identifier (in case the specified sc-identifier is the base or system one, and thus the specified sc-element can be uniquely defined) or the internal ostis-system file itself, which is the sc-identifier (in case this sc-identifier is not the base one).

The *Panel for displaying the user dialog with the ostis-system* displays a time-ordered list of sc-elements, which are signs of actions that are initiated by the user within the dialog with the ostis-system by interacting with displays of the corresponding command classes (that is, if the action was initiated in another way, for example, by an explicit initiation through the creation of an arc of belonging to a set of *initiated actions* in the sc.g-editor, then it will not be displayed on this panel). When the user interacts with each of the displayed signs of actions on the *Panel for visualization and editing knowledge*, a window that contains the result of performing this *action* in the language for visualization is displayed, in which it was displayed when the user viewed it the last (previous) time. Thus, in the current implementation, this panel can work only if the action initiated by the user assumes the result of this action explicitly represented in the memory. In turn, it follows that at present this panel, as the *Implementation of the interpreter of sc-models of user interfaces* in common, allows working with the system only in the "question-answer" dialog mode.

The *Panel for visualization and editing knowledge* displays windows that contain an sc-text represented in some language from a set of *external languages of the SC-code visualization* and that, as a rule, is the result of some action initiated by the user. If the corresponding visualizer supports the possibility of editing texts of the corresponding natural language, then it is at the same time also an editor.

If necessary, the user interface of each specific ostis-system can be supplemented with visualizers and editors of various external languages, which in the current version of the *Implementation of the interpreter of sc-models of user interfaces* will also be placed on the *Panel for visualization and editing knowledge*.

X. Implementation of the interpreter of programs of the base programming language of the OSTIS Technology

As a base programming language within the OSTIS Technology, including for the implementation of programs of sc-agents, an *SCP Language* [8] is proposed. The most important feature of the SCP Language is the fact that its programs are written in the same way as the knowledge they process, that is, in the SC-code. This, on the one hand, allows making ostis-systems platform-independent (clearly separate the *sc-model of a computer system* and the platform for interpreting such models) and, on the other hand, requires the occurrence of the *Implementation of the scp-interpreter* within the platform, that is, an interpreter of SCP-programs.

The structure of the *Implementation of the scp-interpreter* corresponds to the structure of an *Abstract scp-machine* (an abstract model of the scp-interpreter) considered in a number of papers, for example, in [18]. Next, let us consider this structure in the SCn-code.

Implementation of the scp-interpreter
⇐ *software implementation**:
Abstract scp-machine
⇒ *component of the software system**:
- *Implementation of an Abstract sc-agent for creating scp-processes*
- *Implementation of an Abstract sc-agent for interpreting scp-operators*
- *Implementation of an Abstract sc-agent for synchronizing the process of interpreting scp-programs*
- *Implementation of an Abstract sc-agent for killing scp-processes*
- *Implementation of an Abstract sc-agent for synchronizing events in sc-memory and its implementation*

Implementation of an Abstract sc-agent for interpreting scp-operators
⇒ *component of the software system**:
- *Implementation of an Abstract sc-agent for interpreting scp-operators for generating constructs*
- *Implementation of an Abstract sc-agent for interpreting scp-operators of the content-addressable retrieval of constructs*
- *Implementation of an Abstract sc-agent for interpreting scp-operators for deleting constructs*
- *Implementation of an Abstract sc-agent for interpreting scp-operators for checking conditions*
- *Implementation of an Abstract sc-agent for interpreting scp-operators for managing operand values*
- *Implementation of an Abstract sc-agent for interpreting scp-operators for managing scp-processes*
- *Implementation of an Abstract sc-agent for interpreting scp-operators for managing events*
- *Implementation of an Abstract sc-agent for interpreting scp-operators for processing the contents of numerical files*
- *Implementation of an Abstract sc-agent for interpreting scp-operators for processing the contents of string files*

The current *Implementation of the scp-interpreter* does not include specialized tools for working with locks, since the mechanism for locking sc-memory elements is implemented at a lower level within the *Implementation* *of the sc-storage and the mechanism for accessing it.*

XI. Conclusion

The paper considers the current version of the implementation of the software implementation of the platform for interpreting sc-models of computer systems built using the OSTIS Technology. The components of this implementation, their advantages and disadvantages are considered in detail, the problems for the development of components and the platform as a whole are formulated.

It is important to note that an ontological approach was used to describe the implementation of the platform for interpreting sc-models of computer systems, which makes it possible to make such a description understandable not only to a human but also to an intelligent computer system and, ultimately, will allow changing-over to the ontological design of the platform implementation, first in software and later in hardware variants.

Acknowledgment

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help in the work and valuable comments.

The work was carried out with the partial financial support of the BRFFR (agreement No. F21RM-139).

References

[1] L. G. Komarcova and A. V. Maksimov, *Nejrokomp'yutery: Ucheb. posobie dlya vuzov. - 2-e izd., pererab. i dop. [Neurocomputers: A Textbook for Universities. - 2nd ed., revised and enlarged]*, ser. Informatika v tekhnicheskom universitete [Informatics at the Technical University]. Moscow: MGTU im. N.E. Baumana [Bauman Moscow State Technical University], 2004, (In Russ).

[2] (2021, Jun) PK PROLOG [PC PROLOG]. [Online]. Available: http://www.prolog-plc.ru/

[3] (2021, Jun) USB Accelerator | Coral. [Online]. Available: https://coral.ai/products/accelerator/

[4] V. Golenkov, N. Gulyakina, I. Davydenko, and D. Shunkevich, "Semanticheskie tekhnologii proektirovaniya intellektual'nyh sistem i semanticheskie associativnye komp'yutery [Semantic technologies of intelligent systems design and semantic associative computers]," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 42–50, 2019.

[5] D. N. Koronchik, "Unificirovannye semanticheskie modeli pol'zovatel'skih interfejsov intellektual'nyh sistem i tekhnologiya ih komponentnogo proektirovaniya [Unified semantic models of user interface for intelligent systems and technology for their develop]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2013, pp. 403–406.

[6] D. Koronchik, "Realizaciya platformy dlya web-orientirovannyh sistem, upravlyaemyh znaniyami [Implementation of web-platform for systems based on knowledges]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2015, pp. 89–92.

[7] H.-m. Haav, "A comparative study of approaches of ontology driven software development," *Informatica*, vol. 29, pp. 439–466, 11 2018.

[8] (2021, Jun) IMS.ostis Metasystem. [Online]. Available: https://ims.ostis.net

[9] V. Golenkov, N. Guliakina, I. Davydenko, and A. Eremeev, "Methods and tools for ensuring compatibility of computer systems," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2019, pp. 25–52.

[10] O. P. Kuznecov, *Diskretnaya matematika dlya inzhenera: Uchebnik dlya vuzov [Discrete Mathematics for an Engineer: A Textbook for High Schools]*. Moscow: Lan', 2009.

[11] (2021, Jun) Graph Database Platform | Graph Database Management System | Neo4j. [Online]. Available: https://neo4j.com/

[12] (2021, Jun) ArangoDB, the multi-model database for graph and beyond. [Online]. Available: https://www.arangodb.com/

[13] (2021, Jun) Home | OrientDB Community Edition. [Online]. Available: https://orientdb.org/

[14] (2021, Jun) Vaticle | Home. [Online]. Available: https://vaticle.com/

[15] (2021, Jun) OpenLink Software: Virtuoso Homepage. [Online]. Available: https://virtuoso.openlinksw.com/

[16] (2021, Jun) Welcome · Eclipse RDF4J™ | The Eclipse Foundation. [Online]. Available: https://rdf4j.org/

[17] V. P. Ivashenko, N. L. Verenik, A. I. Girel', E. N. Sejtkulov, and M. M. Tatur, "Predstavlenie semanticheskih setej i algoritmy ih organizacii i semanticheskoj obrabotki na vychislitel'nyh sistemah s massovym parallelizmom [Semantic networks representation and algorithms for their organization and semantic processing on massively parallel computers]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2015, pp. 133–140.

[18] D. Shunkevich, "Agentno-orientirovannye reshateli zadach intellektual'nyh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 119–132.

[19] (2021, Jun) OSTIS. [Online]. Available: https://github.com/ostis-dev

[20] (2021, Jun) ostis-apps/sc-web: The sc-web enchancement of https://github.com/deniskoronchik/sc-web/tree/master. [Online]. Available: https://github.com/ostis-apps/sc-web

[21] (2021, Jun) Sc-machine. [Online]. Available: http://ostis-dev.github.io/sc-machine/

[22] (2021, Jun) Sctp-protocol - sc-machine. [Online]. Available: http://ostis-dev.github.io/sc-machine/net/sctp/

[23] (2021, Jun) Websocket - sc-machine. [Online]. Available: http://ostis-dev.github.io/sc-machine/http/websocket/

Онтологический подход к разработке программной модели семантического компьютера на основе традиционной компьютерной архитектуры

Шункевич Д.В., Корончик Д.Н.

В работе рассмотрен онтологический подход к разработке программной модели платформы интерпретации семантических моделей интеллектуальных компьютерных систем (программной модели семантического компьютера). Подробно рассмотрена архитектура указанной программной модели, детально описаны ее компоненты, принципы их реализации, указаны преимущества принятых решений перед аналогами.

Отличительной особенностью работы является демонстрация применения онтологического подхода к разработке программных продуктов на примере указанной программной модели семантического компьютера.

Received 01.06.2021

Model for the representation of artificial neural networks and actions for their processing in the knowledge base

Mikhail Kovalev
Department of Intelligent Information Technologies
Belarusian State University of Informatics and Raddioelectronics
Minsk, Belarus
michail.kovalev7@gmail.com

***Abstract*—The article is dedicated to the problem of the integration of artificial neural networks with knowledge bases. The need for integration and development of a model for the representation of artificial neural networks (ANN) and actions for their processing in the knowledge base as the basis for such integration is justified. The subject domains and the corresponding ontologies of ANN and actions for processing ANN are described.**
***Keywords*—neuro-symbolic AI, ANN, knowledge base**

I. Introduction

Modern problem solvers of intelligent systems are increasingly faced with the need to solve complex problems using various traditional and intelligent methods of solving problems in a single information resource (in the limit – in a single knowledge base) [1].

One of the most popular intelligent methods of solving-problems is the method of solving problems usingartificial neural networks, which is caused, first of all,by the development of both the theory of ANN and thehardware capabilities of machines that are used for their training.

Neural network models show excellent results when working with unstructured data, but the traditional disadvantage of such models is the lack of human-understandable feedback, which could be called a reasoning chain. In other words, most neural network models work as a “black box” [2].

However, the complexity of modern intelligent systems that use neural network models, as well as the large amount of data processed by them, create the need to monitor, explain and understand the mechanisms of their work to verbalize the assessment and optimization of the activity of ANN.

Thus, we have, on the one hand, the need to use ANN in solving complex problems and, on the other hand, the need to explain how these problems were solved. In that context, it becomes relevant to develop neuro-symbolic approaches [3], [4], in particular, approaches for the integration of ANN and knowledge bases that use ontologies. Such integrated systems can combine:

- the possibility of semantic interpretation of the processed data, using the representation of the applied problems being solved by ANN as well as the specification of its input and output data;
- with a representation of the ANN structure itself, a description of its features and states, which make it easier to understand its operation [5].

There are two main directions of integration of ANN with knowledge bases:

- the building of intelligent systems that can use neural network methods jointly with other methods available in the system to solve complex problems. Such systems will be able to take into account the semantics of the problems at a higher level, which will make the solutions of these problems more structured and transparent;
- the building of an intelligent environment for the development, training and integration of various ANN compatible with knowledge bases through the representation of ANN with the help of ontological structures and their interpretation using knowledge representation. Such an environment will provide an opportunity for introspection of ANN, the ability to save the states of ANN after training and reconfiguration of the network, which will allow for a deeper analysis of its operation. Also, a formal description of the knowledge within the framework of the subject domain of ANN will help to lower the threshold for developers to enter the area of methods for solving problems with the help of ANN.

The development of both directions implies that the knowledge base stores knowledge (of various degrees of detail) about what neural network models it contains, how they are internally organized, what problems they can solve, etc. Therefore, the basis for the development of both directions is the development of a model for the representation of artificial neural networks and actions for their processing in the knowledge base. The purpose of this article is to develop such a model.

II. Proposed approach

The basis of the proposed approach is the usage of the OSTIS Technology and its basic principles for building a knowledge base [6].

Within the framework of the OSTIS Technology, powerful tools have been developed. Its allow describing any type of knowledge in a unified form, structuring the knowledge base according to various criteria as well as verifying its quality and editing the knowledge base directly while in operation [7].

The basis of the knowledge base built using the OSTIS Technology is a hierarchical system of subject domains and their corresponding ontologies. An ontology is interpreted as a specification of the system of concepts of the corresponding subject domain.

Further, the development of any subject domain and ontology will mean the development of a subject domain and ontology in the terminology of the OSTIS Technology.

Knowledge in knowledge bases built using the OSTIS Technology is represented using an SC-code. To record formalized texts, in the article, the variants of the external display of SC-code constructions are used – SCg (graphic version) and SCn (hypertext version). It is possible to examine in detail the representation of knowledge with the help of the SC-code here [8].

Thus, the specified problem of developing a model for the representation of artificial neural networks in the knowledge base and actions for their processing within the framework of the OSTIS Technology is reduced to the development of:

- the subject domain and the corresponding ontology of artificial neural networks;
- the subject domain and the corresponding ontology of actions for processing artificial neural networks.

III. Subject domain and the corresponding ontology of artificial neural networks

The subject domain of artificial neural networks contains a formal description of knowledge about particular artificial neural networks. The corresponding ontology contains a formal description of the concepts that are necessary for the representation of such knowledge.

The maximum class of objects of research in the subject domain of artificial neural networks is the ***artificial neural network***. This concept denotes a class, to which all entities that denote specific ANN belong.

An ***artificial neural network*** is a set of neural elements and connections between them [9]. The ANN consists of ***formal neurons***, which are interconnected by ***synaptic connections***. Neurons are organized in ***layers***. Each neuron of the layer receives signals from the synaptic connections that come in it, processes them in a single way using the ***activation function*** prescribed for it or for the entire layer and transmits the result to the synaptic connections that go out of it.

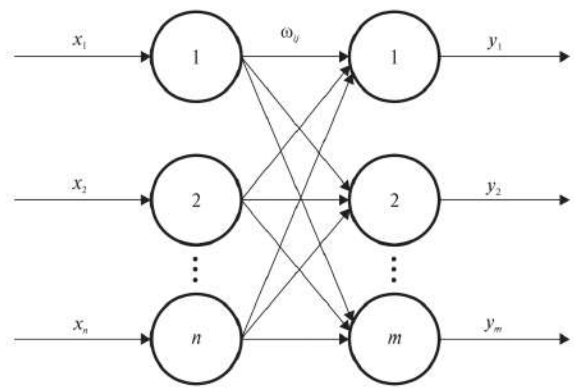


Figure 1. The general scheme of the artificial neural network [9]

Figure 1 shows the general scheme of the ANN.

We will call a set of information, which denotes data about the structure of ANN layers, formal neurons, synaptic connections and activation functions, i.e., something that can be trained and used to solve problems, the ANN architecture.

In accordance with the ANN architecture, it is possible to distinguish the following hierarchy of classes of ANN. Let us consider this hierarchy in the SCn-code.

artificial neural network
⇒ *subdividing**:
  Typology of ANN on the basis of the directivity of connections^
  = { • *ANN with direct connections*
      ⇒ *decomposition**:
        { • *perceptron*
            ⇒ *decomposition**:
              { • *Rozenblatt perceptron*
                • *autoencoder ANN*
              }
          • *support vector machine*
          • *ANN of radial-basis functions*
        }
    • *ANN with inverse connections*
      ⇒ *decomposition**:
        { • *Hopfield ANN*
          • *Hamming ANN*
        }
    • *recurrent artificial neural network*
      ⇒ *decomposition**:
        { • *Jordan ANN*
          • *Elman ANN*
          • *multi-recurrent ANN*
          • *LSTM-element*
          • *GRU-element*
        }
    }

⇒ *subdividing**:

*Typology of ANN on the basis of completeness of connections^*

= {• *fully connected ANN*

• *weakly connected ANN*

}

Next, the key elements of the ontology of ANN, the concepts used to formalize the ANN architecture will be described.

A ***formal neuron*** is the main element of the *artificial neural network*, which applies its *activation function* to the sum of the products of input signals by weight coefficients (Golovko2017):

$$y = F\left(\sum_{i=1}^{n} w_i x_i - T\right) = F(WX - -T)$$

where $X = (x_1, x_2, ..., x_n)^T$ is an input signal vector; $W -$ $-(w_1, w_2, ..., w_n)$ – a vector of weight coefficients; T – a threshold value; F – an activation function. The role relation *formal neuron'* is used to denote the belonging of a particular formal neuron to any ANN.

A separate formal neuron is an artificial neural network with one neuron in a single layer.

formal neuron

:= [an artificial neuron]

:= [a neuron]

⊂ *artificial neural network*

⇒ *subdividing**:

{• *fully connected formal neuron*

:= [a neuron that has a complete set of connections with the neurons of the previous layer]

• *convolutional formal neuron*

⇒ *explanation**:

[A separate processing element of the ANN, which performs a functional transformation of the result of the operation of convolution of the matrix of input values using the activation function.]

⇒ *explanation**:

[A convolutional formal neuron can be represented by a fully connected formal neuron.]

• *recurrent formal neuron*

⇒ *explanation**:

[A formal neuron that has an inverse connection with itself or with other neurons of the ANN.]

}

Figure 2 shows the general scheme of a formal neuron.

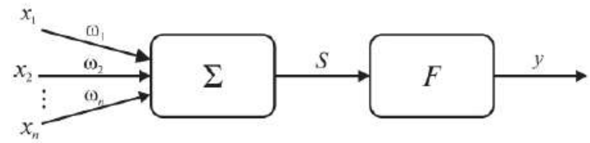


Figure 2. The general scheme of a formal neuron of the ANN [9]

The ***synaptic connection*** is a set of oriented pairs, the first components of which are the neurons, out of which the signal goes, and the second components are the neurons that receive this signal. When denoting the belonging of a particular synaptic connection to any ANN, the role relation *synaptic connection'* is used.

The ***weight sum**** is a non-role relation that connects a formal neuron with a number that is the sum of the products of input signals by the weight coefficients of the synaptic connections that come in the neuron.

The ***activation function**** is a non-role relation that connects a formal neuron with a function, the result of applying which to the weight sum of the neuron determines its output value. To describe such functions, in the knowledge base, classes of such functions (threshold, sigmoid, ReLU, etc.) and information about the cases, in which they are usually used, are described.

The ***ANN layer*** is a set of neural elements that receive information from other neural elements of the network in parallel at each clock period [9]. The activation function of a layer is the activation function of all the formal neurons of this layer. The description of the sequence of ANN layers with the configuration of each layer sets the ANN architecture. The configuration of the layer is set by the type, the number of formal neurons, the activation function. The description of the sequence of ANN layers with the configuration of each layer sets the ANN architecture. A separate layer is an artificial neural network with a single layer.

ANN layer

:= [layer]

:= [set of layers of artificial neural networks]

⊂ *artificial neural network*

⇒ *decomposition**:

{• *fully connected ANN layer*

:= [a layer, in which each neuron has a connection with each neuron of the previous layer]

• *convolutional ANN layer*

:= [a layer, in which each neuron is convolutional]

• *nonlinear transformation ANN layer*

:= [a layer that performs nonlinear transformation of input data]

⇒ *explanation**:

[As a rule, they are allocated into separate layers only in software implementations. In fact, they are considered as the final stage of calculating the output activity of any neuron – the application of the activation function.]

- *dropout ANN layer*
  - := [a layer that implements the dropout regularization technique]
  - ⇒ *explanation**:

    [This type of layer functions only during training the ANN.]
- *pooling ANN layer*
  - := [a subsampling layer]
  - := [a pooling layer]
- := [a layer that reduces the dimensionality of the input data]
- *ANN layer of the butch-normalization*

}

Also, within the framework of the subject domain, the hierarchy of ANN parameters is formalized.

ANN parameter

⊂ *parameter*

⇒ *decomposition**:

{• *configurable ANN parameter*

:= [an ANN parameter, the value of which changes during training]

⇒ *decomposition**:

{• *weight coefficient of the synaptic connection*

• *threshold value*

• *convolution kernel*

}

• *architectural ANN parameter*

⇒ *note**:

[The ANN parameter that defines its architecture.]

⇒ *decomposition**:

{• *number of layers*

• *number of neurons*

• *number of synaptic connections*

}

}

The following concepts are also distinguished within the subject domain: *threshold formal neuron', input value of formal neuron*, output value of formal neuron*, distributing layer*, processing layer*, output layer**.

With the help of the allocated concepts, it becomes possible to formalize the architecture of a particular ANN in the knowledge base. Figure 3 shows an example of the formalization of a fully connected two-layer ANN with two neurons on the input layer and one neuron on the processing layer.

It should be noted that within the subject domain, it is possible to formalize knowledge about the ANN without formalizing knowledge about the ANN architecture. This is possible when the ANN is a sign of an external entity

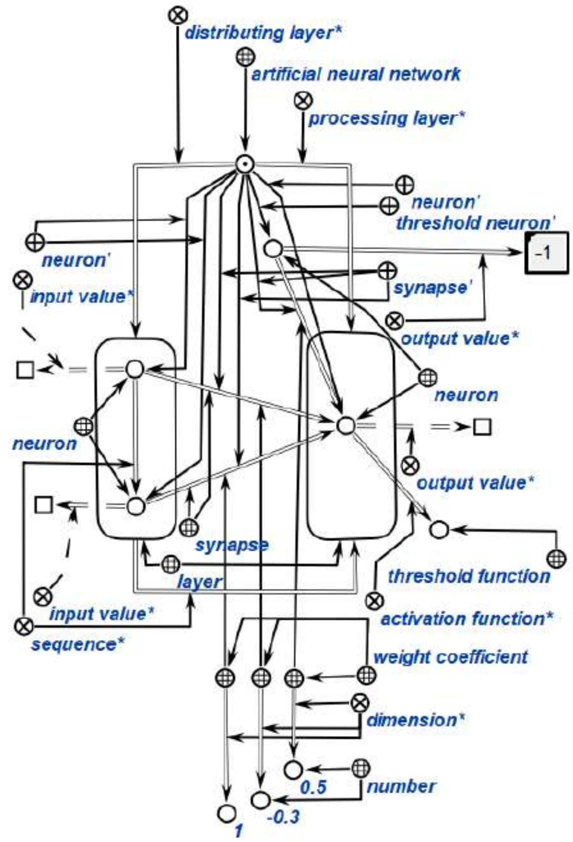


Figure 3. An example of the formalization of the architecture of an artificial neural network in the knowledge base

in relation to the knowledge base, for example, the ANN is implemented and works on a third-party service. In this case, it is sufficient to formalize the specification of such an ANN that includes:

- the ANN class;
- the ANN parameters;
- the type of problem being solved;
- the types of input and output data;
- the identifier, by which the specification of the ANN in the knowledge base can be correlated with the model in a third-party service;
- other knowledge that will help the system decide whether to use ANN as a method for solving a specific problem.

The description of each knowledge about ANN will allow the system that uses the subject domain under consideration to decide to use one or another ANN to solve a specific problem or to help the developer make such a decision when choosing an ANN to train for solving such a problem. For the same purposes, for each particular ANN, it is necessary to specify the problem that it can solve. The following classes of problems are identified, which can be solved with the help of the ANN with acceptable accuracy:

- the *classification problem*. The problem of creating a classifier, i.e., displaying $\tilde{c}: X \to C$, where $X \in \mathbb{R}^m$ is the attribute space of the input pattern, $C = C_1, C_2, \ldots C_k$ – a finite and usually small set of class labels.
- the *regression problem*. The problem of constructing an evaluation function based on examples $(x_i, f(x_i))$, where $f(x)$ is an unknown function. The *evaluation function* is the display of the form $\tilde{f}: X \to \mathbb{R}$, where $X \in \mathbb{R}^m$ is the attribute space of the input pattern.
- the *clustering problem*. The problem of dividing the set of input patterns into groups (clusters) according to some similarity metric.
- the *problem of reducing the dimensionality of the feature space*;
- the *control problem*;
- the *filtering problem*;
- the *detection problem*. It is a special case of a classification problem and a regression problem.
- the *problem with associative memory*.

IV. Subject domain and ontology of actions for processing artificial neural networks

In the previous section, we considered the subject domain and the corresponding ontology of ANN, within which the ANN architecture is described. However, the formalization of the ANN architecture is not enough to use ANN as a method of solving problems; the system should perform some *actions* on the configuration of the architecture, training and usage (interpretation) of ANN.

A. Action for processing the ANN

An ***action for processing the ANN*** is the maximum class of objects of research of the subject domain of actions for processing artificial neural networks. This concept denotes a class, to which all entities that denote specific actions for processing the ANN belong.

Within the framework of the *OSTIS Technology*, an *action* is defined as a purposeful *process* performed by one or more subjects (cybernetic systems) with the possible usage of certain tools [1].

Therefore, the *action for processing the ANN* is an action, the object of which is some ANN, the subject of which is an intelligent system, in the knowledge base of which this ANN is described.

Actions for processing the ANN are carried out by the appropriate group of agents [1].

Depending on whether the ANN is a sign of an entity external to the system memory, the elements of the set of the actions for processing the ANN are either elements of the set *action performed by a cybernetic system in its environment* or an element of the set *action performed by a cybernetic system in its memory*.

Let us consider the hierarchy of classes of actions for processing the ANN in the SCn-code.

action for processing the artificial neural network
:= [action for processing the ANN]
:= [action with the artificial neural network]
⊂ *action*
⇒ *decomposition**:
  {• *action for configuring the ANN*
   • *action for configuring the weight coefficients of the ANN*
   • *action for interpreting the ANN*
  }

action for configuring the ANN
⊂ *action for processing the ANN*
⇒ *decomposition**:
  {• *action for creating the ANN*
   • *action for editing the ANN*
   • *action for deleting the ANN*
   • *action for configuring the ANN layer*
     ⇒ *decomposition**:
       {• *action for adding a layer to the ANN*
        • *action for editing a layer of the ANN*
        • *action for deleting a layer of the ANN*
        • *action for setting the activation function of the neurons of the ANN layer*
        • *action for configuring a neuron in the ANN layer*
          ⇒ *decomposition**:
            {• *action for adding a neuron to the ANN layer*
             • *action for editing a neuron in the ANN layer*
             • *action for deleting a neuron from the ANN layer*
             • *action for setting the activation function of a neuron in the ANN layer*
            }
       }
  }

action for configuring the weight coefficients of the ANN
⊂ *action for processing the ANN*
⊃ *action for training the ANN*
⊃ *action for initializing the weights of the ANN*
  ⊃ *action for initializing the weights of neurons of the ANN layer*
    ⊃ *action for initializing the weights of the neuron of the ANN*

Since as a result of the actions for processing the ANN, the object of these actions, a particular ANN, can change significantly (the configuration of the network, its

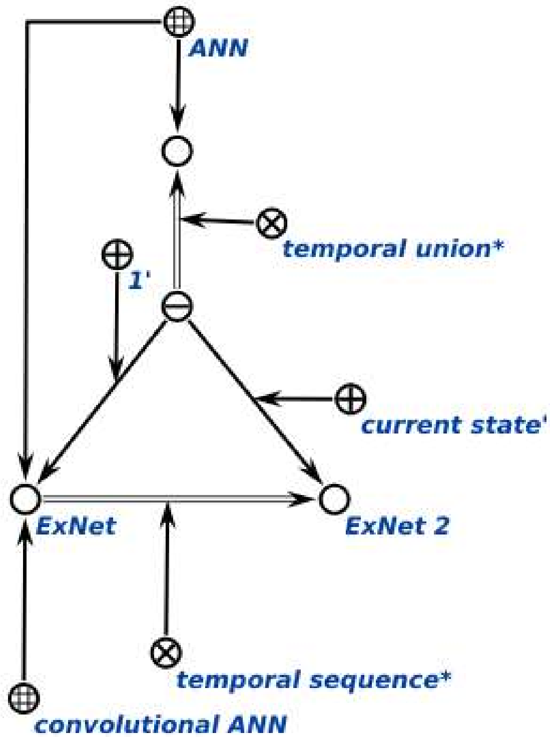


Figure 4. The representation of the artificial neural network as a temporal union of all its versions

weight coefficients change), the ANN is represented in the knowledge base as a temporal union of all its versions. Each version is the ANN and a temporal entity. On the set of these temporal entities, a temporal sequence is set with the indication of the first and last versions. Specific knowledge is described for each version. The knowledge common to all versions is described for the ANN, which is a temporal union of all versions.

Figure 4 shows an example of the representation of the ANN as a temporal union of all its versions.

B. Action for training the ANN

Special attention should be paid to the description of the ***action for training the ANN*** as a basic action, as a result of which it is possible to obtain an ANN suitable for usage as a method for solving problems.

An ***action for training the ANN*** is an action, during which a certain *method of training the ANN* is implemented with the specified *parameters of training the ANN*, *optimization method* and *loss function* on a given set of input data – a *training dataset*.

Let us consider the key concepts used to describe the action for training.

The ***method of training the ANN*** is a method of iterative search for optimal values of the configurable parameters of the ANN, which minimize some given loss function. It is worth noting that although the purpose of using the method of training is to minimize the loss function, the "utility" of the model obtained after training can be assessed only by the achieved level of its generalizing ability. The role relation *method of training'* is used to indicate the belonging of a particular method of training to any action for training the ANN.

Let us consider the hierarchy of methods of training the ANN in the SCn-code.

method of training the ANN
⊂ *method*
⊃ *method of training with a teacher*
  ⇒ *explanation**:
    [a ***method of training with a teacher*** is a method of training using the set target variables. **]**
  ⊃ *method of backward propagation of the error*
    := [MBPE]
    ⇒ *note**:
      [MBPE uses a certain optimization method and a certain loss function to implement the phase of backward propagation of the error and change the configurable ANN parameters. One of the most common optimization methods is the method of stochastic gradient descent. **]**
    ⇒ *note**:
      [It should also be noted that despite the fact that the method is classified as one of the methods of training with a teacher, in the case of using MBPE for training autoencoders, in classical publications, it is considered as a method of training without a teacher, since in this case there is no marked data. **]**
⊃ *method of training without a teacher*
  ⇒ *explanation**:
    [a ***method of training without a teacher*** is the method of training without using the set target variables (in the self-organization mode) **]**
  ⇒ *explanation**:
    [During the performance of the algorithm of the method of training without a teacher, useful structural properties of the set are revealed. Informally, it is understood as a method for extracting information from a distribution, the dataset for which was not manually annotated by a human [10]. **]**

An ***optimization method*** is a method for minimizing the target loss function during training the ANN. When denoting the belonging of a particular optimization method to any method of training the ANN, the role relation *optimization method'* is used.

optimization method
⊂ *method*
⇒ *definition**:
[an ***optimization method*** is a method for minimizing the target loss function during training the ANN]
⇒ *inclusion**:
- *SGD*
  := [stochastic gradient descent]
  := [SGD]
- *Nesterov method*
- *AdaGrad*
  := [adaptive gradient]
- *RMSProp*
  := [root mean square propagation]
- *Adam*
  := [adaptive moments]

The ***loss function*** is a function used to calculate the error, which is calculated as the difference between the factual reference value and the predicted value being obtained by the ANN. When denoting the belonging of a particular loss function to any method of training the ANN, the role relation *loss function'* is used.

loss function
⊂ *function*
⇒ *inclusion**:
{• *MSE*
  := [mean square error]
• *BCE*
  := [binary cross entropy]
• *MCE*
  := [multi-class cross entropy]
}

A ***parameter of training the ANN*** is a group of the most general parameters of the method of training the ANN. The following parameters of training the ANN are distinguished:
- the ***learning rate*** is a parameter that determines the rate of change in the weight coefficients of synaptic connections of the ANN in the training process.
- The *moment parameter* is a parameter used in the training process to eliminate the problem of the "blocking" of the training algorithm in the local minima of the minimized loss function. When training the ANN, the situation of stopping of the process at a certain point of the local minimum without achieving the desired level of generalizing ability of the ANN is frequent. To eliminate such an undesirable phenomenon, an additional parameter (moment) is introduced, which allows the training algorithm to "jump" through the local minimum and continue the process.
- The ***regularization parameter*** is a parameter used to control the level of retraining the ANN. ***Regularization*** is the addition of extra restrictions to the rules for changing the configurable ANN parameters to avoid retraining.
- The ***size of the training group*** is the size of the group from the dataset that is used to change the weight coefficients of synaptic connections at each elementary step of training.
- The ***training epoch*** is one iteration of the training algorithm, during which all the images from the training dataset were used once.

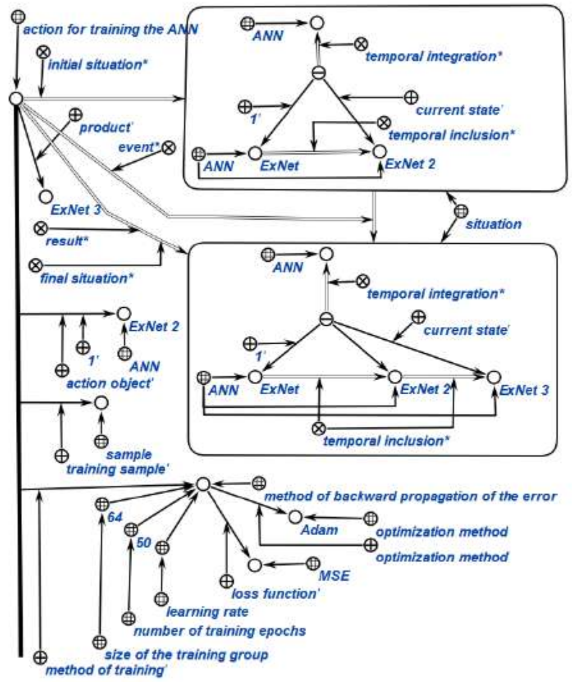


Figure 5. An example of the formalization of the action for training the artificial neural network in the knowledge base

With the help of the concepts highlighted in the subject domain under consideration, it becomes possible to formalize a specific action for processing the ANN in the knowledge base. Figure 5 shows an example of the formalization of the action for training the ANN.

Conclusion

A model for the representation of artificial neural networks and actions for their processing in the knowledge base on the ground of the OSTIS Technology has been developed. The model is represented in the form of subject domains and corresponding ontologies of artificial neural networks and their processing actions.

Within the framework of the described subject domains, with the help of their ontologies, it becomes possible to describe ANN in the knowledge base with all the variety of their architectures, classes, methods of training, etc.

The presence of such a possibility allows creating intelligent systems that can:

- use neural network methods along with other methods available in the system to solve complex problems in a common memory;
- configure, train and interpret ANN within the knowledge base for the purpose of their introspection and a deeper analysis of their work.

The described model creates the basis for further research in the field of developing:

- universal integration with the knowledge base of any neural network models, whose architecture is not formalized directly in the knowledge base itself;
- a group of agents capable of performing the described actions for processing ANN;
- an approach to automatic decision-making on the usage of a particular neural network model for solving system problems;
- an approach to using the knowledge base to improve the training results of artificial neural networks.

Acknowledgment

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help and valuable comments. In particular, Vladimir Adamovich Golovko, Vladimir Vasilyevich Golenkov, Daniil Vyacheslavovich Shunkevich, Aleksandr Aleksandrovich Kroshchenko and Egor Vladimirovich Mikhno.

References

[1] D. Shunkevich, "Agentno-orientirovannye reshateli zadach intellektual'nyh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 119–132.

[2] D. Castelvecchi, "Can we open the black box of ai?" *Nature News*, vol. 538, no. 7623, Oct 2016.

[3] T. R. Besold, A. d'Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Kuehnberger, L. C. Lamb, D. Lowd, P. M. V. Lima, L. de Penning, G. Pinkas, H. Poon, and G. Zaverucha, "Neural-symbolic learning and reasoning: A survey and interpretation," Nov. 2017, (accessed 2020, Jun). [Online]. Available: https://arxiv.org/pdf/1711.03902.pdf

[4] A. d'Avila Garcez, T. R. Besold, L. de Raedt, P. Földiak, P. Hitzler, T. Icard, K.-U. Kühnberger, L. C. Lamb, R. Miikkulainen, and D. L. Silver, "Neuralsymbolic learning and reasoning: Contributions and challenges," *In: McCallum, A., Gabrilovich, E., Guha, R., Murphy, K. (eds.) Proceedings of the AAAI 2015 Propositional Rule Extraction under Background Knowledge 11 Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches. AAAI Press Technical Report*, vol. SS-15-03, 2015.

[5] V. A. Golovko, V. V. Golenkov, V. P. Ivashenko, V. V. Taberko, D. S. Ivaniuk, A. A. Kroshchanka, and M. V. Kovalev, "Integration of artificial neural networks and knowledge bases," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, ser. 2, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 133–145.

[6] V. Golenkov, N. Gulyakina, I. Davydenko, and D. Shunkevich, "Semanticheskie tekhnologii proektirovaniya intellektual'nyh sistem i semanticheskie associativnye komp'yutery [Semantic technologies of intelligent systems design and semantic associative computers]," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 42–50, 2019.

[7] I. Davydenko, "Semantic models, method and tools of knowledge bases coordinated development based on reusable components," in *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2018, pp. 99–118.

[8] (2021, Jun) IMS.ostis Metasystem. [Online]. Available: https://ims.ostis.net

[9] V. A. Golovko and V. V. Krasnoproshin, *Nejrosetevye tekhnologii obrabotki dannyh [Neural network data processing technologies]*. Minsk: Publishing House of the BSU, 2017.

[10] J. Goodfellow, I. Benjio, and A. Courville, *Deep learning*. Moscow: DMK Press, 2017.

Модель представления искусственных нейронных сетей и действий по их обработке в базе знаний

Ковалёв М. В.

Статья посвящена рассмотрению проблемы интеграции искусственных нейронных сетей с базами знаний. Обоснована необходимость в интеграции и разработке модели представления искусственных нейронных сетей(и.н.с.) и действий по их обработке в базе знаний как основы такой интеграции. Описаны предметные области и соответствующие им онтологии и.н.с. и действий по обработке и.н.с.

Received 30.08.2021

Development of an aggregator for choosing the best forecasting method for groups of time series

Vadim Moshkin, Irina Moshkina, Dmitry Yashin, Nadezhda Yarushkina
Information Systems department, Faculty of Information Systems and Technologies
Ulyanovsk State Technical University, UlSTU
Ulyanovsk, Russia
v.moshkin@ulstu.ru, i.timina@ulstu.ru, taurusrulez@yandex.ru, jng@ulstu.ru

***Abstract*—The paper presents a general algorithm for computing an aggregated time series forecast (TS), within which machine learning methods are used to adjust the parameters of a hybrid combined forecasting model. Also presented are the results of experiments on the application of the developed algorithm using the TS competition "Computational Intelligence in Forecasting" (CIF). The use of a neural network for choosing forecasting methods allowed, on average, for all experiments to reduce the error by 7.225%, as can be seen from the results of the experiments. The average error for the eight prediction methods chosen by the neural network turned out to be lower than the average error for all methods in 47 cases out of 50 (94%)**

***Keywords*—Time series, forecasting, aggregated forecast, machine learning**

I. Overview of forecast aggregation methods

There are many methods for predicting vehicles. Combined models are used to take advantage of several methods at once. According to [2], a combined forecasting model is a forecasting model consisting of several individual models, called a base set.

In [3], a number of factors are highlighted that emphasize the effectiveness of the combined model:

1. impossibility of choosing a single model based on experimental data, according to the theory of multiple models [4];
2. an attempt to choose the only best model leads to the need to choose from a group of models with similar statistical characteristics [5];
3. The choice of a forecasting model for a vehicle with a pronounced dynamics of level and properties leads to the choice of an averaged model [2]. It is impossible to quickly replace one forecasting model with another by analyzing its dynamics;
4. each forecasting model considers only one side of the dynamics of the analyzed process. The set of models allows a more detailed description of the dynamics. Any forecast rejected due to non-optimality contains information important for modeling [5].

According to [2], combined forecasting models are divided into selective and hybrid ones.

In the selective model, the current predicted value is calculated from the selected value according to the selective criterion of the model from the base set.

The selective criterion can be:

- the minimum of the absolute value of the forecast error of the current member of the series
- minimum of the absolute value of the error for the last K observations (K-test)
- the minimum of the exponentially smoothed squared deviation error (B-criterion).

Thus, when using a selective model, at each moment in time, the forecast is built according to a single method selected from the basic set, hybrid models, in turn, make it possible to build a forecast using several models at once, using the advantages of their joint application.

In a hybrid model, the predicted value is obtained by aggregating the predicted results from several models from the base set. As a rule, the final forecast is a weighted sum of individual forecasts.

In [6], for the first time, the idea of creating a hybrid model based on combining forecasts of several statistical models was substantiated; in [7], this idea was developed, and it was proposed to use the arithmetic mean of forecasting results of the models included in the base set [3] as the final forecast.

According to [5], the main problem of constructing hybrid forecasting models is to determine the optimal weights of individual forecasting models from the base set.

In [3], the following main directions of development of hybrid forecasting models are identified:

1. inclusion in the basic set of new (emerging) forecasting models
2. development of new methods for combining forecasts.

There are 7 main groups of methods for combining forecasts [8]:

1. Methods based on the arithmetic mean of particular predictions [6] [9] [10]. However, the presence of anomalous forecasts as part of a combined forecast

significantly reduces its accuracy [11]. It is proposed to exclude anomalous predictions by using a truncated arithmetic mean [12] [13].

2. Methods based on minimizing the final forecast error by the least squares method [14].
3. Methods based on minimizing the variance of the combined forecast error (works by Bates and Granger [6], Ershov [15], Baltrushevich [16]).
4. Methods based on retrospective forecasts. This group includes the AFTER method [17]. The weights of the private forecasts are calculated based on their own past values, conditional variance, and the past values of the private forecasts. The weights are updated after each new observation.
   The following disadvantages of the AFTER method were noted in [8]:
   - difficult applicability in practice;
   - strong dependence of the weights on the first set value.

   This group includes the following methods:
   - ARM, developed by Yang [18];
   - the Bunn method [19], which assumes finding the distribution function for the weight coefficient through the beta distribution;
   - an adaptive method based on exponential smoothing [2], [20].
5. Methods based on factor analysis. These methods were proposed by Frenkel [21] and Gorelik and Frenkel [5]. The idea of using factor analysis is based on the fact that particular forecast results using a separate forecasting method are an external expression of some really existing but immeasurable forecast value, which is taken as a combined forecast [8].
6. The method of Gupta and Wilton, based on finding the optimal weights of the coefficients of particular predictions using a matrix of pairwise preferences, has been placed in a separate group [22].
7. Methods based on quadratic programming. The paper [23] describes a method for calculating the weights of particular predictions by minimizing the retrospective relative errors of particular predictions using quadratic programming methods.

The main advantage of the method is efficiency and ease of implementation. The main disadvantage is the obligatory preliminary selection of particular forecasting methods in order to comply with the requirement of error independence [8].

Most of the methods for combining forecasts are based on the assumptions about the independence of the absolute forecast errors and their distribution in accordance with the normal law with zero mathematical expectation and unknown variance. However, these assumptions are often not met [3], and therefore, methods based on fuzzy logic and stable statistical estimates are currently being actively developed, for example:

1. method of combining forecasts by Kovalev [24] based on a system of fuzzy rules;
2. the Davydov union method [25], based on the use of a robust M-estimate;
3. Methods for combining particular forecasts by Vasiliev [26] based on the robust Huber estimate of the truncated mean type and on the basis of the Hodges-Lehmann R-estimate.

Thus, despite a significant number of publications on the topics of forecasting methods for time series and methods for aggregating individual forecasts, the question of choosing the most appropriate aggregating method and its constituent forecasting models for the predicted time series remains.

II. Developed algorithm for calculating the aggregated forecast of time series

Figure 1 shows a schematic description of the developed algorithm for calculating the aggregated forecast of time series.

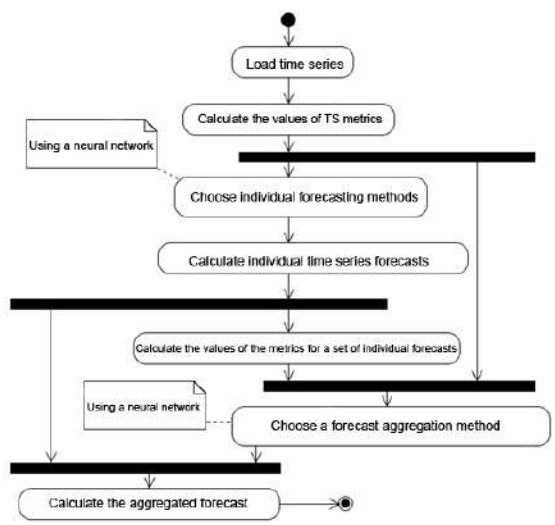


Figure 1. Algorithm for calculating the aggregated forecast of time series

In this paper, 2 methods of setting the forecast weights are considered:

- the first method is based on the values of the prediction error on the control part of the time series;
- the second method is based on the error values assumed by the neural network for choosing a prediction method.

The structure of the neural network for choosing the aggregating method is close to the structure of the neural network for choosing individual prediction methods, but it includes more input neurons corresponding to the metrics. Neurons corresponding to individual prediction methods have been replaced with neurons corresponding to aggregation methods. The structure of the neural network of the aggregating method is shown in Figure 2.

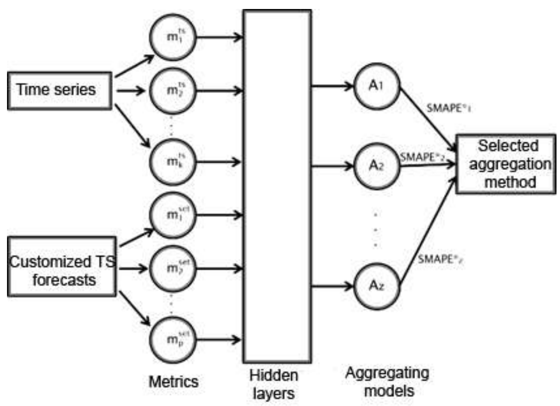


Figure 2. The structure of the neural network for choosing the aggregating method

Input neurons $m_1^{ts}, \ldots, m_k^{ts}$ correspond to time series metrics. Neurons $m_1^{set}, \ldots, m_p^{set}$ set correspond to the metrics of the aggregated set of individual predictions. The output values correspond to the estimated prediction error (SMAPE) values calculated by the neural network for each aggregator $(A_1, \ldots, A_z)$ from the base set. The aggregator from the base set of the combined model with the lowest estimated error value is used to obtain the final forecast.

The following main reasons for choosing just such a set of metrics for the input layer of the neural network can be distinguished:

- it is difficult to correctly train a neural network if the input neurons correspond to individual forecasting methods, since different individual methods will be selected for each time series. This means that the values of the signals arriving at the input of the neural network under consideration will be equal to zero for the unselected methods;
- the choice of the aggregator depends on the values of the time series metrics, but transitively;
- the direct dependence of the choice of an aggregator based on particular forecasting results is implemented through the metrics of a set of individual forecasts.

An error backpropagation algorithm with a logistic activation function is used to train the neural network. The training sample file contains the metric values and prediction errors (SMAPE) for each time series included in the set for each aggregation method.

This method of setting the weights includes dividing the time series into training and control parts, followed by forecasting by each individual method using the training part of the control values and calculating the prediction error. The weights of individual forecasts as part of the aggregator are set in inverse proportion to the magnitude of the error of each method.

III. Software system and experimental results

The developed program is designed to solve the problem of obtaining an aggregated forecast for the time series of the states of a technical system.

The software product is developed on the .NET Framework 4.6.1 platform in the C # language. The development environment was Microsoft Visual Studio 2015.

The "neuralnet" library [27] for the R language was used to work with neural networks. It made it possible to create neural networks with the structure of a multilayer preceptron, trained by the method of back propagation of the error (ordinary or elastic propagation). This library has a user-friendly interface and a high degree of configuration flexibility, allowing you to select the activation function and the error function.

The R library "ForecastComb" was used to compute the aggregated forecast. This library contains more than 12 aggregation methods (fig. 3).

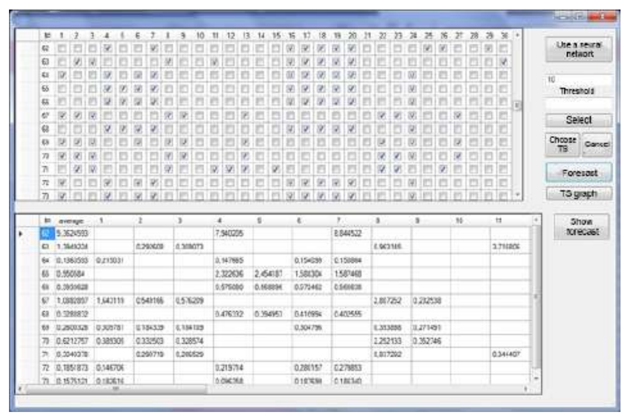


Figure 3. Forecast aggregation system form

Time series from the Computational Intelligence in Forecasting (CIF) competition [28] were selected to test the effectiveness of the developed solution.

- The first CIF benchmark contains 91 time series of different lengths (from 12 to 1089 observations) and different frequency of observations: day, month, quarter, year.
- The second CIF benchmark includes 72 time series with a frequency of a month and a length of 28 to 120 observations.

Five experiments with identical algorithm were carried out. The averaged final result was obtained for them.

1. A set of 152 time series of the benchmark was randomly divided into training (142 time series) and control parts (10 time series) during each experiment.
2. The time series of the control part were excluded from the general table of the training sample.

3. The neural network for choosing prediction methods (with automatic selection of the optimal number of neurons) was trained using the remaining data.
4. The resulting neural network was used to select the 8 best forecasting methods from the base set for the time series of the control part of the general table [29].

Figure 4 is a diagram showing, for each of the five experiments, the average SMAPE error for best practices and SMAPE for all methods.

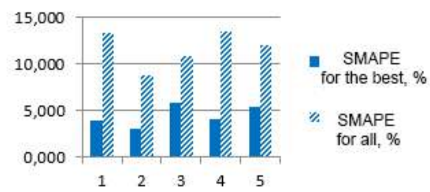


Figure 4. Results of the conducted experiments

IV. Conclusion

The use of a neural network for choosing forecasting methods allowed, on average, for all experiments to reduce the error by 7.225%, as can be seen from the results of the experiments. The average error for the eight prediction methods chosen by the neural network turned out to be lower than the average error for all methods in 47 cases out of 50 (94%).

Acknowledgment

This work was supported by the RFBR. Projects No. 18-47-730022, No. 19-07-00999 and 18-47-732007.

References

[1] Averkin, A.N. Soft computations and measurements // Intelligent systems. 1997.Vol. 2, no. 1-4. pp. 93-114.
[2] Lukashin, Yu.P. Adaptive methods of short-term forecasting of time series / Moscow, Finance and statistics, 2003. – 416 p.
[3] Vasiliev, A.A. Genesis of hybrid forecasting models based on the combination of forecasts /Bulletin of TVGU. Series "Economics and Management". - 2014. - No. 23. - pp. 316-331
[4] Ivakhnenko A.G. Inductive method of self-organization of models of complex systems: monograph / Kiev: Naukova Dumka, 1981 .—296 p.
[5] Gorelik N.A. Statistical problems of economic forecasting / N.A. Gorelik, A.A. Frenkel // Statistical methods of analysis of economic dynamics: textbook. app. on statistics, vol. 46. - M .: Nauka, 1983. - pp. 9-48.
[6] Bates J.M., Granger C.W.J. The Combination of Forecasts. – Oper. Reser. Quart., 1969, v.20, No. 4. P. 451-468
[7] Newbold P., Granger C.W.J. Experience with Forecasting Univariate Time Series and Combination of Forecasts. – J. of Royal Statistical Society. Ser. A. 1974, v. 137, No. 2. P. 131-164
[8] Frenkel, A.A. Methodological approaches to improving forecasting accuracy by combining forecasts / A.A. Frenkel, A.A. Surkov. - M: Questions of statistics. - 2015. - No. 8. - pp. 17-36
[9] Barnard G.A. New methods of quality control //J. R. Statist. Soc.. 1963. Vol. 126. P. 255-259
[10] Makridakis S. and Winkler R.L. Averages of forecasts: some empirical results //Management Science. 1983. Vol. 9. P. 987-996
[11] Winkler, R.L., &Clemen, R.T. Sensitivity of weights in combining forecasts // Operations Research. 1992. Vol. 40. P. 609-614
[12] Goodwin P. New evidence on the value of combining forecasts // FORESIGHT. 2009. Vol. 12. P. 33-35
[13] Jose V.R.. and Winkler R.L. Simple robust averages of forecasts: some empirical results //International Journal of Forecasting. 2008. Vol. 24. P. 163-169
[14] Granger C.W.J. and Ramanathan R. Improved methods of combining forecasts // Journal of Forecasting. 1984. Vol. 3. P. 197-204
[15] Ershov E.B. On one method of combining private forecasts // In the book: Statistical methods of analysis of economic dynamics. Uch. app. according to statistics / E.B. Ershov. Moscow: Nauka, 1973.T. XXII-XXIII. pp. 87-105
[16] Baltrushevich, T.G. Models and methods for assessing the effectiveness of flexible production systems // Avtoref. dis. ... Cand. econom. sciences. - M., 1991.pp. 17-20.
[17] Yang Y. Combining forecasting procedures: some theoretical results // Department of Statistics and Statistical Laboratory Iowa State University. 2001
[18] Yang Y. Adaptive regression by mixing //Journal of American Statistical Association. 2001b. Vol. 96. P. 574-588
[19] Bunn D.W. A Bayesian approach to the linear combination of forecasts // Operational Research Quarterly. 1975. Vol. 26. P. 325-329
[20] Dubrova T.A. Statistical analysis and forecasting of economic dynamics: problems and approaches // In the book: Methodology of statistical research of socio-economic processes. - M .: Unity, 2012.pp. 129-138.
[21] Frenkel A.A. Forecasting labor productivity: methods and models. - M .: Economics, 1989.pp. 142-154.
[22] Gupta S. and Wilton P. C. Combination of forecasts: an extension // Management Science. 1987. Vol. 3. pp. 356-371
[23] Beilinson, Ya.E. Combined forecast models / Ya.E. Beilinson, M.A. Motova // Expert information. Series: Modeling of socio-economic processes. M., 1990. Issue. 2.pp.110-121.
[24] N. G. Yarushkina ; A. A. Filippov ; A. A. Romanov ; V. S. Moshkin ; E. N. Egov Developing a system for time series data mining on the basis of F-transform and the domain-specific ontology // Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), 2017 Joint 17th World Congress of International//IEEE, pp.1-6.
[25] Davydov, A.A. Forecasting social phenomena using "neural" networks / Sociological methods in modern sociological practice: collection of articles. materials Vseross. scientific. conf. in memory of A.O. Kryshtanovsky. - M .: GU-HSE, 2008 .– pp. 41-49.
[26] Vasiliev, A.A. Forecasting in logistics based on stable hybrid models / A.A. Vasiliev, V.M. Kurganov // Logistics: modern development trends: materials of the X international. scientific-practical conf. April 14, 15, 2011; ed. count .: V.S. Lukinsky (editor-in-chief) [and others]. SPb .: SPbGIEU, 2011. - pp. 52-55
[27] Yashin D., Moshkina I., Moshkin V. (2020) An Approach to the Selection of a Time Series Analysis Method Using a Neural Network. In: Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2020. ICCSA 2020. Lecture Notes in Computer Science, vol 12249. Springer, Cham. pp 692-703, https://doi.org/10.1007/978-3-030-58799-4_50
[28] Ayvazyan, S.A. Applied statistics and foundations of econometrics / S.A. Ayvazyan, V.S. Mkhitaryan. - M.: UNITI, 1998 .– 1024 p.
[29] Yarushkina N., Andreev I., Moshkin V., Moshkina I. (2019) Integration of Fuzzy OWL Ontologies and Fuzzy Time Series in the Determination of Faulty Technical Units. In: Misra S. et al. (eds) Computational Science and Its Applications – ICCSA 2019. Lecture Notes in Computer Science, vol 11619. Springer, Cham, pp 545-555, https://doi.org/10.1007/978-3-030-24289-3_40

Разработка агрегатора для выбора наилучшего метода прогнозирования групп временных рядов

Мошкин В.С., Мошкина И.А.,
Яшин Д.С., Ярушкина Н.Г.

В статье представлен общий алгоритм вычисления агрегированного прогноза временных рядов, в рамках которого используются методы машинного обучения для настройки параметров гибридной комбинированной модели прогнозирования. Также представлены результаты экспериментов по применению разработанного алгоритма с использованием временных рядов конкурса «Вычислительный интеллект в прогнозировании» (CIF). Использование нейронной сети для выбора методов прогнозирования позволило в среднем по всем экспериментам снизить ошибку на 7,225%. Средняя ошибка для восьми методов прогноза, выбранных нейронной сетью, оказалась ниже средней ошибки для всех методов в 47 случаях из 50 (94%).

Received 06.05.2021

Ontological approach to the building of semantic models of user interfaces

Sadouski M.E.
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
sadovski@bsuir.by

***Abstract*—The article is dedicated to the description of the ontological approach to the building of the user interface based on the OSTIS Technology. The existing approaches in the field of the building of user interfaces are considered and an ontological model of support of the building process is described.**

***Keywords*—OSTIS, user interface, building of the user interface, ontological approach, semantic model, server-driven user interface**

I. Introduction

In the modern world and the daily life of humans, there is a growing need for the usage of computer systems. The effectiveness of their usage depends largely on the user interface, since it is the user interface as a component of the system that is a way to interact with the user.

At present, the user interface is the most frequently changed component of the system and is the part of the application that requires the maximum number of updates. Approximately 80% of the costs in the development of computer systems are accounted for by the design, testing and development of the user interface. At the same time, almost all applications change their interface after the release of the first version and the addition of new functionality always affects the already developed component. In addition, most modern systems should be cross-platform for the convenience of users, which implies the development of a web version of the interface as well as mobile and desktop versions [1].

When building the user interface, the following problems remain relevant:

- the portability of user interfaces from one implementation platform to another is difficult;
- the lack of general principles for building user interfaces limits the possibility of reuse of already developed components and increases the time required to train the user in new user interfaces, which also increases the development time and the cost of designing and supporting user interfaces;
- in most systems, there is no possibility of modifying the user interface during running;
- in most systems, there is no possibility of flexible adaptation of the user interface to the needs of a particular user.

Within the framework of this article, an ontological approach to the building of semantic models of user interfaces based on the OSTIS Technology is proposed to solve the above problems. A technique for developing a user interface is also proposed. The article elaborates the ideas proposed in [2] and is aimed at more detailed consideration of the problem of the automatic building of the user interface, which is a key one to support the flexibility and simplicity of improving the designed interfaces.

The building of the user interface within the framework of the proposed approach will be carried out on the basis of its complete semantic model that contains a precise specification of all the used concepts with the help of a hierarchical system of formal ontologies, which will ensure the integration of various aspects of the user interface within a unified system, the ability of the system to analyze the actions performed within the user interface and its flexible configuration in the process of operation. Thus, the development of the user interface will be reduced to the building and improvement of its semantic model.

II. Analysis of existing approaches

Currently, there are several basic approaches to generating a user interface, that consider its various aspects:

- an approach based on specialized description languages;
- a context-sensitive approach;
- a data-based approach;
- an ontological approach.

The approach based on specialized description languages assumes the representation of a particular user interface in a platform-independent form. Examples of interface description languages are **UIML** [3], **UsiXML** [4], **XForms** [5] and **JavaFX FXML** [6]. The key idea of the represented languages is to build a model of dialogues and interface forms in a form independent of the used technology, a description of visual elements as well as the relations between them and their features for creating a certain user interface.

The context-sensitive approach integrates the usage of a structural description of the interface based on description languages with a behavioral specification, that is, the generation of the interface is based on user actions. As part of the approach, transitions between different types of a particular user interface are specified. Examples of the implementation of this approach are **CAP3** [7] and **MARIA** [8].

A data-based, or a model-oriented, approach uses a model of the subject domain as the basis for creating user interfaces. Implementation includes **JANUS** [9] and **Mecano** [10].

Existing ontological approaches are usually based on the approaches presented earlier and use ontologies as a way of representation of information about a particular user interface. For example, by analogy with the approach based on specialized description languages, the framework [11] was proposed, which uses an ontology to describe the user interface based on concepts stored in the knowledge base. By analogy with the context-dependent approach, within the framework of the article [12], the model of the subject domain together with the user interface model is used, associated with the ontology of actions. The **ActiveRaUL** [13] project combines UIML with a model-oriented approach. Within the framework of this project, the ontological model of the subject domain is correlated with the ontological representation of the user interface. The approach proposed in [14] combines application data with the user interface ontology for the creation of a single description in the knowledge base for the subsequent automatic generation of various interface options for questionnaire applications with ready-made user interaction scenarios. It is also worth noting the articles [15] and [16], in which a concept is proposed, that allows combining information that is homogeneous in content into components of the interface model, liberating the interface developer from encoding, and forming information for each component of the interface model using editors controlled by the corresponding ontology models.

The principle of generating an interface based on a declarative description is the basis of a number of applied projects. For example, the **mermaid** [17] project allows automatically generating diagrams based on their description, and the **rjsf** [18] project allows generating forms for user input. In addition, a general approach to generating and displaying an interface based on its description from the server side of the application can also be found in industrial development under the terms **Server-Driven UI** or **Backend-Driven UI** [19].

The disadvantages of existing solutions for generating the user interface include the following:

- as a rule, the created models are specific to a particular platform or a certain implementation of the user interface, which hinders their reuse for other purposes;
- solutions that offer a platform-independent description allow generating only simple user interfaces that are limited in functionality (questionnaire applications, diagrams, etc.).

Among the represented approaches, the ontological one is the most preferable for the following reasons:

- it allows integrating earlier proposed approaches due to a single way of representation of knowledge;
- it allows creating the most complete description of various aspects of the user interface. The composition of this description will be discussed in more detail below;
- it simplifies the reuse of the interface by applying a single representation of the interface model for different platforms.

However, for existing solutions based on the ontological approach, the problem of compatibility of various ontologies within a unified system remains relevant as well as the lack of the ability to adapt to user requests and analyze their actions for independent improvement (as a basis for such an analysis, the ontology of user actions serves).

III. Proposed approach

Based on the conducted analysis, an approach on the ground of an ontological one is proposed, which consists in creating a complete semantic model of the interface, which will eliminate the shortcomings of existing solutions. The key features of the approach are:

- the fixation of the interface description in the form of an abstraction, regardless of the platform and device;
- the presence of a complete semantic model of the interface, that contains a "lexical" interface description (a description of the components, from which the interface is formed), a "syntactic" interface description (rules for forming a correct and full interface from its components) but also its semantic description (knowledge of which entity the displayed component a sign is). At the same time, the semantic description also includes the purpose, scope of application of the interface components and a description of the interface user activity;
- the representation of the specification of the interface generation tools and, if necessary, the tools themselves in a common format with a description of the interface through some unified knowledge representation language;
- the reduction of development costs due to the reuse of interface components;
- the reduction of development costs due to the usage of a hierarchical structuring of the user interface model, which allows independent development of components;
- universality, that is, the possibility of using the approach to build interfaces of any systems, regard-

less of their purpose. The unified principles of the building of the interface will allow the user to easily switch from using one system to another, significantly reducing the cost of training;
- the integration of the semantic interface model with other models within a unified system. For example, integration with the user model (biographical information, knowledge about the user's behavior within the system) will make the interface adaptive. In this case, an adaptive interface is understood as an interface that can adapt to a certain user or category of users (which implies not only a change in the visual component of the interface but also a change in its internal functionality).

Thus, based on the above, the following demands can be made to the technology, on the basis of which this approach can be implemented:

- the technology should provide an opportunity to describe various semantic models and their components of various types of knowledge in a common format;
- the technology should allow the simple integration of various semantic models within a unified system;
- the technology should support a component approach to creating semantic models.

Among the existing system design technologies, the *OSTIS Technology* meets the specified requirements, among the advantages of which it is also possible to additionally highlight the presence of a basic set of ontologies that can serve as the basis for the user interface model being developed.

Thus, within the framework of this approach, in the article, an option for implementing a framework for building user interfaces is proposed, which is based on the *OSTIS Technology*, which provides a universal language for the semantic representation (encoding) of information in the memory of intelligent computer systems, called an *SC-code*. Texts of the *SC-code* (sc-texts) are unified semantic networks with a basic set-theoretic interpretation. The elements of such semantic networks are called *sc-elements* (*sc-nodes* and *sc-connectors*, which, in turn, can be *sc-arcs* or *sc-edges*, depending on the directivity). The *SC-code alphabet* consists of five main elements, on the basis of which SC-code constructs of any complexity are built, as well as more particular types of sc-elements (for example, new concepts) are introduced. The memory that stores SC-code constructs is called semantic memory or *sc-memory* [20].

The architecture of each ostis-system includes a platform for interpreting semantic models of ostis-systems as well as a semantic model of the ostis-system described using the SC-code (sc-model of the ostis-system). In turn, the sc-model of the ostis-system includes the sc-model of the knowledge base, the sc-model of the problem solver and the sc-model of the interface. The principles of the structure and design of knowledge bases and problem solvers are discussed in more detail in [21] and [23], respectively. Within this article, the sc-model of the user interface will be considered, which is included in the sc-model of the interface. Its principles were described in the article [2], the development of which is this article.

The SC-code representation languages include:

- **SCg-code** – one of the possible ways of visual representation of SC-texts. The basic principle that underlies the SCg-code is that each sc-element is assigned a sc.g-element (graphical representation);
- **SCs-code** – a string (linear) version of the SC-code representation. It is designed to represent sc-graphs (SC-code texts) in the form of sequences of characters;
- **SCn-code** – a string non-linear version of the SC-code representation. The SCn-code is designed to represent sc-graphs in the form of sequences of characters formatted according to given rules, in which basic hypermedia tools, such as graphic images as well as tools of navigation between parts of sc.n-texts, can also be used [2].

Within the framework of this article, fragments of SCg- and SCn-codes [24] will be used, which are simultaneously fragments of the source texts of the knowledge base, that are understandable to both a human and to a machine. This allows making the text more structured and formalized while maintaining its readability.

IV. Problem definition

The user interface within the framework of the proposed approach is a specialized ostis-system focused on solving interface problems and that includes a knowledge base and a problem solver of the user interface. The general architecture of the ostis-system is shown in figure 1.

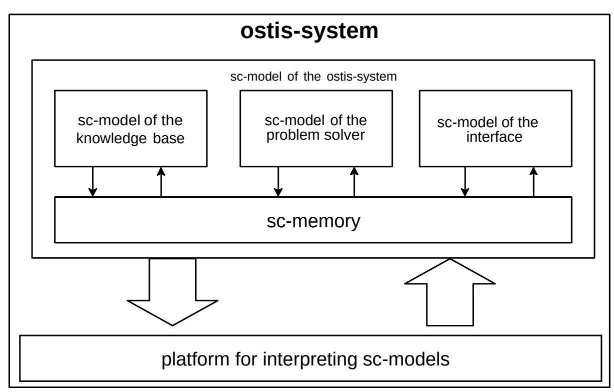


Figure 1. The architecture of the ostis-system

To solve the problem of building a user interface, the user interface knowledge base requires the presence of an sc-model of user interface components, interface actions as well as the classification of user interfaces in general, as shown in figure 2. When designing the interface, it is proposed to use a component approach, which assumes the representation of the entire application interface in the form of separate specified components that can be developed and improved independently. It is important that, as a result of the building, the user interface should be not only static (visually formed) but also dynamic (with the ability to perform various actions, including those initiated by the user).

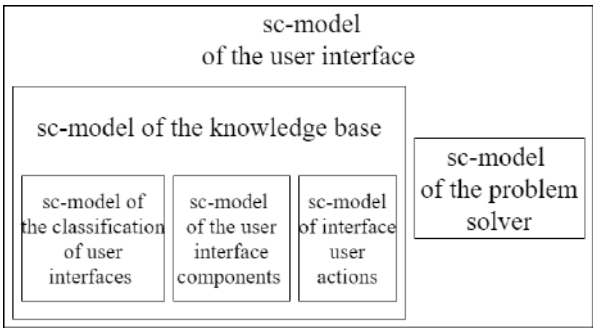


Figure 2. The structure of the sc-model of the user interface

The basis of the *sc-model of the ostis-system knowledge base* is a hierarchical system of subject domains and their corresponding ontologies. Accordingly, within the framework of the proposed approach, it is necessary to develop:

- the Subject domain of user interfaces;
- the Subject domain of user interface components;
- the Subject domain of interface user actions.

The problem solver is a hierarchical system of agents of knowledge processing in semantic memory (sc-agents), which interact with each other exclusively by specifying the actions they perform in the memory. An sc-agent is a certain subject that can perform actions in sc-memory, which belong to a certain class of autonomous actions. An autonomous action is an action that is performed regardless of whether the specified action is part of the decomposition of a more common action [22]. To build a user interface, it is necessary to implement the following agents:

- the Agent of interpretation of the sc-model of the user interface knowledge base;
- the Agent of processing of user actions.

V. Sc-model of the knowledge base

A. Subject domain of user interfaces

The subject domain of user interfaces includes the classification of user interfaces.

user interface
⊃ *command-line interface*
⊃ *graphical user interface*
  ⊃ *WIMP-interface*
⊃ *SILK-interface*
  := [(Speech – речь, Image – образ, Language – язык, Knowledge – знание)]
  ⊃ *natural-language interface*
    ⊃ *speech interface*

A *user interface* is one of the most important components of a computer system. It is a set of hardware and software tools that provide information exchange between the user and the computer system.

A *command-line interface* is a user interface, in which information is exchanged between a computer system and a user by writing text instructions or commands.

A *graphical user interface* is a user interface, in which information is exchanged between a computer system and a user using the graphical components of a computer system.

A *WIMP-interface* is a user interface, in which information is exchanged between a computer system and a user in the form of a dialogue using windows, menus and other controls.

A *SILK-interface* is a user interface that is closest to the natural form of human communication. The computer system initiates commands independently, analyzing human speech and finding key phrases in it. The result of running commands is converted into a form that is understandable to a human, for example, into a natural-language form or an image.

A *natural-language interface* is a SILK-interface, in which the exchange of information between a computer system and a user occurs through a dialogue. The dialogue is conducted in one of the natural languages.

A *speech interface* is a SILK-interface, in which information is exchanged through a dialogue, during which the computer system and the user communicate using speech. This type of interface is the closest to natural communication between humans.

B. Subject domain of user interface components

The subject domain of user interface components describes the structure and features of the visual representation of user interface components. The Ui2Ont ontology [26] was taken as the basis of this subject domain.

A *user interface component* is a sign of a fragment of the knowledge base, that has a certain form of external representation on the screen.

user interface component
⇒ *decomposition**:
  { • *atomic user interface component*
    • *non-atomic user interface component*
  }

An *atomic user interface component* is a user interface component that does not contain other user interface components in its structure.

A *non-atomic user interface component* is a user interface component that consists of other user interface components.

Below is the classification of the components:

user interface component
⊃ *presentation user interface component*
  ⊃ *output*
    ⊃ *image-output*
    ⊃ *graphical-output*
      ⊃ *chart*
      ⊃ *map*
      ⊃ *progress-bar*
    ⊃ *video-output*
    ⊃ *sound-output*
    ⊃ *text-output*
      ⊃ *headline*
      ⊃ *paragraph*
      ⊃ *message*
  ⊃ *decorative user interface component*
    ⊃ *separator*
    ⊃ *blank-space*
  ⊃ *container*
    ⊃ *menu*
    ⊃ *menu-bar*
    ⊃ *tool-bar*
    ⊃ *status-bar*
    ⊃ *table-row-container*
    ⊃ *list-container*
    ⊃ *table-cell-container*
    ⊃ *tree-container*
    ⊃ *labeled-group*
    ⊃ *tab-pane*
    ⊃ *spin-pane*
    ⊃ *tree-node-container*
    ⊃ *scroll-pane*
    ⊃ *window*
      ⊃ *modal-window*
      ⊃ *non-modal-window*
⊃ *interactive user interface component*
  ⊃ *data-input-component*
    ⊃ *data-input-component-with-direct-feedback*
      ⊃ *text-input-component-with-direct-feedback*
        ⊃ *multi-line-text-field*
        ⊃ *single-line-text-field*
      ⊃ *slider*
      ⊃ *drawing-area*
      ⊃ *selection-component*
        ⊃ *selection-component-multiple-values*
        ⊃ *selection-component-single-values*
      ⊃ *selectable-data-representation*
        ⊃ *check-box*
        ⊃ *radio-button*
        ⊃ *toggle-button*
        ⊃ *selectable-item*
    ⊃ *data-input-component-without-direct-feedback*
      ⊃ *spin-button*
      ⊃ *speech-input*
      ⊃ *motion-input*
  ⊃ *presentation-manipulation-component*
    ⊃ *activating-component*
    ⊃ *continuous-manipulation-component*
      ⊃ *scrollbar*
      ⊃ *resizer*
  ⊃ *operation-trigger-component*
    ⊃ *command-selection-component*
      ⊃ *button*
      ⊃ *menu-item*
    ⊃ *command-input-component*

A *presentation user interface component* is a component of the user interface that does not imply interaction with the user.

A *decorative user interface component* is a user interface component designed to style the interface.

A *container* is a user interface component, whose task is to place a set of components included in its structure.

A *window* is a separate screen panel that contains various elements of the user interface. Windows can be placed on top of each other.

A *modal-window* is a window that blocks the user experience with the application until the user closes the window.

A *non-modal-window* is a window that allows the user to interact with other windows without having to close this window.

An *interactive user interface component* is a user interface component that is used to interact with the user.

A *data-input-component* is a user interface component designed for input of information.

A *presentation-manipulation-component* is a user interface component designed to represent information and interact with the user.

An *operation-trigger-component* is a user interface component that requests the user to perform some action.

A non-atomic component is connected to its constituent components using the *decomposition** relation. Here is an example of a description of a non-atomic component of the main window. The appearance of the display of this component is shown in figure 3.

The formalization of this component in the SCn-language looks like in the following manner:

MainPage
∈ *window*
⇒ *decomposition**:
  {• *Navigation*
    ∈ *non-atomic user interface component*
    ⇒ *decomposition**:

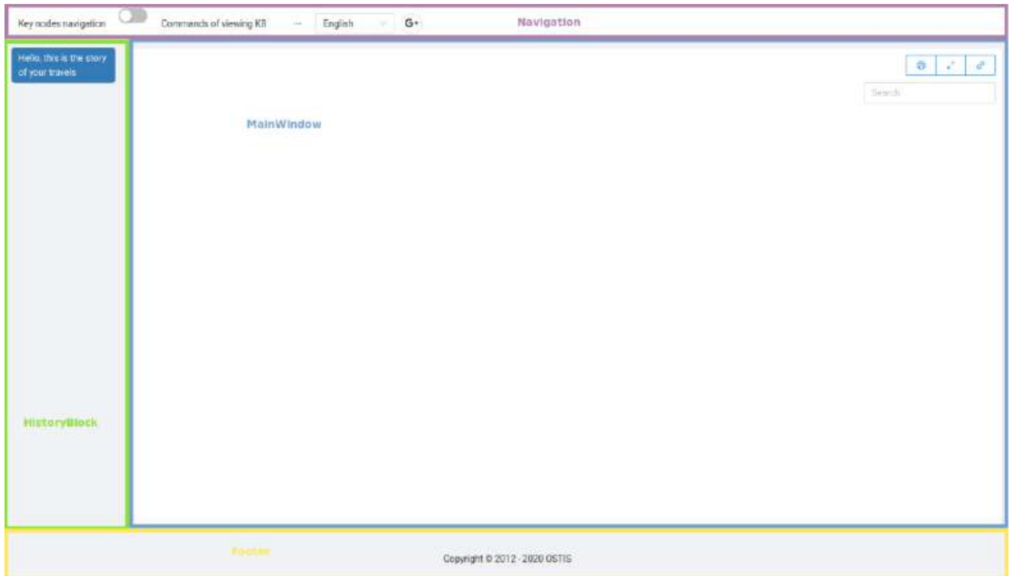


Figure 3. An example of the display of a non-atomic component

{• *mainMenu*
  ∈ *menu*
  ⇒ *decomposition**:
    {• *item1*
        ∈ *menu-item*
     • *item2*
        ∈ *menu-item*
     • *switch*
        ∈ *toggle-button*
    }
 • *languageSelect*
    ∈ *selection-component-single-values*
 • *googleAuth*
    ∈ *button*
}
• *HistoryBlock*
  ∈ *non-atomic user interface component*
• *MainBlock*
  ∈ *non-atomic user interface component*
  ⇒ *decomposition**:
  {• *mainWindow*
      ∈ *window*
   • *tool-bar*
      ⇒ *decomposition**:
      {• *printButton*
          ∈ *button*
       • *resizeButton*
          ∈ *button*
       • *linkButton*
          ∈ *button*
       • *searchInput*
          ∈ *single-line-text-field*
      }
  }
• *Footer*
  ∈ *non-atomic user interface component*
}

The subject domain of user interface components also contains a description of the properties of the components.

As part of the work on the knowledge base of the *IMS.ostis Metasystem* [24], the subject domain of spatial entities and their forms was created. The *IMS.ostis Metasystem* is an intelligent metasystem built according to the standards of the OSTIS Technology and aimed at usage by ostis-system engineers – at supporting the design, implementation and updating (reengineering) of ostis-systems – and at developers of the OSTIS Technology – at supporting collective activities for the development of standards and libraries of the OSTIS Technology. In the subject domain of spatial entities and their forms, there are descriptions of such concepts as:

- spatial entity;
- form;
- coordinate system;
- Cartesian coordinate system;
- two-dimensional Cartesian coordinate system;
- point of reference;
- point;
- segment;
- length;
- thickness;
- height;
- width;
- rectangle.

The subject domain of user interface components intersects with the subject domain of spatial entities and their forms and adds a set of concepts to describe the properties of components, part of which is given below.

*Text** is a binary relation that connects a user interface component to a file that contains the text of the user interface component.

Color is a parameter of the user interface component that determines its color.

Text color is a parameter of the user interface component that determines the color of its text.

Text size is a parameter of the user interface component that determines the size of its text.

Text font is a parameter of the user interface component that determines the font of its text.

The *deactivation property* is a logical parameter of a user interface component that can be set to inhibit the usage of the component until a certain action is performed.

The *maximum number of characters* is a parameter of the text-input-component-with-direct-feedback component, which sets the maximum number of characters that can be input by the user.

Thus, within the framework of this subject domain, both component classes and their instances are described as well as the properties of components for visualization, regardless of the platform. At the same time, these components and properties are easily changeable and extensible.

C. Subject domain of interface user actions

The Subject domain of interface user actions contains a specification of user actions, which can be performed for the components of the user interface. The Ui2Ont ontology [26] was used as the basis of this subject domain.

An *interface user action* is a minimally meaningful fragment of some activity of the user, performed through the interface.

Next is the classification of interface user actions.

interface user action
- ⊃ *mouse-action*
  - ⊃ *mouse-scroll*
    - ⊃ *mouse-scroll-up*
    - ⊃ *mouse-scroll-down*
  - ⊃ *mouse-hover*
  - ⊃ *mouse-drop*
  - ⊃ *mouse-click*
    - ⊃ *mouse-double-click*
    - ⊃ *mouse-single-click*
  - ⊃ *mouse-gesture*
  - ⊃ *mouse-unhover*
  - ⊃ *mouse-drag*
- ⊃ *speech-action*
- ⊃ *keyboard-action*
  - ⊃ *press-function-key*
  - ⊃ *type-text*
- ⊃ *tangible-action*
- ⊃ *touch-action*
  - ⊃ *touch-click*
    - ⊃ *touch-single-click*
    - ⊃ *touch-double-click*
  - ⊃ *touch-gesture*
    - ⊃ *one-fingure-gesture*
    - ⊃ *multiple-finger-gesture*
  - ⊃ *touch-drop*
  - ⊃ *touch-drag*
- ⊃ *pen-base-action*
  - ⊃ *touch-function-key*
  - ⊃ *draw*
  - ⊃ *write-text*

A *mouse-hover* is the interface user action, which corresponds to the appearance of the mouse cursor within the user interface component.

A *mouse-drop* is the interface user action, which corresponds to dropping some component of the user interface within another user interface component using the mouse.

A *mouse-gesture* is an interface user action, which corresponds to the performance of a certain gesture through the movement of the mouse.

A *mouse-unhover* is an interface user action, which corresponds to the exit of the mouse cursor outside the framework of the user interface component.

A *mouse-drag* is an interface user action, which corresponds to dragging a user interface component with the mouse.

A *tangible-action* is an interface user action performed using taction.

A *touch-action* is an interface user action performed using the sensor.

A *touch-gesture* is an interface user action, which corresponds to the performance of a certain gesture with the movement of fingers on the screen of the sensor.

A *one-fingure-gesture* is an interface user action, which corresponds to the performance of a certain gesture by moving one finger on the screen of the sensor.

A *multiple-fingure-gesture* is an interface user action, which corresponds to the performance of a certain gesture by moving several fingers on the screen of the sensor.

A *touch-drop* is an interface user action, which corresponds to dropping a certain component of the user interface within another component of the user interface using a sensor.

A *touch-drag* is an interface user action, which corresponds to dragging a certain component of the user interface using a sensor.

A *pen-base-action* is an interface user action performed using a pen on a graphics tablet.

A *touch-function-key* is an interface user action, which corresponds to pressing the function key of the graphic tablet with a pen.

The above user actions are common to all systems. It should be noted that the interface user action, as a rule, initiates some internal action of the system.

internal action of the system
⊃ *internal action of the ostis-system*

In the case of ostis-systems, as part of the work on the knowledge base of the IMS.ostis Metasystem [24], the *Subject domain and ontology of actions, problems, plans, protocols and methods implemented by the ostis-system in its memory as well as internal agents that perform these actions* was allocated. A fragment of this subject domain is shown below.

internal action of the ostis-system
:= [an action in sc-memory]
:= [an action performed in sc-memory]

Each *internal action of the ostis-system* denotes some transformation performed by some *sc-agent* (or a group of *sc-agents*) and focused on the transformation of *sc-memory*.

action in sc-memory
⊃ *action in sc-memory initiated by a question*
⊃ *action of editing the ostis-system knowledge base*
⊃ *action of setting the mode of the ostis-system*
⊃ *action of editing a file stored in sc-memory*
⊃ *action of interpreting a program stored in sc-memory*

An *action in sc-memory initiated by a question* is an action aimed at forming an answer to the raised question.

To define an action that is initiated when interacting with the user interface, the *action initiated by the user interface** relation is used.

action initiated by the user interface*
∈ *quasi-binary relation*
∈ *oriented relation*
⇒ *first domain**:
user interface component ∪ *user interface action class*
⇒ *second domain**:
class of internal actions of the system

The first component of the binding of the *action initiated by the user interface** relation is a binding, the elements of which are an element of the set of user interface components and an element of the *user interface action class* set. The second component is an element of the *class of internal actions of the system* set. An example of using this relation is shown in figure 4.

Thus, within the framework of these subject domains, interface user actions and internal actions of the system are described. These actions are basic and can be easily expanded and refined.

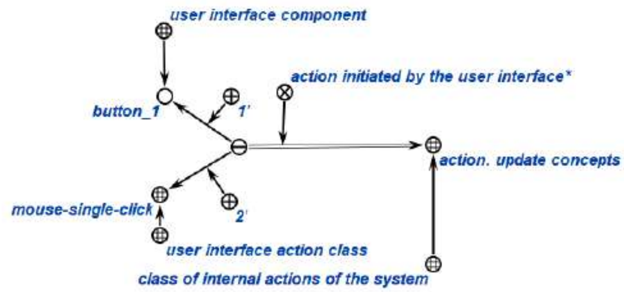


Figure 4. An example of using the action initiated by the user interface* relation

The integration of the abovementioned ontologies allows implementing an approach, within the framework of which:

- all the components of the user interface correspond to a certain fragment of the knowledge base. It allows addressing various questions about these components to the system. As examples of such questions, the following ones can act: "what class of components does the specified component belong to?", "what is the specified component designed for?", "what does the specified component consist of", etc.;
- the classification of components allows the further building of the user interface taking into account the knowledge about them. For example, the presentation user interface component can be highlighted in one color and the interactive user interface component – in another;
- it is possible to accumulate the results of the interface user activity to further adapt the interface for them. Changing the interface is reduced to changing its sc-model, which can be carried out on the basis of logical rules, which are also described in the system knowledge base. For example, the system may contain a logical rule for adding the most frequently initiated interface user actions to a separate component;
- it is possible to analyze the efficiency of user actions for further improvement of the interface (for example, several interface actions performed by the user in a row can be replaced by one);
- the user will have the opportunity to receive answers to questions about the organization of interface activities. Examples of such questions include: "what interface actions can be performed for the specified component?", "what interface actions were performed the most often?", etc.

VI. Sc-model of the problem solver

From the point of view of processing the sc-model of the user interface knowledge base, the following problems should be solved:

- the interpretation of the sc-model of the user interface knowledge base (building the user interface);
- the processing of user actions.

User interface problem solver
$\Leftarrow$ *decomposition of an abstract sc-agent**:
{
- *Agent of interpretation of the sc-model of the user interface knowledge base*
- *Agent of processing of user actions*

}

An *Agent of interpretation of the sc-model of the user interface knowledge base* accepts an instance of the user interface component for displaying as an input parameter. In this case, the component can be either atomic or non-atomic (for example, a component of the main application window). The result of the operation of the agent is a graphical representation of the indicated component, taking into account the used implementation of the platform for interpreting semantic models of ostis-systems.

The operation algorithm of this agent is as follows:

- the input component type (atomic or non-atomic) is checked;
- if the component is atomic, then to display its graphical representation based on the properties specified for it. If this component is not included in the decomposition of any other component, to complete the performance. Otherwise, to determine the component, the decomposition of which includes the considered component, apply its properties to the current atomic component and start processing the found non-atomic component, going to the first item;
- if the component is non-atomic, then to check whether the components, into which it was decomposed, were displayed. If yes, then to complete the performance, otherwise to determine the component from the decomposition of the non-atomic component being processed, which has not yet been displayed, and start processing the found component by going to the first item.

An *agent of processing of user actions* is a non-atomic agent that includes many agents, each of which processes user actions of a certain class (for example, an agent of processing a mouse click action, an agent of processing a mouse drop action, etc.). The agent reacts to the appearance of an instance of an interface user action in the knowledge base of the system, finds an internal action class associated with it and generates an instance of this internal action for subsequent processing.

VII. Implementation of the proposed approach

The current implementation of the sc-model interpretation platform is web-oriented [27].

Taking into account the features of the platform and for the possibility of integrating the proposed approach with existing solutions in the field of building user interfaces, it is proposed to implement the agent of interpretation of the sc-model of the user interface knowledge base as a non-atomic agent that is decomposed into the agent of translation of the sc-model of the user interface knowledge base into a format compatible with existing solutions and the agent of displaying the specified format in the graphical representation of the user interface.

It is proposed to use the JSON format as an intermediate description format for a number of reasons:

- it is the most popular format for data transmission and storage in modern systems;
- the compact and simple syntax;
- the simplicity of making changes;
- the simplicity of transmission and processing.

Thus, an additional *agent of translation of the description of the user interface component from the sc-model to the JSON format* is introduced. As an input parameter, this agent accepts an instance of the translation user interface component in the JSON format. The JSON description is formed by recursively processing the description of components from non-atomic to atomic ones.

The *Agent of displaying the specified format in the graphical representation of the user interface* is non-atomic and is decomposed into a set of agents that perform displaying for various interpretation platforms (web, mobile, desktop computer platforms, etc.). As input, this agent accepts a description of the user interface component in the JSON format. The result of the operation of the agent is a graphical representation of the user interface.

For the possibility of changing the sc-model of the user interface, editing tools such as the SCg-, SCs- and SCn-editors are implemented within the OSTIS Technology. So, the framework proposed within the approach includes three key parts:

- an sc-model of the user interface;
- tools of visualization of the sc-model of the user interface;
- tools of editing of the sc-model of the user interface.

The general structure of the framework is shown in figure 5.

VIII. Technique of developing user interface components

One of the advantages of the proposed approach is the accumulation of frequently used components. To do this, it is supposed to create a library of components with preset properties, which is included in the subject domain of user interface components. The components included in the library are platform-independent (they can be visualized regardless of the used interpretation platform).

The process of creating an instance of a user interface component can be described as follows:

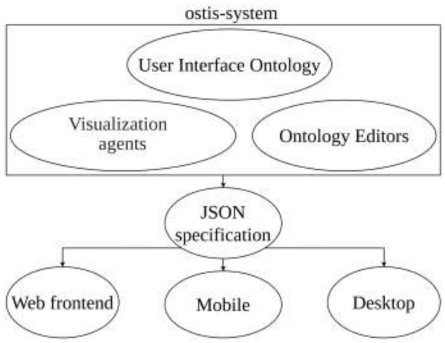


Figure 5. The structure of the framework for generating the user interface

- to check whether the class of the necessary component is present in the subject domain of the user interface components;
- if it is not available, it is necessary to create a class of the necessary component, specifying the set of properties for it;
- to check whether there is a description of the instance of the required component in the component library;
- if it is missing, it is required to supplement the component library with a description of a new instance of the component with preset properties;
- if necessary, to create a new instance of the class of the necessary component, setting it certain properties and actions based on the ontology of the subject domain of a particular system;
- to run the agent of visualization of the component instance for the used interpretation platform.

The described process is shown in figure 6.

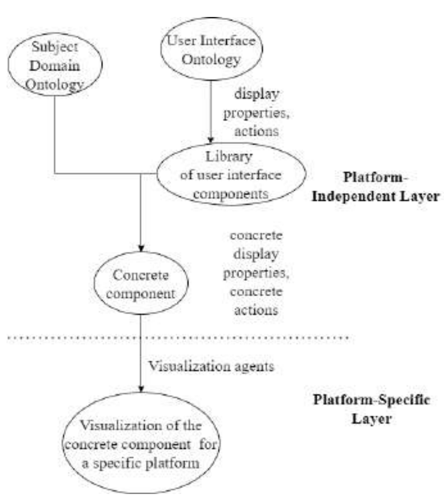


Figure 6. The process of creating an instance of a user interface component

IX. Examples of the framework operation

Next, we will give some examples of the description of interface components in the knowledge base of the system in the SCg-language and the result of their visualization. Figure 7 shows the formalization of the "button" component, figure 8 shows the result of its display in the web interface. In figures 9, 11, the description of the "text field" and "form" components is presented, in figures 10, 12 is their display, respectively.

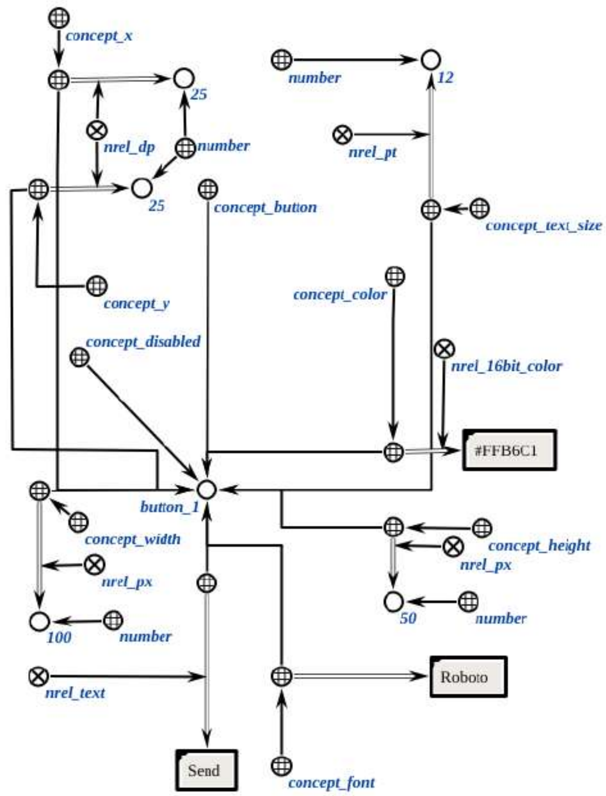


Figure 7. The formalization of the "button" component

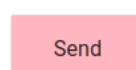


Figure 8. The result of the display of the "button" component in the interface

X. Conclusion

In this article, an ontological approach to the building of semantic models of user interfaces based on the OSTIS Technology is proposed.

The analysis of existing approaches to the building of user interfaces is carried out, the structure and technique of building user interface components for the framework proposed within the approach are presented. Examples of the ontological description of the interface components

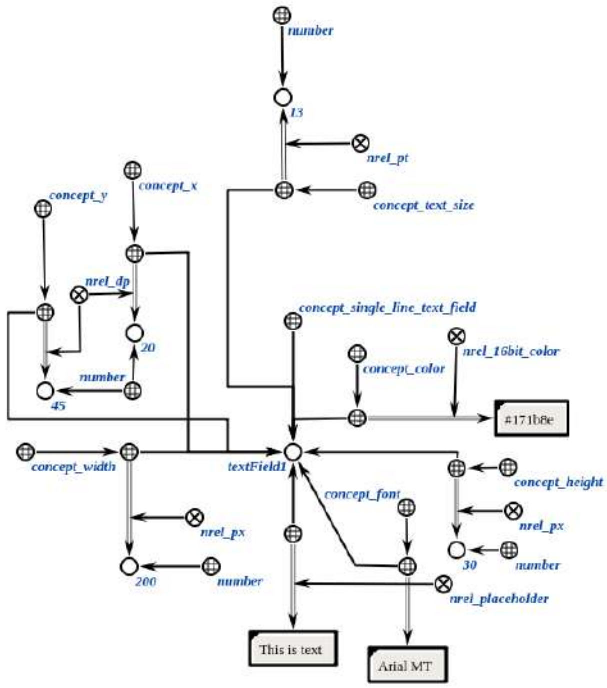


Figure 9. The formalization of the “text field” component

Figure 10. The result of the display of the “text field” component in the interface

and the results of their visualization using the developed tools for the automatic building are also presented.

In contrast to the existing approaches, the proposed approach will allow:

- taking into account the semantics of the user interface components when building it;
- generating questions to the system related to the user interface;
- taking into account the history of the interface user activity to improve the quality of their work with the system;
- rebuilding the user interface by changing its model during the operation of the system.

At this stage, according to the proposed approach, the following were implemented:

- a fragment of the subject domain of user interface components;
- a fragment of the subject domain of interface user actions;
- an agent of translation of the description of the user interface component from the sc-model to the JSON format;
- an agent of displaying the JSON format in a graphical representation of the user interface for a web platform.

As part of further work, it is planned to expand specified subject domains and implement visualization agents for other platforms.

Acknowledgment

The author would like to thank the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics for the help and valuable comments.

References

[1] Data-driven UI: unlimited power [Electronic resourse]. Access mode: https://mobius-piter.ru/en/2018/spb/talks/v96lokugwe8cwggio8ois/ Date of access: 21.05.2021.

[2] Boriskin A. S., Sadouski M. E., Koronchik D. N., Zhukau I. I., Khusainov A. F. Ontology-Based Design of Intelligent Systems User Interface. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems], Minsk, 2017, pp. 95–106

[3] Abrams M., Phanouriou C., Batongbacal A. L., Williams S. M., Shuster J. E. UIML: An appliance-independent XML user interface language. 99 Proceedings of the eighth international conference on World Wide Web, 1999, pp. 1695–1708

[4] Limbourg Q. USIXML: A User Interface Description Language Supporting Multiple Levels of Independence. Matera, M. and Comai, S. (eds.) ICWE Workshops, Rinton Press, 2004, pp. 325–338

[5] XForms 1.1 [Electronic resourse]. Access mode: https://www.w3.org/TR/xforms Date of access: 25.05.2021.

[6] Introduction to FXML [Electronic resourse]. Access mode: https://openjfx.io/javadoc/12/javafx.fxml/javafx/fxml/doc-files/introduction_to_fxml.html Date of access: 23.05.2021.

[7] Van den Bergh J., Luyten K., Coninx K. CAP3: Context-Sensitive Abstract User Interface Specification. Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems – EICS’11, 2011, pp. 31–40

[8] Paterno F., Santoro C., Spano L.D. Maria: A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environment. ACM Trans. Comput. Interact. 16, 2009.

[9] Balzert H., Hofmann F., Kruschinski V. The JANUS Application Development Environment—Generating More than the User Interface. Computer Aided Design of User Interfaces, Vol. 96, 1996, pp. 183–206

[10] Puerta A.R., Eriksson H., Gennari J.H., Musen M.A. Beyond data models for automated user interface generation. In Proceedings British HCI’94, 1994.

[11] Liu B., Chen H., He W. Deriving user interface from ontologies: A model-based approach. Int. Conf. Tools with Artif. Intell. ICTAI, 2005, pp. 254–259

[12] Gaulke W., Ziegler J. Using profiled ontologies to leverage model driven user interface generation. 7th ACM SIGCHI Symp. Eng. Interact. Comput. Syst. – EICS ’15, 2015, pp. 254–259

[13] Sahar A., Armin B., Shepherd H., Lexing L. ActiveRaUL : Automatically generated Web Interfaces for creating RDF data, 2013.

[14] Michael Hitz, Thomas Kessel Using Application Ontologies for the Automatic Generation of User Interfaces for Dialog-Based Applications. Research and Practical Issues of Enterprise Information Systems, CONFENIS 2016

[15] Gribova V. V., Cherkezishvili N. N. Avtomatizaciya razrabotki pol’zovatel’skih interfeysov s dinamicheskimi dannymi [Automating the development of user interfaces with dynamic data]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems], 2011, pp. 287–292 (in Russian)

[16] Gribova V.V, Tarasov A.V. Generator koda pol’zovatel’skogo interfejsa, upravlyaemyy ontologiey. Artificial Intelligence, Vol. 4, 2005, pp. 457 – 464 (in Russian)

[17] Mermaid documentation [Electronic resourse]. Access mode: https://mermaid-js.github.io/mermaid Date of access: 18.04.2021.

[18] React Json Schema Form [Electronic resourse]. Access mode: https://rjsf-team.github.io/react-jsonschema-form/ Date of access: 20.04.2021.

[19] Exploring Server-Driven UI [Electronic resourse]. Access mode: https://betterprogramming.pub/exploring-server-driven-ui-cf67b3da919 Date of access: 22.04.2021.

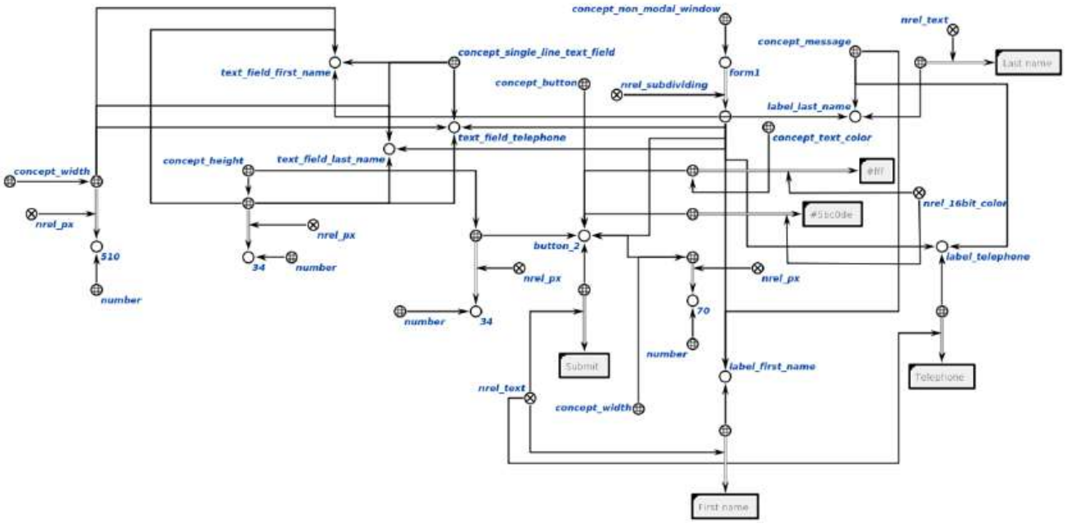


Figure 11. The formalization of the “form” component

Figure 12. The result of the display of the “form” component in the interface

[20] Golenkov V., Guliakina N., Davydenko I., Eremeev A. Methods and tools for ensuring compatibility of computer systems. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems], 2019, pp. 25–52

[21] Davydenko I. Semantic models, method and tools of knowledge bases coordinated development based on reusable components. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems], 2018, pp. 99–118

[22] Shunkevich D. Ontology-based Design of Knowledge Processing Machines. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems], 2017, pp. 73–94

[23] Shunkevich D. Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems], 2018, pp. 119–132

[24] IMS.ostis Metasystem [Electronic resourse]. Access mode: http://ims.ostis.net/ Date of access: 03.06.2021.

[25] Golenkov V. [et al.] Principles of organization and automation of the semantic computer systems development. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems], 2019, pp. 53-90

[26] Paulheim H., Probst F. UI2Ont – A Formal Ontology on User Interfaces and Interactions. Semantic Models for Adaptive Interactive Systems, 2013, pp. 1–24

[27] OSTIS Web Platform [Electronic resourse]. Access mode: https://github.com/ostis-dev/ostis-web-platform Date of access: 25.04.2021.

Онтологический подход к построению семантических моделей пользовательских интерфейсов

Садовский М. Е.

В работе предлагается онтологический подход к построению семантических моделей пользовательских интерфейсов на основе Технологии OSTIS.

Проведен анализ существующих подходов к построению пользовательских интерфейсов, приведена структура и представлена методика создания компонентов пользовательского интерфейса для предлагаемого в рамках подхода фреймворка. Также представлены примеры онтологического описания компонентов интерфейса и результаты их визуализации с помощью разработанных средств автоматического построения.

В отличие от существующих подходов предлагаемый подход позволит:

- учитывать семантику компонентов пользовательского интерфейса при его построении;
- формировать вопросы к системе, связанные с пользовательским интерфейсом;
- учитывать историю интерфейсной деятельности пользователя для повышения качества его работы с системой;
- перестраивать пользовательский интерфейс путем изменения его модели в процессе работы системы.

На данном этапе согласно предлагаемому подходу были реализованы:

- фрагмент предметной области компонентов пользовательских интерфейсов;
- фрагмент предметной области интерфейсных действий пользователя;
- агент трансляции описания компонента пользовательского интерфейса из sc-модели в JSON формат;
- агент отображения JSON формата в графическое представление пользовательского интерфейса для web-платформы.

В рамках дальнейшей работы планируется расширение указанных предметных областей и реализация агентов визуализации для других платформ.

Received 01.06.2021

Ontological approach to the integration of knowledge from external sources

Korshunov R.A., Sadouski M.E., Zagorskij A.G.
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
korshunov19101998work@gmail.com, sadovski@bsuir.by, alexandr.zagorskiy.work@gmail.com

***Abstract*—The article is dedicated to the problem of automatized provisioning of the knowledge base from external sources. Within the framework of the article, the classification of data sources by the type of stored data, the justification of the need to integrate data into the knowledge base, the analysis of existing approaches to solving the specified problem are given. Our own ontological approach is proposed as well as the implementation of the approach on the example of WikiData is considered.**

***Keywords*—OSTIS, ontological approach, integration of knowledge**

I. Introduction

A **knowledge base** is a finite information structure that is a formal representation of all the knowledge sufficient for the operation of a computer system and that is stored in its memory [1]. The knowledge base is a key component of any intelligent system. The main problem that it solves is the acceleration of the search for the necessary information, which is achieved through the accumulation of any intellectual accomplishments of experts in the subject domain.

Any knowledge base should be loaded to work with it. The process of loading the knowledge base has a number of features. Firstly, it is necessary to involve qualified experts who will be responsible for the correctness of the embedded knowledge. Secondly, it is necessary to spend a certain amount of labor and time resources.

Undoubtedly, any complication of certain processes causes a logical need to find less expensive solutions. Within the framework of this article, an automatized mechanism for integrating knowledge from external sources is proposed.

The implementation of this mechanism will allow solving a number of problems. Firstly, to reduce the time and labor costs for enriching the knowledge base. This may be particularly relevant when it is necessary to add a large amount of data to the knowledge base, which has a common, recurrent format (for example, data from tables of a relational database). Secondly, the proposed approach will allow using various external sources to load the knowledge base. This can be useful when it is necessary to unify several sources in one common resource. At the same time, such sources can radically differ from each other in the format of knowledge representation. Thirdly, the mechanism allows automating the process of integrating knowledge from external sources. As a result, the knowledge base of the intelligent system will be in a current, updated state.

From the point of view of the quality of knowledge in the knowledge base, the implementation of the proposed approach will allow:

- solving the problem of synonymy of source data, thereby reducing storage expenses and increasing the processing speed of the total knowledge base;
- unifying the format of information representation, which will allow producing new knowledge based on logical rules from already formed one;
- monitoring the consistency and quality of integrated knowledge.

II. Classification of data sources

Before getting to the proposed mechanism for integrating knowledge from external sources, it is worth paying attention to what these sources can be. Today, there are various data sources, but they generally can be classified according to the type of data they store. In turn, data can be divided into the following groups. This is [2]:

- structured data;
- unstructured data;
- semi-structured data;
- metadata.

Figure 1. The data types

A. Structured data

Structured data is data that corresponds to a predefined data model and can therefore be easily analyzed [2]. Structured data corresponds to a tabloid format with relations between different rows and columns. Common examples of structured data are Excel files or SQL databases. Each of them has structured rows and columns that can be sorted.

Structured data depends on a data model – a model of how data can be stored, processed and accessed. Because of the data model, each field is discrete and can be accessed separately or jointly with data from other fields. This makes structured data extremely powerful: it is possible to quickly aggregate data from various places in the database.

Structured data is considered as the most "traditional" form of data storage, since the earliest versions of database management systems (DBMS) could store, process and access structured data.

B. Unstructured data

Unstructured data is information that either does not have a predefined data model or is not organized in a predefined way [2]. Unstructured information usually contains a lot of text but can also contain such data as dates, numbers and facts. This leads to inaccuracies and ambiguities that make it difficult to understand the usage of traditional programs compared to data stored in structured databases. Unstructured data includes [3]:

- text sources: books, journals, presentations;
- media: MP3, digital photos, audio and video files;
- data from websites and social networks.

In recent years, the ability to store and process unstructured data has become much in demand, and many new technologies and tools have appeared on the market, which can store specialized types of unstructured data.

The ability to analyze unstructured data is especially relevant in the context of big data, since most of the data in organizations is unstructured.

C. Semi-structured data

Semi-structured data is a form of structured data, which does not correspond to the formal structure of data models connected to relational databases or other forms of data tables but nevertheless contains tags or other markers to separate semantic elements and provide a hierarchy of records and fields within data [2]. Therefore, it is also known as a self-describing structure. As examples of tools for representing semi-structured data, it is possible to distinguish [3]:

1. **XML markup language.** This is the language of semi-structured documents. XML is a set of rules for encoding documents that determine the format that can be read by a human and a computer (although the statement that XML is readable is of little consequence: anyone who tries to read an XML document can better master their time). Its value is that its tag-driven structure is very flexible and encoders can adapt it to generalize the data structure, storage and transport in the network.
2. **JSON** is another semi-structured data exchange format. Its structure consists of name/value pairs (or an object, hash table, etc.) and an ordered list of values (or an array, sequence, list). Since the structure is interchangeable between languages, JSON successfully handles data transfer between web applications and servers.
3. **NoSQL** is a type of database that differs from traditional ones in that they do not separate the organization (schema) from data. This makes NoSQL the best choice for storing information that does not fall within the format of a record and a table, for example, when the question is about the texts of various lengths. This also makes it easier to exchange data between databases. Some new NoSQL databases, such as MongoDB and Couchbase, also include partially structured documents, initially storing them in the JSON format.

The reason why this category exists (between structured and unstructured data) is that semi-structured data can much easier be analyzed than unstructured data. Many big data solutions and tools can "read" and process JSON or XML. This simplifies the analysis of structured data compared to unstructured data.

D. Metadata

Metadata is data about data [2]. From a technical point of view, this is not a separate data structure but one of the most important elements for big data analysis and big data solutions.

For example, in a set of photos, metadata can describe when and where the photos were taken. Then the metadata provides fields for dates and locations, which themselves can be considered as structured data. For this reason, metadata is often used by big data solutions for initial analysis.

From all the above, it can be concluded that as the simplest sources for integration, those will serve, which store structured data. This simplicity is based on the rigor of the model of such data. At the same time, the rigor of the model limits the data, makes it less diverse and makes it impossible to add something to them that does not fall within the framework of the model. In contrast to unstructured data, structured data is more flexible and diverse. However, the absence of any clear structure complicates the processing process and makes it impossible to write simple integration rules.

III. Overview of existing approaches

The problem of enriching the knowledge base is not new. Attempts to solve it have been made before. For example, a solution worthy of attention was proposed by the Bosch Center for AI.

The Bosch Center has developed a semantic data lake for automotive data as a centralized platform for developing and testing applications for autonomous driving. The architecture of the lake can be found in figure 2 [5].

The **"data lake"** is a centralized storage that allows storing all structured and unstructured data at any scale

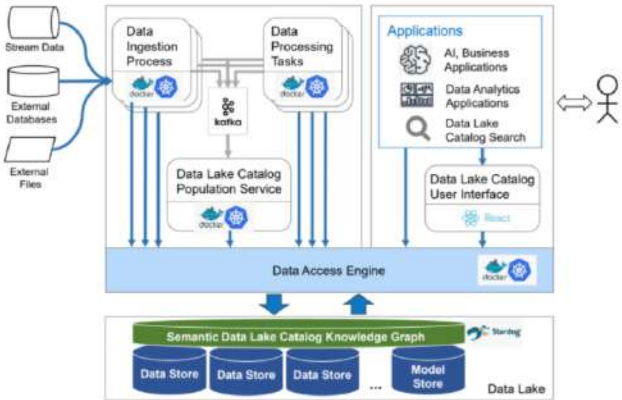


Figure 2. The architecture of the semantic data lake

[4]. A **"semantic data lake"** is a special form of data lake, in which the upper semantic layer enriches and connects data semantically. The semantic level overcomes the fragmentation of data and enables semantic search across all data [5]. The semantic data lake is more flexible than a regular data lake, which allows using the data stored in it more efficiently. This prevents the transformation of the data lake into a "data swamp" – a storage, in which potentially valuable information is simply stored on drives without usage.

A special DCPAC (Data Catalog, Provenance and Access Control) ontology was developed for data processing, which accumulates and coheres other W3C ontologies [5].

During the receiving and processing of new input data, the service for enriching the data lake catalog is launched, which automatically creates a semantic description and a layer on top of the data. The semantic level has the form of a knowledge graph. The service for enriching the data lake catalog reads the available metadata about the loaded data assets and creates the corresponding semantic data by reconciling, annotating and enriching the input data with the concepts of the DCPAC ontology. The resulting knowledge graph forms a semantic layer on the assets of the data lake and describes their contents, origin and access rights. This information plays a decisive role in the semantic data lake, since it semantically describes the stored data and, therefore, turns it into valuable information and information assets. This allows tracking aspects of the data related to the origin and determining access rights to data assets [5].

Semantic Web proposes to use its own technologies to integrate data from various sources. The essence of integration is to create mappings from the source format to the RDF [6] data model. For example, a special language R2RML [7] has been developed to display data from relational databases. An extended version of the language called RML also allows creating rules for displaying data from the CSV, TSV, XML and JSON formats [8].

It is also worth highlighting the approach described in the article [9]. It proposes the usage of a specially developed framework for integrating and supporting dynamic ontologies developed by means of Semantic Web.

Integration takes place with the help of several software components-wrappers. At first, the new ontology O_b is mapped to the initial one O_a, forming a third unified ontology O_c. The formed ontology O_c, as well as its two initial ones, are merged into a single integrated ontology O_d. At the same stage, semantic integration problems are being solved. Further, from the difference between the ontologies O_d and O_b, the specialist is offered to merge and create a new version of the knowledge base.

IV. Proposed approach

As mentioned earlier in Section II, data can be divided into 4 basic groups: structured, unstructured, semi-structured data and metadata. Various external languages are used to represent them. For example, natural languages are used to represent unstructured data. Formal languages are used to represent un-, semi-structured data and metadata.

Natural languages are understood as languages that developed in the usual way for communication between humans [10]. These include Russian, English and other existing languages. In turn, formal languages are languages created by a human to solve any specialized problems. These include all kinds of programming languages (C++, Python, Java), markup languages (XML, HTML), etc.

The essence of the proposed approach for the integration of knowledge is as follows. The address of some external source is input to the system. If the system supports the source format, then with the help of a special agent, it "pulls in" the proposed knowledge and converts it according to the format that is accepted within this system. Then, with the help of another agent, the system tries to integrate the acquired knowledge into the hierarchy of existing one.

Within the framework of the approach, it is necessary to solve some problems. **Firstly**, it is necessary to describe the syntax and semantics of the external language. **At the second step**, the problem is to build rules for the transition of knowledge from constructions in an external language to constructions in an internal language used in the system to represent knowledge. At this step, the previously obtained descriptions of syntax and semantics are applied. **The last stage** is the integration of the obtained knowledge with the knowledge base. This stage includes checking the correctness of the received constructions.

There may be difficulties at each of the stages. The first difficulty arises during the description of the syntax. Since each external language can have its own set of unique features that distinguish it from the internal language of the system and other external languages, it is necessary to describe the concepts, relations and rules for transiting the constructions of the external language into the constructions of the internal language. For these purposes, the subject domain for a specific external language should be described. This subject domain is inherited from the *Subject domain of information constructions and languages* and the *Subject domain of entities*.

In addition to the problems described above, there are difficulties in the final integration of the obtained constructions. It is not enough just to immerse them in memory – they need to be correctly implemented into existing knowledge while eliminating synonymy, contradictions and other problems.

V. Implementation of the proposed approach

As an example, the process of integrating knowledge from WikiData into the ostis-system knowledge base will be considered. **WikiData** is an open and free knowledge base. Its purpose is to represent the actual information from Wikipedia in a compatible, machine-readable format [11].

The OSTIS Technology is a complex (family) of technologies that provide design, production, operation and reengineering of intelligent computer systems (ostis-systems) designed to automate a variety of human activities and that are based on the semantic representation and ontological systematization of knowledge as well as agent-oriented knowledge processing [12].

Among the advantages of ostis-systems, it is possible to distinguish:

- the ability to implement semantic integration of knowledge in its memory at a high level;
- the ability to integrate different types of knowledge;
- the ability to integrate various problem-solving models.

The json-format is used to represent knowledge in WikiData. This is a text format for data exchange. Its basic syntactic units are:

- a "key-value" pair;
- an ordered set of values.

As an example, let us consider a json-fragment with information about the city of Minsk. It consists of three basic blocks:

- "entities";
- "relations";
- "triplets".

The block "entities" includes the entity "Minsk" as well as all the entities, with which "Minsk" is related. The block "relations" includes all relations, in which the entity "Minsk" participates. The block "triplets" includes all the triples "entity-relation-entity".

To add knowledge from WikiData to the ostis-system knowledge base, it is necessary to convert json to the SC-code. This is an internal language used in ostis-systems. For the operation of intelligent computer systems built on the basis of the SC-code, in addition to the method of abstract internal representation of knowledge bases (SC-code), there are several ways of external representation of abstract sc-texts that are convenient for users and used in the design of the source texts of the knowledge bases of these intelligent computer systems and the source texts of fragments of these knowledge bases, as well as used to display various fragments of knowledge bases to users according to user requests. As such methods of external display of sc-texts, the following ones are distinguished:

- SCg-code, the texts of which are graph structures of a common type;
- SCs-code, the texts of which are strings of characters;
- SCn-code, the texts of which are two-dimensional character matrices that are the result of formatting, two-dimensional structuring of the texts of the SCs-code.

To solve the integration problem, the usage of the ostis-system is advisable for many reasons. The technology initially provides tools for describing the syntax and semantics of external languages. This toolkit allows reducing the development time by several times.

To integrate knowledge, a special translator, as well as the *Subject domain of WikiData*, are required.

Subject domain of WikiData
∋ *class of study objects'*:
- *wiki-entity*
- *wiki-relation*

∋ *relation being studied'*:
- *wiki-identifier*
- *wiki-analog*

Let us consider the process of generating sc-constructions using the example of the entity "belarus" from the block "entities". First, an sc-node with the identifier "belarus" is created. Since "belarus" is located in the block "entities", an arc of belonging is generated between the class "wiki-entity" and the node "belarus". Then it is necessary to determine the wiki-identifier of this entity. For these purposes, an sc-link with the content "belarus" is generated, and then a relation "wiki-identifier*" is generated between the sc-link and the sc-node. At the end, it is necessary to generate the main identifier of the entity. To do this, the sc-link "Belarus" is generated, which belongs to the class "English language". Then the relation "main identifier*" between the sc-node of the entity and the sc-link is generated. The same procedure is repeated for all other entities.

The process of generating relations from the block "relations" is similar to generating entities, just this once, instead of the class "wiki-entity", an arc of belonging is generated between the class "wiki-relation" and the sc-node of the relation.

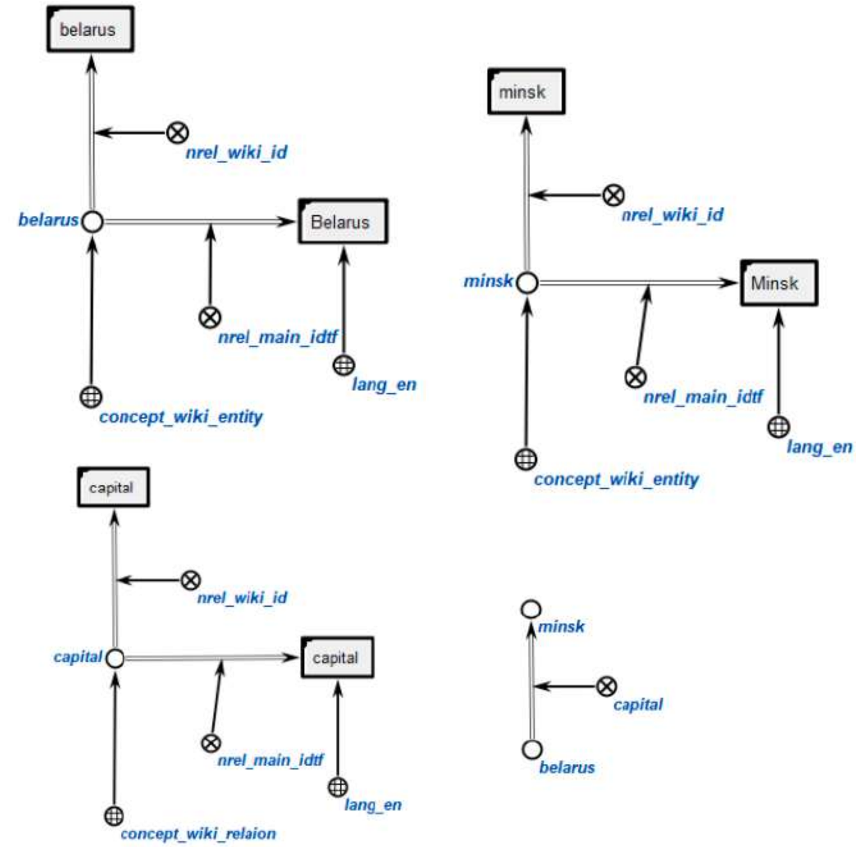


Figure 3. Generated wiki-entities and wiki-relations

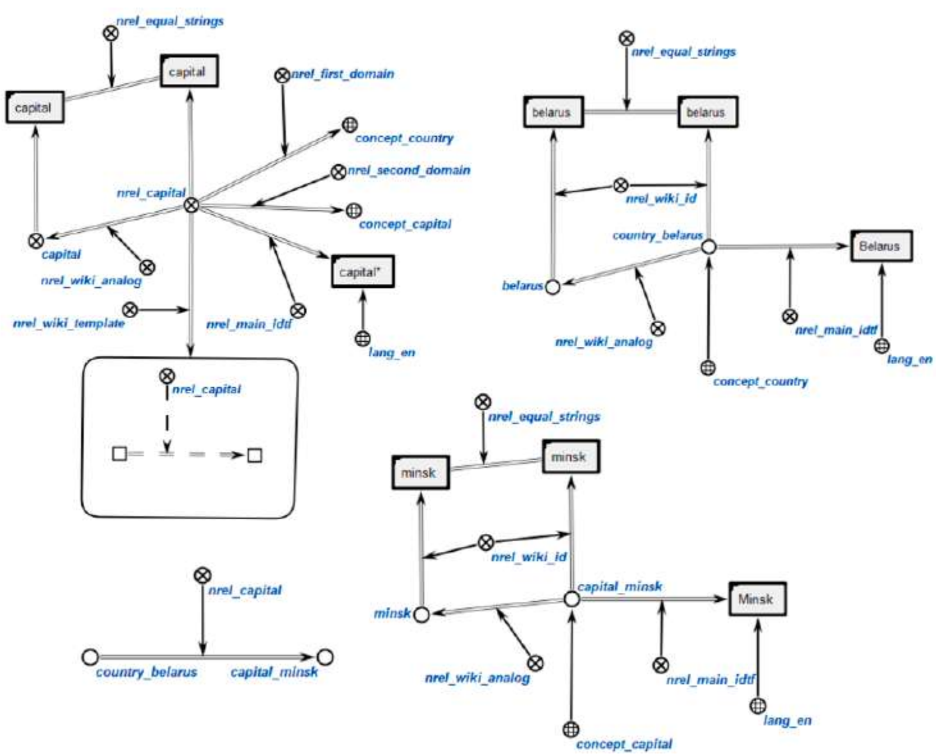


Figure 4. The result of the integration of knowledge

Finally, it is necessary to generate constructions from the block "triplets". Previously obtained wiki-entities and wiki-relations are used for them. The resulting constructs are shown in fig. 3.

It is important to note that the generated constructions are temporary. Next, they should be integrated with the permanent constructions of the ostis-system knowledge base. In general, there can be two main situations for each generated wiki-relation or wiki-entity: in the knowledge base either there are analogs for them or not. In the previously represented fragment of the *Subject domain of WikiData*, the relations "wiki-identifier*" and "wiki-analog*" were specified. The relation "wiki-identifier*" connects the generated entity (or a relation) with the identifier that was used in WikiData. The relation "wiki-analog*" connects an entity (or a relation) that exists in the knowledge base with the generated wiki-entity (or a wiki-relation). Obviously, there may be a situation when there is an analog in the knowledge base for a wiki-entity (or a wiki-relation), but they are not connected by the relation "wiki-analog*". Thus, the following scenarios appear:

- there is an analog for a wiki-entity (or a wiki-relation), and they are connected by the relation "wiki-analog*";
- there is an analog for a wiki-entity (or a wiki-relation), but they are not connected by the relation "wiki-analog*";
- there is no analog for a wiki-entity (or a wiki-relation).

Next, each of the three situations is considered by examples.

It is necessary to check whether the wiki-relation "capital" is connected with some other relation by the binding "wiki-analog*". Such a relation was found. This is "nrel_capital".

A similar check is carried out for the wiki-entity "belarus". The entity connected to it by the relation "wiki-analog*" was not found, so the wiki-identifier is used for the search. Thanks to the identifier, the entity "country_belarus" was found.

The wiki-entity "minsk" is considered lastly. When searching through the relation "wiki-analog*" or the wiki-identifier, no matches were found. Therefore, the entity "capital_minsk" was generated.

After all, it is necessary to generate arcs between the found analogs. The result is a binding of the non-role relation "nrel_capital", which connects "country_belarus" and "capital_minsk".

The result of the integration described above is shown in figure 4.

VI. Conclusion

The mechanism proposed in the article ensures the integration of knowledge from external sources into the knowledge base of an intelligent system built on the basis of the OSTIS Technology.

The essence of the approach is the ontological description of the syntax and semantics of the external language, the construction of rules for the transition of knowledge from a representation in an external language to a representation in an internal language and the subsequent integration of the acquired knowledge with existing knowledge in the knowledge base.

The approach was applied to integrate WikiData with the ostis-system and is considered in the article.

The proposed mechanism allows solving the problem of synonymy of source data, unifying the format of information representation and monitoring the consistency and quality of integrated knowledge.

References

[1] Golenkov, V. V. A Project of an Open Semantic Technology for the Component Design of Intelligent Systems. Part 1: Principles of Creation. Ontologiya proyektirovaniya [Ontology of Design], 2014, № 11, pp. 42–64.

[2] Data Types: Structured vs. Unstructured Data. Available at: https://www.bigdataframework.org/data-types-structured-vs-unstructured-data/(accessed 2021, June)

[3] Datamation. Structured vs. Unstructured Data. Available at: https://www.datamation.com/big-data/structured-vs-unstructured-data/ (accessed 2021, June)

[4] Amazon AWS: What is a data lake?. Available at: https://aws.amazon.com/ru/big-data/datalakes-and-analytics/what-is-a-data-lake/ (accessed 2021, June)

[5] H. Dibowski, S. Schmid. Using Knowledge Graphs to Manage a Data Lake. Lecture Notes in Informatics (LNI), Gesellschaft für Informatik. Bonn, 2021, pp. 41-50.

[6] RDF 1.1 Concepts and Abstract Syntax. Available at: https://www.w3.org/TR/rdf11-concepts/ (accessed 2021, June)

[7] R2RML: RDB to RDF Mapping Language. Available at: https://www.w3.org/TR/r2rml/ (accessed 2021, June)

[8] RDF Mapping Language (RML). Available at: https://rml.io/specs/rml/ (accessed 2021, June)

[9] Vit Novacek, Loredana Laera, Siegfried Handschuh, Brian Davis. Infrastructure for Dynamic Knowledge Integration – Automated Biomedical Ontology Extension Using Textual Resources. Journal of Biomedical Informatics, 41(4), 2008.

[10] Cambridge Dictionary. Available at: https://dictionary.cambridge.org/ (accessed 2021, June)

[11] D.Hernández, A.Hogan, M.KrötzSCh. Reifying RDF: What Works Well With Wikidata? Proceedings of the 11th International Workshop on SCalable Semantic Web KB Systems, 2015, pp. 32–47.

[12] IMS.OSTIS Metasystem. Available at: http://ims.ostis.net/. (accessed 2021, June)

Онтологический подход к интеграции знаний из внешних источников

Коршунов Р. А., Садовский М. Е., Загорский А. Г.

В рамках данной статьи предлагается автоматизированный механизм интеграции знаний из внешних источников.

Суть подхода заключается в онтологическом описании синтаксиса и семантики внешнего языка, построении правил перехода знаний с представления на внешнем языке на представление на внутреннем языке и последующей интеграции полученных знаний с уже существующими знаниями в базе знаний.

Реализация подхода рассмотрена на примере интеграции WikiData с ostis-системой.

Внедрение данного механизма позволит решить ряд задач: сокращение временных и трудовых затрат на пополнение базы знаний, использование различных внешних источников, автоматизация процессов интеграции знаний.

С точки зрения качества знаний реализация предлагаемого подхода позволит решить проблему синонимии исходных данных, унифицировать формат представления информации, следить за целостностью и качеством интегрируемых знаний.

Received 01.06.2021

Processing and understanding of the natural language by an intelligent system

Hardzei A.
Minsk State
Linguistic University
Minsk, Belarus
Email: alieks2001@yahoo.com

Svyatoshchik M.
Minsk State
Linguistic University
Minsk, Belarus
Email: svyatoshchikm@mail.ru

Bobyor L.
Intelligent Semantic Systems Ltd.
Minsk, Belarus
Email: lizabobyor@gmail.com

Nikiforov S.
Belarusian State University
of Informatics and Radioelectronics
Minsk, Belarus
Email: nikiforov.sergei.al@gmail.com

***Abstract*—The article is dedicated to Natural Language Processing in the Theory for Automatic Generation of Knowledge Architecture (TAPAZ-2) paradigm and the immersion of the obtained semantic formalisms into the software environment through the Open Semantic Technology for Intelligent Systems (OSTIS). A specific feature of the approach is the formalization of natural language semantics based on the World Model and the combination of Semantic Coding with the ontology and taxonomy of semantic networks.**

***Keywords*—Natural Language Processing (NLP), Natural Language Understanding (NLU), macroprocess, roles of individs, Parts of Language, Parts of the Sentence, taigens, yogens, Combinatory Semantics, Theory for Automatic Generation of Knowledge Architecture (TAPAZ-2), ontology, taxonomy, semantic network, Open Semantic Technology for Intelligent Systems (OSTIS), SC-code (Semantic Computer Code)**

I. Introduction

The present research is carried out in the framework of Combinatory Semantics, which studies the linguistic mapping of the dynamics of individs' roles in an event. [1, p. 13]. Natural language is understood as a system of figures and signs for decoding the World Model and conscious management of intellectual activity [2, p. 35]. The World Model (hidden knowledge) is the architecture of patterns, i.e. the ordered set of patterns and the ordered set of transformations of some patterns in others [3, p. 226], [46, p. 182]. It is necessary to distinguish between verbal and non-verbal knowledge. Non-verbal knowledge is beyond the rational approach for comprehension of the World, it is beyond any term system, whether it is mathematics, computer science, linguistics, paralinguistics and semiotics as a whole. It is impossible to explain and show, how the imageries of Raphael's paintings came up, therefore, non-verbal knowledge should not be confused with facial expressions and gestures. Sign or finger language (language of the deaf and dumb) is as a natural language as any other hieroglyphic language [4, p. 18]. Verbal knowledge consists of information and fascination. According to Yu. V. Knorozov, the maximal information is contained in mathematics and the maximal fascination is contained in music [5]. Further studies have shown that fascination, along with factuation or factualization, is still a kind of information [6].

In this case, fascination that involves any stylistic nuance, all kinds of emotional and expressive shades and "induced" emotions, including those created with the help of meter, rhythm, pause, chant, representational devices and other accentological means, are equally covered by declarative and procedural methods of representing knowledge. Due to the fact that language categories as supporting constructs of the metatheory of any natural language are linked to verbal knowledge, only those that are distinguished procedurally, fixed declaratively and confirmed combinatorially can be determined as relevant [7]. Meanwhile, "The "chunks" of reasoning, language, memory, and "perception" ought to be larger and more structured; their factual and procedural contents must be more intimately connected in order to explain the apparent power and speed of mental activities" [8, p. 1].

The purpose of the article is to demonstrate the possibility of understanding texts in a natural language by computer systems with semantic software that allows creating a problem solver architecture based on a combination of Semantic Coding with the ontology and taxonomy of semantic networks.

To write formalized texts, in the article, the variants of the external displaying of SC-code constructs – SCg (graphic version) and SCn (hypertext version) – are used.

II. Problem definition

The discrepancy of the World Model and the Linguistic Image of the World, indefinite meaning of lexical units and syntactic incompleteness of sentences are the main stumbling blocks in Natural Language Processing (NLP) [9]. Attempts to remove or circumvent these problems with the help of statistics based on co-occurrence by Z. S. Harris [10] resemble guessing the correct answers by schoolchildren during centralized testing. No matter how sophisticated the methods of statistical processing of structured or unstructured natural language content are, they only imitate the intellectual or inventive activity of a human, guessing the correct answer with more or less certainty, but we do not doubt that neural networks are able to efficiently scale the solutions found by combinatory methods. As for the currently popular combinatory methods, they go back to the semantic cases of Ch. Fillmore [11]–[22] and Stowell's "theta-grids" [23] and are used, in particular, in the Semantic Web project of T. Berners-Lee [24]–[27] and on an international community resource Schema.org of the Google, Microsoft, Yahoo and Yandex developer communities [28]. The main disadvantage of these methods is their empirical character and the lack of unified algebraic bases of semantic calculus. Because of these reasons, the creators of the Semantic Web, despite titanic efforts to standardize the technology, have not yet managed to reduce various subject ontologies to a top-level ontology, which, as many commentators emphasize, is "critical for the entire concept" [29, p. 94]. The fact is that the top-level ontology cannot be built from below, it, so to speak, "does not grow" from the ontologies of subject domains but must be initially set from above and in an algebraic standard that is suitable for formalizing texts in natural language including sentences, free and stable strings of combinatory variants of lexical items and lexical items themselves that make up these texts. In other words, to embed patterns of the World Model in presentation formalisms, a formalized language is required, which is comparable in semantic power to a natural language, as V. V. Martynov pointed out at the time [30]. Otherwise, as a result, we will get, as D. G. Bobrow and T. Winograd wrote, an extremely fragile structure, often resembles a house of cards and collapses immediately "if swayed in the slightest from the specific domain (often even the specific examples) for which it was built" [31, p. 4].

Let us pay attention one more time: semantics as the relation of language to the World Model is manifested in the dynamics of individs' roles in an event, which is reflected in the content of patterns, the meaning of signs and the sense of sentences [1]–[4]. It is possible to arbitrarily declare any top-level object-oriented programming language, such as Java, C++, C# or the next version of the OWL language from the Semantic Web project, as a top-level ontology, but until such languages can encode the content of patterns, the meaning of signs and the sense of sentences and then reduce them to semantic primitives underlying calculus [32], such statements will be only declarations. If the OWL language allowed encoding patterns of the World Model and conjugate code with natural language semantics, then the Internet would already be turned into the Global Artificial Intelligence through the Semantic Web project. It should be noted that linguistics has only one synthetic (consistently deductive and procedural) language model – Panini grammar that is dated from the 5th century BC, in which, with the help of 3959 short rules (sutras), the generation, construction and transformation of all Sanskrit units are exhaustively described, starting from the phonetic-phonological level and ending with the semantic-syntactic one [33]–[35]. Unfortunately, it has not yet been clarified what formalisms are the basis for such an accurate description of natural language and how it was possible to achieve this in such a long time. From modern methods of encoding language semantics, six versions of Universal Semantic Code (USC) of V. V. Martynov [30], [36]–[40] and their finalization in the Theory for Automatic Generation of Knowledge Architecture (TAPAZ-2) by A. N. Hardzei [4], [41]–[45] are known.

TAPAZ Semantic Classifier as a top-level ontology includes the Set of Macroprocesses as Semantic Primitives (Paradigm of Actions) ordered by TAPAZ-algebra, Role List of Individs and TAPAZ Knowledge Graph [46].

Taking into account that the calculus of subject domains and the semantics of each subject domain is implemented in TAPAZ-2 separately using a specially oriented knowledge graph, the most effective means of immersing TAPAZ formalisms in the software environment are dynamic graph models, primarily an SC-code (Semantic Computer Code) of the Open Semantic Technology for Intelligent Systems (OSTIS) developed by the school of V. V. Golenkov [47], [70], [71].

We suppose that combining efforts and an organic combination of semantic coding with the ontology and taxonomy of semantic networks will solve a number of central problems of automatic data processing in natural language (Natural Language Processing), shifting the emphasis towards machine understanding of natural language (Natural Language Understanding, NLU).

III. Proposed approach

According to T. N. Domnina and O. A. Khachko, at the beginning of 2014, the number of scientific peer-reviewed journals was 34,274. If the average amount of articles is at least 50 per year, then 1,713,700 articles are published per year [48]. T. V. Chernigovskaya complains that "the number of articles related to the brain exceeds 10 million – they simply cannot be read. Every day about ten pieces get out" [49]. The average growth in the number of peer-reviewed scientific journals is 4% per year. In 2018, 1.6 million scientific articles were included in the Web of Science database [50]. So, it is essential to use automatic text analysis, artificial intelligent systems for searching and processing information.

In 1994, A. N. Hardzei, in a group led by V. V. Martynov, for the first time proposed a procedure for calculating semantics in the form of a specially oriented graph for ranking complex strings [41]. Use of the procedure has required the establishment of a one-to-one (vector) transition between actions in basic semantic classifier and has led to the creation of the Theory for Automatic Generation of Knowledge Architecture (TAPAZ) which was founded on: the formal theory; the semantic counterpart; the set of macroprocesses (actions) as semantic primitives; the algorithm defining roles of individs, and the graph for searching processes through macroprocesses (knowledge graph) [44, p. 11]

In 2014, the second version of TAPAZ appeared, in which greatly simplified algebraic apparatus, increased number of rules of interpretation of the standard superposition of individs [43].

At the same time, the problems of unifying the principles of building various components of computer systems were solved within the framework of the OSTIS project [51] aimed at creating an open semantic technology for engineering knowledge-driven systems. This technology allows combining heterogeneous models of problem solving as a universal platform and reducing costs that arise during development and modification, including when adding new components to the system. The OSTIS Technology makes it possible to use both statistical and combinatory methods that operate with knowledge. It is based on a unified version of information encoding based on semantic networks with a basic set-theoretic interpretation called SC-code. The architecture of each system built using the OSTIS Technology (ostis-system) includes a platform for interpreting semantic models of ostis-systems as well as a semantic model of the ostis-system described using the SC-code (sc-model of the ostis-system). In turn, the sc-model of the ostis-system includes the sc-model of the knowledge base, the sc-model of the problem solver and the sc-model of the interface (in particular, the user one).

The basis of the knowledge base of any ostis-system (sc-model of the knowledge base) is a hierarchical system of subject domains and corresponding ontologies. The upper level of the hierarchy of the knowledge base fragment related directly to natural language processing is shown below.

Knowledge base on natural language processing
⇐ *section decomposition**:
{
- *Section. Subject domain of lexical analysis*
- *Section. Subject domain of syntactic analysis*
- *Section. Subject domain of semantic analysis*
- *Section. Subject domain of TAPAZ-2*

}

The problem solver of any ostis-system (sc-model of the ostis-system problem solver) is a hierarchical system of agents of knowledge processing in semantic memory (sc-agents) that interact with each other solely by specifying the acts they perform in the specified memory.

Problem solver for natural language processing
⇐ *decomposition of an abstract sc-agent**:
{
- *Agent of lexical analysis*
- *Agent of syntactic analysis*
- *Agent of semantic analysis*
- *Agent of merging structures in the knowledge base*
- *Agent of logical inference*

}

Agent of merging structures in the knowledge base
⇐ *decomposition of an abstract sc-agent**:
{
- *Agent of searching for contradictions*
- *Agent of resolving contradictions*

}

The agent of lexical analysis decomposes the text into lexemes and nominative units (stable strings of combinatory variants of lexemes) based on the dictionary included in the subject domain of lexical analysis. Note that the lexicographic description also presupposes the establishment of the linguistic semantic category for the lexeme, i.e. its belonging to a certain Part of Language [52]–[56]. The agent of syntactic analysis builds the syntactic structure of the analyzed text using the specified rules. The agent of semantic analysis performs the transition from the text specification created by the previous agents to the structure that describes its semantics. The agent of merging structures in the knowledge base compares the structures obtained as a result of the text analysis with the data available in the knowledge base and, if contradictions are detected, resolves them.

The agent of logical inference uses logical rules written by means of the SC-code and interacts with the agents of syntactic and semantic analysis.

A more detailed explanation of the abovementioned subject domains and agents of the proposed approach is given on the example of processing of a particular fragment of natural-language text, namely the description of the technological process of production of cottage cheese: «Производство творога кислотным способом включает в себя: приёмку молока, нормализацию молока до жирности 15%, очистку и пастеризацию молока, охлаждение молока до температуры заквашивания, внесение закваски в молоко, сквашивание молока, разрезку сгустка, подогрев и обработку сгустка, отделение сыворотки, охлаждение творога»[1].

[1] “Manufacture of cottage cheese by the acid method includes: acceptance of milk, normalization of milk to 15% fat, purification and pasteurization of milk, cooling of milk to the fermentation temperature, adding sourdough to milk, fermentation of milk, cutting of the clot, heating and processing of the clot, separation of whey, cooling of cottage cheese”.

From the point of view of the ostis-system, any text is a file (i.e., an sc-node with content). An example of such a node is shown in Figure 1.

Figure 1. Representation of natural language text in the system.

Let us consider each of the stages of processing this text.

IV. Lexical analysis

It is a decomposition of a text by an agent of lexical analysis into separate lexemes and stable strings of combinatory variants of lexemes (nominative units) based on a dictionary that is part of the subject domain of lexical analysis. Below is a fragment of the ontology that contains knowledge about Parts of Language.

lexeme
⊂ *file*

nominative unit
⊂ *file*

combinatory variant of the lexeme
⊂ *file*

The *lexeme* is a taigen or yogen of a particular natural language [2, p. 35]. A *combinatory variant of lexeme* is a variant of a lexeme in an ordered set of its variants (paradigm) [57, p. 351].

A *nominative unit* is a stable string of combinatory variants of lexemes, in which one variant of the lexeme (modificator) defines another one (actualizator), for example: 'записная книжка' = 'note book', 'бежать галопом' = 'run at a gallop' [2, p. 35].

morphological paradigm*
∈ *quasi-binary relation*
⇒ *first domain**:
 word form
⇒ *second domain**:
 lexeme

natural language
⇒ *decomposition**:
 {• *Part of Language*
  ⇒ *decomposition**:
   {• *taigen*
    • *yogen*
   }
 • *sign of syntax alphabet*
 }

The *morphological paradigm** is a quasi-binary relation, connecting a lexeme with its combinatory variants. The *lexeme* – taigen or yogen of a particular natural language, being a sign, it has a combination of figures in the aspect of expression, and it has a pattern in the aspect of content; in synthetic languages it has a developed morphological paradigm and is the central unit of lexicographic description.

*Signs of Syntax Alphabet** are auxiliary syntactic means (at the macrolevel – prepositions, postpositions, conjunctions, particles, etc., at the microlevel – flexions, prefixes, postfixes, infixes, etc.) that serve for connecting the components of language structures and the formation of morphological paradigms [2, p. 35].

A *nominative unit* is a stable string of combinatorial variants of lexemes, in which one variant of a lexeme (modificator) defines another (actualizator).

taigen
⇒ *decomposition**:
 {• *expanded taigen*
  ⇒ *decomposition**:
   {• *composite taigen*
    • *complex taigen*
   }
 • *reduced taigen*
  ⇒ *decomposition**:
   {• *contracted taigen*
    • *constricted taigen*
   }
 }

constricted taigen
⇒ *decomposition**:
 {• *informational taigen*
 • *physical taigen*
  ⇒ *decomposition**:
   {• *constant taigen*
    • *variable taigen*
   }
  ⇒ *decomposition**:
   {• *quantitative taigen*
    • *qualitative taigen*
   }
  ⇒ *decomposition**:
   {• *single-place taigen*
    • *multi-place taigen*
     ⊃ *intensive taigen*
     ⊃ *extensive taigen*
   }
 }

A *taigen* is a Part of Language that denotes an individ.

An *informational taigen* denotes an individ in the informational fragment of the World Model, a *physical taigen* denotes an individ in the physical fragment of the World Model.

A *constant taigen* denotes a constant individ, a *variable taigen* denotes a variable individ [54, pp. 70–72], [58, pp. 12–13].

yogen

⇒ *decomposition**:
- { • *expanded yogen*
  - ⇒ *decomposition**:
    - { • *composite yogen*
    - • *complex yogen*
    - }
- • *reduced yogen*
  - ⇒ *decomposition**:
    - { • *contracted yogen*
    - • *constricted yogen*
    - }
- }

contracted yogen

⇒ *decomposition**:
- { • *informational yogen*
- • *physical yogen*
  - ⇒ *decomposition**:
    - { • *constant yogen*
    - • *variable yogen*
    - }
  - ⇒ *decomposition**:
    - { • *quantitative yogen*
    - • *qualitative yogen*
    - }
  - ⇒ *decomposition**:
    - { • *single-place yogen*
    - • *multi-place yogen*
    - }
- }

multi-place yogen

⇒ *decomposition**:
- { • *intensive yogen*
- • *extensive yogen*
- }

A *yogen* is a Part of Language that denotes an attribute of an individ.

An *informational yogen* denotes the attribute of an individ in the informational fragment of the World Model, a *physical yogen* denotes the attribute of an individ in the physical fragment of the World Model.

A *constant yogen* denotes a constant attribute of an individ, a *variable yogen* denotes a variable attribute of an individ [54, pp. 71–74], [58, p. 12–13].

The lexemes in the knowledge base are described in the form shown in Figure 2.

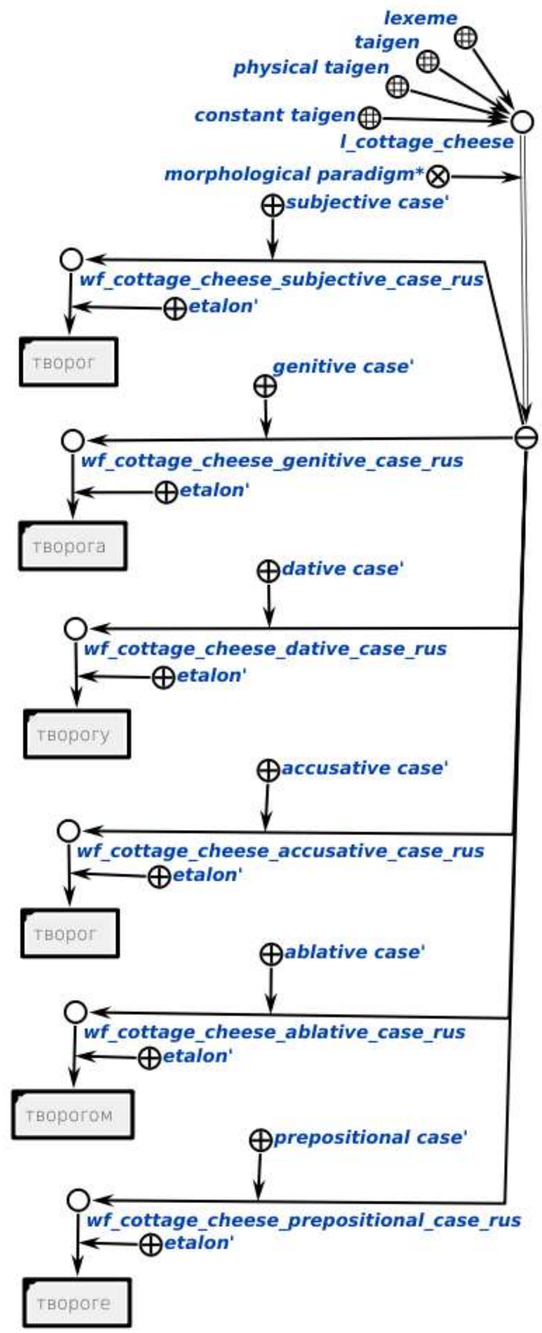


Figure 2. The description of the lexeme in the knowledge base.

The construct, which is the result of lexical analysis, is shown in Figure 3.

V. Syntactic analysis

The agent of the syntactic analysis performs the transition from the lexically marked text to its syntactic structure based on the rules described in the correspond-

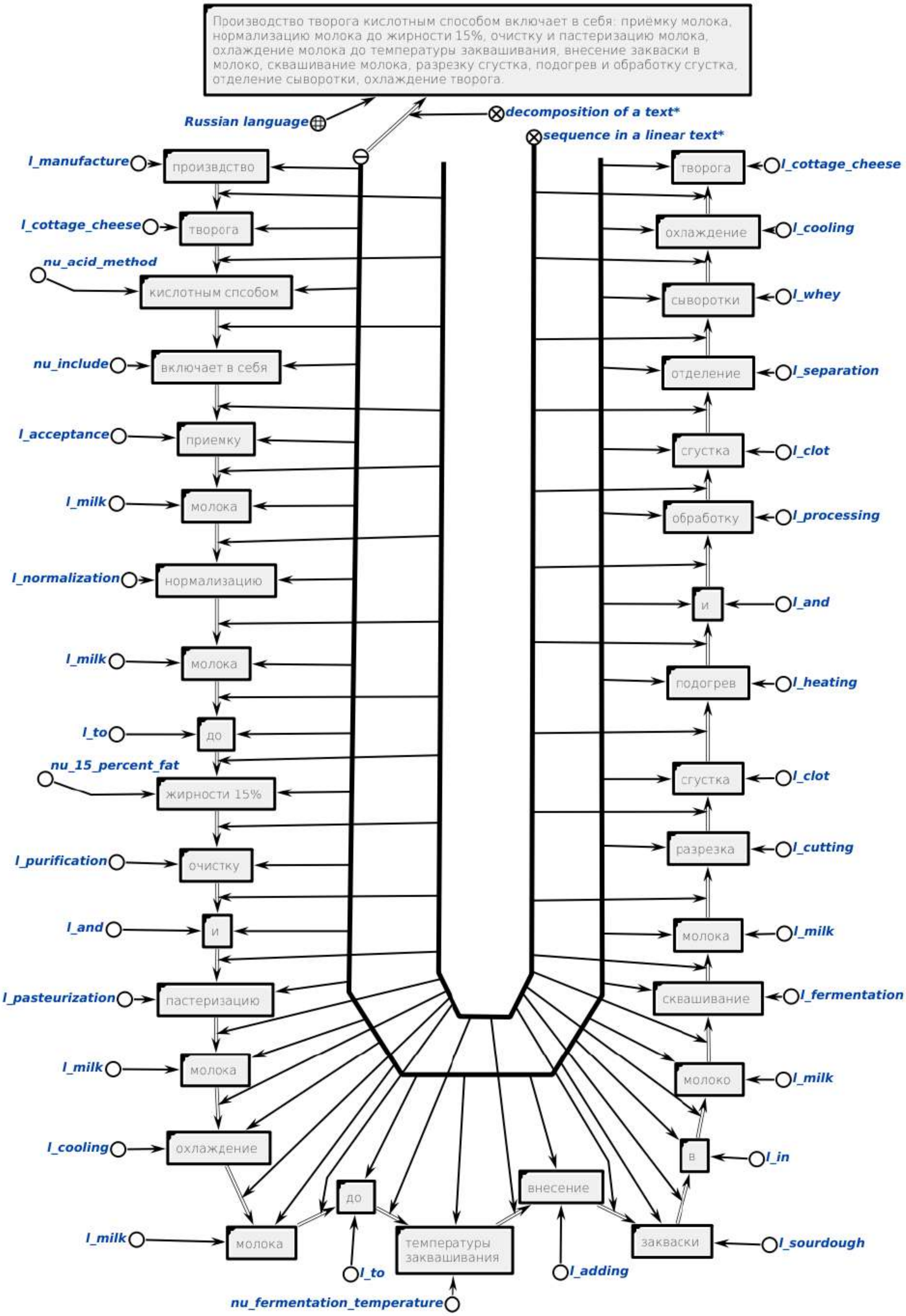


Figure 3. The result of lexical text analysis.

ing subject domain. A fragment of the ontology of the subject domain is presented below:

part of the sentence'
∈ *role relation*
⇒ *decomposition**:
{• *principal part of the sentence'*
⇒ *decomposition**:
{• *grammatical subject'*
• *grammatical predicate'*
• *grammatical direct object'*
}
• *subordinate part of the sentence'*
⇒ *decomposition**:
{• *grammatical indirect object'*
• *grammatical attribute'*
• *grammatical circumstance'*
}
}

A *Part of the Sentence'* is a relation that connects the decomposition of a text with a file whose contents (Part of Language) play a certain syntactic role in the decomposed text [2, p. 35].

The *grammatical subject'* is one of the principal role relations that connects the decomposition of a text with the file, the contents of which denotes the starting point of the event description selected by the observer.

The *grammatical direct object'* is one of the principal role relations that connects the decomposition of a text with the file, the contents of which denotes the final point of the event description selected by the observer.

The *grammatical predicate'* is one of the principal role relations that connects the decomposition of the text with the file, the contents of which denotes the mapping by the observer of the starting point of the event description to the final point [46, p. 184].

A *grammatical circumstance'* is one of the subordinate role relations that connects the decomposition of a text with a file, the contents of which denote either a modification or localization of the grammatical predicate; the grammatical circumstance of degree and the grammatical circumstance of manner denote the modification of the grammatical predicate, the grammatical circumstance of place and the grammatical circumstance of time denote the spatial and, accordingly, the temporal localization of the grammatical predicate.

A *grammatical attribute'* is one of the subordinate role relations that connects the decomposition of a text with the file, the contents of which denote a modification of the grammatical subject, grammatical object, grammatical circumstance of place and time [57, pp. 352–354, 357], [59, pp. 29–33], [60].

grammatical circumstance'
⇒ *decomposition**:
{• *grammatical circumstance of degree'*
• *grammatical circumstance of manner'*
• *grammatical circumstance of place'*
⇒ *decomposition**:
{• *dynamical grammatical circumstance of place'*
• *static grammatical circumstance of place'*
}
• *grammatical circumstance of time'*
⇒ *decomposition**:
{• *dynamical grammatical circumstance of time'*
• *static grammatical circumstance of time'*
}
}

A fragment of the ontology that is the result of this stage is presented in Figures 4 and 5.

VI. Construction in terms of TAPAZ-2

The agent of semantic analysis performs the transition from the processed text to its semantics formulated in terms of TAPAZ-2 on the basis of the rules described in the corresponding subject domain [4], [42]–[46], [61]. A fragment of the ontology of this subject domain is presented below:

participant of the exposure*
:= [participant of the event*]
∈ *non-role relation*
⇒ *first domain**:
individ
⇒ *second domain**:
action
⇒ *decomposition**:
{• *subject**
⇒ *decomposition**:
{• *initiator**
• *inspirer**
• *spreader**
• *creator**
}
• *instrument**
⇒ *decomposition**:
{• *activator**
• *suppressor**
• *enhancer**
• *converter**
}
• *mediator**
⇒ *decomposition**:
{• *landmark**
• *locus**
• *carrier**
• *adapter**
• *acceptor**

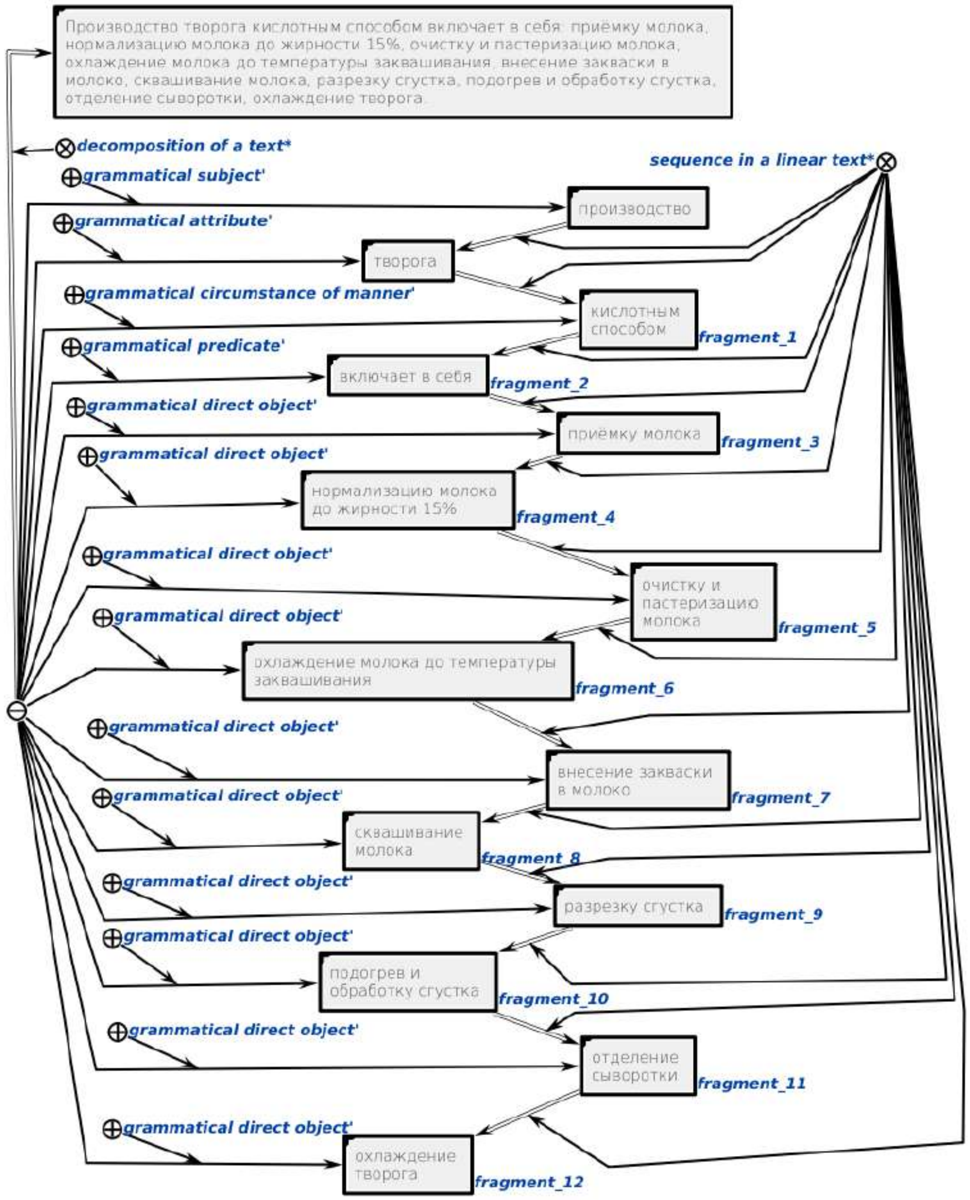


Figure 4. The result of the syntactic analysis of the text, the first fragment.

- *stock**
- *separator**
- *material**
- *model**
- *retainer**
- *resource**
- *stimulus**
- *regulator**
- *chronotope**
- *source**
- *indicator**

}

- *object**

⇒ *decomposition**:

{
- *coating**
- *hull**
- *interlayer**
- *kernel**

}

- *product**

⇒ *decomposition**:

{
- *billet**
- *semi-product**
- *prototype**
- *end item**

}

}

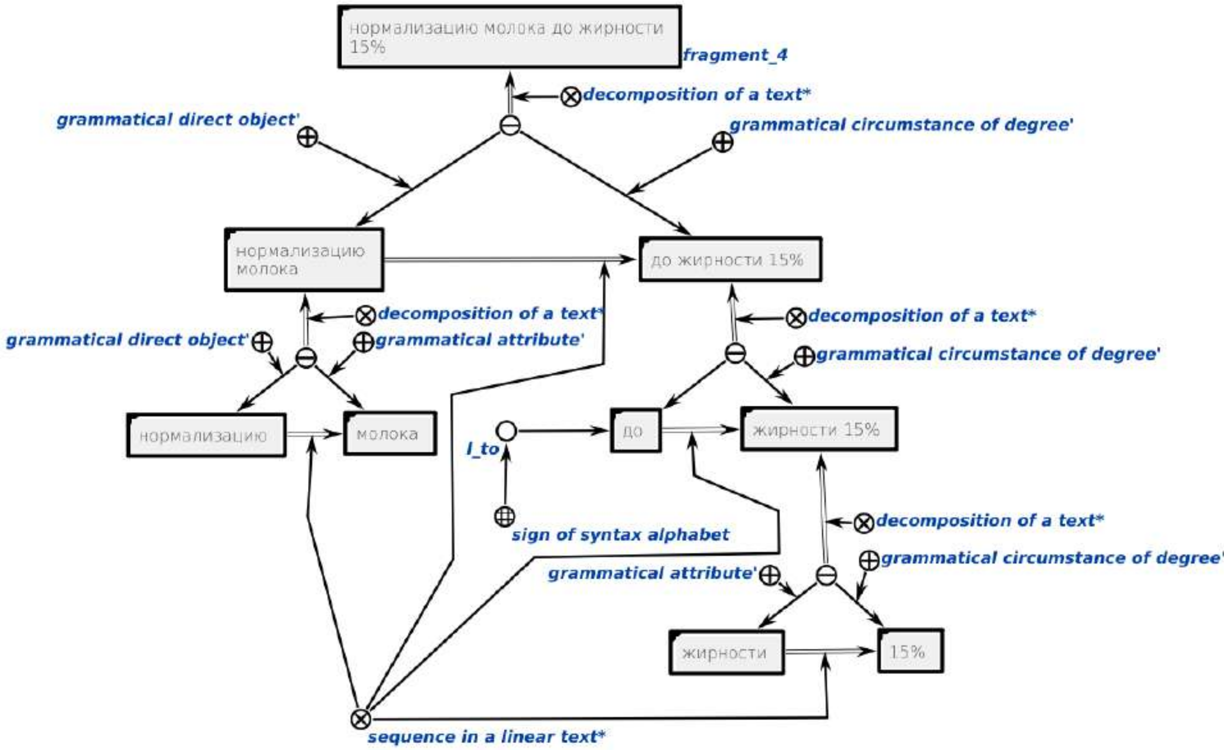


Figure 5. The result of the syntactic analysis of the text, the second fragment.

The *individ* is a kind of the pattern as a separate entity in the selected fragment of the World Model [2, p. 34].

The *participant of the action** is a non-role relation that connects the action with the individ that participates in it.

The *subject** – the originator of the action, varieties of the subject: *initiator** – initiates the action, *spreader** – spreads the action, *inspirer** – involves into the action, *creator** – completes the action by making a product from the object.

*Instrument** – the performer of the action, the closest individ to the subject, varieties of the instrument: *activator** – directly affects the mediator, *suppressor** – suppresses the resistance of the mediator, *enhancer** – increases the effect on the mediator, *converter** – converts the mediator into the instrument.

*Mediator**, i.e. the mediator of the action – the closest individ to the object; varieties of the mediator: *landmark** – orientates the impact on the object, *locus** – the closest environs of the object partially or completely surrounding the object that localizes the object in space and thereby containing (enclosing) it, *carrier** – carries the object, *adapter** – adapts the instrument to affect the object, *acceptor** – catches the object, *stock** – the object collected for processing, *separator** – sorts the object, *material** – the object used as a raw material for making a product, *model** – the physical or informational original sample for making a product from the object, *retainer** – turns a variable locus of the object into a constant one, *resource** – feeds the instrument, *stimulus** – reveals the parameter of the object, *regulator** – serves as an instruction in making a product from the object, *chronotope** – localizes the object in time, *source** – provides instructions for the instrument, *indicator** – displays a parameter of impact on the object or a parameter of the product as the result of subject's impact on the object.

*Object** – the recipient of the action, varieties of the object: *coating** – the outer insulation of the individ's shell, *hull** – the individ's shell, *interlayer** – the inner insulation of the individ's shell, *kernel** – the core of the individ.

*Product** – the result of the subject's impact (action) on the object (the individ adapted to a given role in a new action), varieties of the product: *billet** – the object turned into a raw material, *semi-product** – the product half-made from raw materials, *prototype** – the prototype product, *end item** – the finished product [61, p. 10, 15-16].

TAPAZ distinguishes between physical and informational processes, since on the highest abstract semantic level the physical action was considered as an influence of one individ onto another through its shell, and the informational action – through its surroundings [44, p. 12], [63], [64]. Below is the classification of semantic elements (macroprocesses) of TAPAZ written by means of the SC-code [44, p. 34], [46, p. 185].

exposure

:= [action]

:= [event]

⊂ *act*

⇒ *explanation**:

[The action is an influence of one individ onto another [44, p. 6].]

⇒ *decomposition**:

{• *activation exposure*

⇒ *decomposition**:

{• *m_perceive*
• *m_reflect*
• *m_comprehend*
• *m_understand*
• *m_attract*
• *m_cumulate*
• *m_constrict*
• *m_attain*
• *m_adopt*
• *m_memorize*
• *m_contemplate*
• *m_learn*
• *m_absorb*
• *m_accumulate*
• *m_center*
• *m_assimilate*
• *m_feel*
• *m_behold*
• *m_feel profoundly*
• *m_experience*
• *m_over absorb*
• *m_concentrate*
• *m_centrifuge*
• *m_dissimilate*
• *m_reject*
• *m_erase*
• *m_rethink*
• *m_overcome*
• *m_expel*
• *m_decompress*
• *m_force off*
• *m_disassociate*

}

• *exploitation exposure*

⇒ *decomposition**:

{• *m_notify*
• *m_advertise*
• *m_instill*
• *m_state*
• *m_approach*
• *m_joint*
• *m_press down*
• *m_connect*
• *m_explain*
• *m_propagandize*
• *m_prove*
• *m_certify*
• *m_insert*
• *m_pump*
• *m_press in*
• *m_link*
• *m_reveal*
• *m_prophesize*
• *m_enlighten*
• *m_divine*
• *m_conduct*
• *m_spread*
• *m_squeeze out*
• *m_disconnect*
• *m_darken*
• *m_encode*
• *m_discredit*
• *m_disavow*
• *m_take out*
• *m_pull up*
• *m_push out*
• *m_unlink*

}

• *transformation exposure*

⇒ *decomposition**:

{• *m_inform*
• *m_interest*
• *m_assure*
• *m_predispose*
• *m_touch on*
• *m_envelope*
• *m_clamp*
• *m_mold*
• *m_admonish*
• *m_teach*
• *m_convince*
• *m_nurture*
• *m_rip up*
• *m_fill up*
• *m_press*
• *m_form*
• *m_pierce*
• *m_intend*
• *m_transfigure*
• *m_reincarnate*
• *m_penetrate*
• *m_overflow*
• *m_unclamp*
• *m_eviscerate*
• *m_pester*
• *m_mesmerize*
• *m_lose conscious*
• *m_go mad*
• *m_punch*

- *m_uplift*
- *m_disband*
- *m_annihilate*

}

- *normalization exposure*

⇒ *decomposition**:

{
- *m_recollect*
- *m_recreate*
- *m_restart*
- *m_render*
- *m_recrystallize*
- *m_reintegrate*
- *m_regenerate*
- *m_restore*
- *m_reproduce*
- *m_reclaim*
- *m_renew*
- *m_revive*
- *m_recuperate*
- *m_rehabilitate*
- *m_reactivate*
- *m_reanimate*

}

}

⇒ *decomposition**:

{
- *surroundings-shell exposure*

⇒ *decomposition**:

{
- *m_perceive*
- *m_reflect*
- *m_comprehend*
- *m_understand*
- *m_attract*
- *m_cumulate*
- *m_constrict*
- *m_attain*
- *m_notify*
- *m_advertise*
- *m_instill*
- *m_state*
- *m_approach*
- *m_joint*
- *m_press down*
- *m_connect*
- *m_inform*
- *m_interest*
- *m_assure*
- *m_predispose*
- *m_touch on*
- *m_envelope*
- *m_clamp*
- *m_mold*
- *m_recollect*
- *m_recreate*
- *m_restart*
- *m_render*
- *m_recrystallize*
- *m_reintegrate*
- *m_regenerate*
- *m_restore*

}

- *shell-core exposure*

⇒ *decomposition**:

{
- *m_adopt*
- *m_memorize*
- *m_contemplate*
- *m_learn*
- *m_absorb*
- *m_accumulate*
- *m_center*
- *m_assimilate*
- *m_explain*
- *m_propagandize*
- *m_prove*
- *m_certify*
- *m_insert*
- *m_pump*
- *m_press in*
- *m_link*
- *m_admonish*
- *m_teach*
- *m_convince*
- *m_nurture*
- *m_rip up*
- *m_fill up*
- *m_press*
- *m_form*
- *m_reproduce*
- *m_reclaim*
- *m_renew*
- *m_revive*
- *m_recuperate*
- *m_rehabilitate*
- *m_reactivate*
- *m_reanimate*

}

- *core-shell exposure*

⇒ *decomposition**:

{
- *m_feel*
- *m_behold*
- *m_feel profoundly*
- *m_experience*
- *m_over absorb*
- *m_concentrate*
- *m_centrifuge*
- *m_dissimilate*
- *m_reveal*
- *m_prophesize*
- *m_enlighten*
- *m_divine*
- *m_conduct*

- *m_spread*
- *m_squeeze out*
- *m_disconnect*
- *m_pierce*
- *m_intend*
- *m_transfigure*
- *m_reincarnate*
- *m_penetrate*
- *m_overflow*
- *m_unclamp*
- *m_eviscerate*

}

- *shell-surroundings exposure*

⇒ *decomposition**:

{
- *m_reject*
- *m_erase*
- *m_rethink*
- *m_overcome*
- *m_expel*
- *m_decompress*
- *m_force off*
- *m_disassociate*
- *m_darken*
- *m_encode*
- *m_discredit*
- *m_disavow*
- *m_take out*
- *m_pull up*
- *m_push out*
- *m_unlink*
- *m_pester*
- *m_mesmerize*
- *m_lose conscious*
- *m_go mad*
- *m_punch*
- *m_uplift*
- *m_disband*
- *m_annihilate*

}

}

⇒ *decomposition**:

{
- *initiation exposure*

⇒ *decomposition**:

{
- *m_perceive*
- *m_attract*
- *m_adopt*
- *m_absorb*
- *m_feel*
- *m_over absorb*
- *m_reject*
- *m_expel*
- *m_notify*
- *m_approach*
- *m_explain*
- *m_insert*
- *m_reveal*
- *m_conduct*
- *m_darken*
- *m_take out*
- *m_inform*
- *m_touch on*
- *m_admonish*
- *m_rip up*
- *m_pierce*
- *m_penetrate*
- *m_pester*
- *m_punch*
- *m_recollect*
- *m_recrystallize*
- *m_reproduce*
- *m_recuperate*

}

- *accumulation exposure*

⇒ *decomposition**:

{
- *m_reflect*
- *m_cumulate*
- *m_memorize*
- *m_accumulate*
- *m_behold*
- *m_concentrate*
- *m_erase*
- *m_decompress*
- *m_advertise*
- *m_joint*
- *m_propagandize*
- *m_pump*
- *m_prophesize*
- *m_spread*
- *m_encode*
- *m_pull up*
- *m_interest*
- *m_envelope*
- *m_teach*
- *m_fill up*
- *m_intend*
- *m_overflow*
- *m_mesmerize*
- *m_uplift*
- *m_recreate*
- *m_reintegrate*
- *m_reclaim*
- *m_rehabilitate*

}

- *amplification exposure*

⇒ *decomposition**:

{
- *m_comprehend*
- *m_constrict*
- *m_contemplate*
- *m_center*
- *m_feel profoundly*

- *m_centrifuge*
- *m_rethink*
- *m_force off*
- *m_instill*
- *m_press down*
- *m_prove*
- *m_press in*
- *m_enlighten*
- *m_squeeze out*
- *m_discredit*
- *m_push out*
- *m_assure*
- *m_clamp*
- *m_convince*
- *m_press*
- *m_transfigure*
- *m_unclamp*
- *m_lose conscious*
- *m_disband*
- *m_restart*
- *m_regenerate*
- *m_renew*
- *m_reactivate*

}

- *generation exposure*
  ⇒ *decomposition**:
  { • *m_understand*
  - *m_attain*
  - *m_learn*
  - *m_assimilate*
  - *m_experience*
  - *m_dissimilate*
  - *m_overcome*
  - *m_disassociate*
  - *m_state*
  - *m_connect*
  - *m_certify*
  - *m_link*
  - *m_divine*
  - *m_disconnect*
  - *m_disavow*
  - *m_unlink*
  - *m_predispose*
  - *m_mold*
  - *m_nurture*
  - *m_form*
  - *m_reincarnate*
  - *m_eviscerate*
  - *m_go mad*
  - *m_annihilate*
  - *m_render*
  - *m_restore*
  - *m_revive*
  - *m_reanimate*

  }

}

⇒ *decomposition**:
{ • *physical action*
  ⇒ *explanation**:
  [The *physical action* is an influence in which the subject's shell acts as an instrument.]
  ⇒ *decomposition**:
  { • *m_attract*
  - *m_cumulate*
  - *m_constrict*
  - *m_attain*
  - *m_absorb*
  - *m_accumulate*
  - *m_center*
  - *m_assimilate*
  - *m_over absorb*
  - *m_concentrate*
  - *m_centrifuge*
  - *m_dissimilate*
  - *m_expel*
  - *m_decompress*
  - *m_force off*
  - *m_disassociate*
  - *m_approach*
  - *m_joint*
  - *m_press down*
  - *m_connect*
  - *m_insert*
  - *m_pump*
  - *m_press in*
  - *m_link*
  - *m_conduct*
  - *m_spread*
  - *m_squeeze out*
  - *m_disconnect*
  - *m_take out*
  - *m_pull up*
  - *m_push out*
  - *m_unlink*
  - *m_touch on*
  - *m_envelope*
  - *m_clamp*
  - *m_mold*
  - *m_rip up*
  - *m_fill up*
  - *m_press*
  - *m_form*
  - *m_penetrate*
  - *m_overflow*
  - *m_unclamp*
  - *m_eviscerate*
  - *m_punch*
  - *m_uplift*
  - *m_disband*

- *m_annihilate*
- *m_recrystallize*
- *m_reintegrate*
- *m_regenerate*
- *m_restore*
- *m_recuperate*
- *m_rehabilitate*
- *m_reactivate*
- *m_reanimate*

}

- *informational action*

⇒ *explanation**:

[The *informational action* is an influence in which the subject's surroundings acts as an instrument.]

⇒ *decomposition**:

{
- *m_perceive*
- *m_reflect*
- *m_comprehend*
- *m_understand*
- *m_adopt*
- *m_memorize*
- *m_contemplate*
- *m_learn*
- *m_feel*
- *m_behold*
- *m_feel profoundly*
- *m_experience*
- *m_reject*
- *m_erase*
- *m_rethink*
- *m_overcome*
- *m_notify*
- *m_advertise*
- *m_instill*
- *m_state*
- *m_explain*
- *m_propagandize*
- *m_prove*
- *m_certify*
- *m_reveal*
- *m_prophesize*
- *m_enlighten*
- *m_divine*
- *m_darken*
- *m_encode*
- *m_discredit*
- *m_disavow*
- *m_inform*
- *m_interest*
- *m_assure*
- *m_predispose*
- *m_admonish*
- *m_teach*
- *m_convince*
- *m_nurture*
- *m_pierce*
- *m_intend*
- *m_transfigure*
- *m_reincarnate*
- *m_pester*
- *m_mesmerize*
- *m_lose conscious*
- *m_go mad*
- *m_recollect*
- *m_recreate*
- *m_restart*
- *m_render*
- *m_reproduce*
- *m_reclaim*
- *m_renew*
- *m_revive*

}

}

The transition can be carried out in 2 stages:

- the transition from the initial version of the text to the reconstructed one;
- the transition from the reconstructed text to semantics.

The reconstruction of the text occurs through the reconstruction of the missing parts of the sentence based on the World Model or the Linguistic Image of the World and then through normalization of its syntactic structure by rewriting the Parts of the Sentence.

During the work, the following rules for the reconstruction of the text were formulated:

- the grammatical direct object of the initial text is displayed in the grammatical predicate of the reconstructed text (for example, the grammatical direct object *производство = a production* of the initial text is mapped to the grammatical predicate of the reconstructed text *производит = produces*);
- the grammatical attribute of the initial text is mapped to the grammatical direct object of the reconstructed text (for example, the grammatical attribute *творога = cottage cheese's* of the initial text is mapped to the grammatical direct object of the reconstructed text *творог = cottage cheese*).

The result of reconstruction of the initial text under consideration: «Некто принимает молоко, затем окисляет молоко, а именно: нормализует молоко до 15-процентной жирности, затем очищает молоко, затем пастеризует молоко, затем охлаждает молоко до определённой температуры, затем вносит закваску в молоко, затем сквашивает молоко, затем режет сгусток, затем подогревает сгусток, затем обрабатывает сгусток, затем отделяет сыворотку, затем охлаждает сгусток и, в итоге, производит

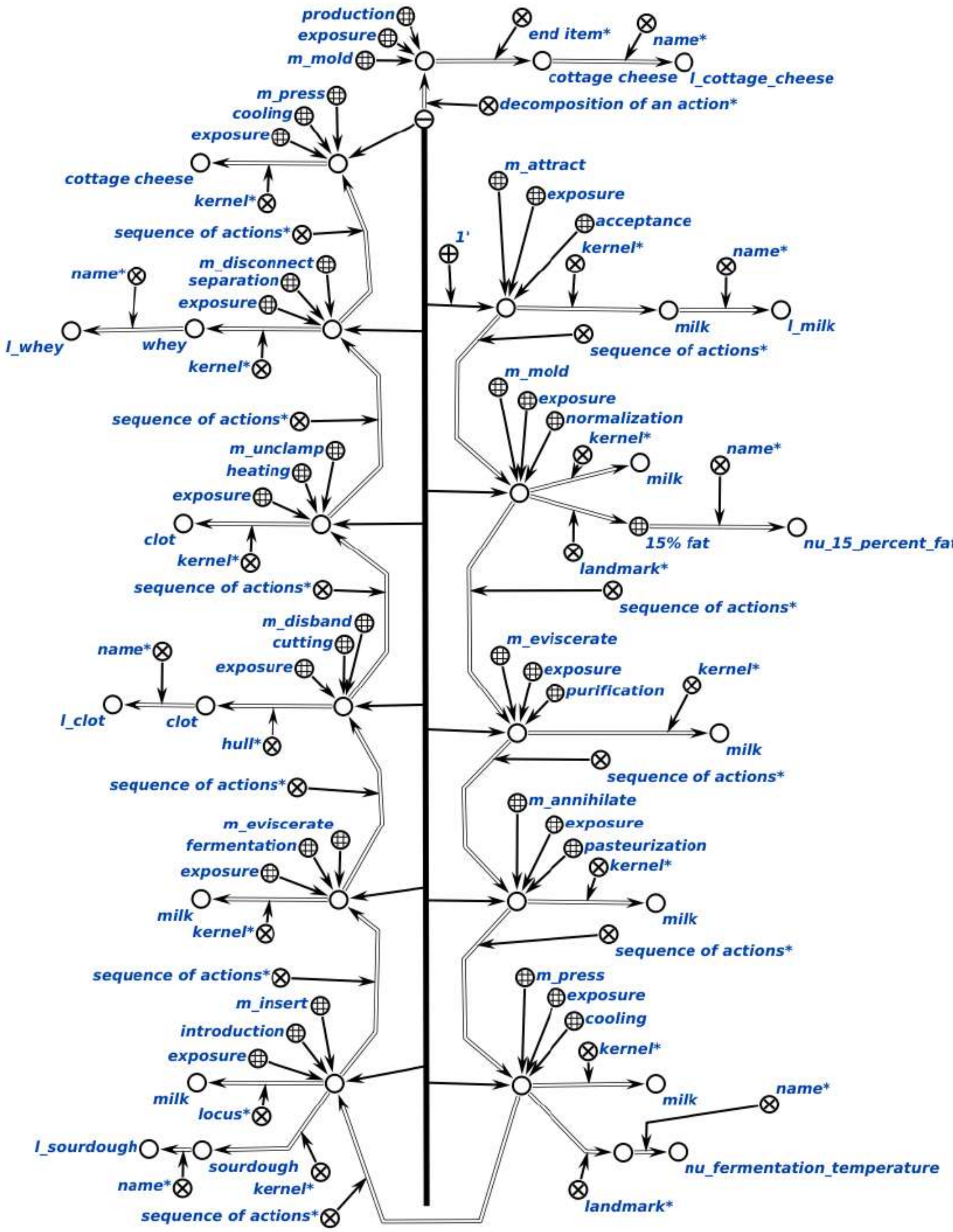


Figure 6. The result of the transition to the terms of TAPAZ-2.

творог»[2].

During the work, the following rules for the transition to semantics were also formulated:

- the grammatical predicate of the reconstructed text is mapped to the action (for example, the grammatical predicate *производит* = *produces*);
- the grammatical direct object of the reconstructed text is mapped to the grammatical object (for example, the grammatical direct object *творог* = *cottage cheese*).

When combining the transition rules from the initial version of the text to the reconstructed one with the rules

[2]"Someone accepts milk, then acidifies milk, namely: normalizes milk to 15% fat, then purifies milk, then pasteurizes milk, then cools milk to a certain temperature, then adds sourdough to milk, then ferments milk, then cuts the clot, then heats the clot, then processes the clot, then separates whey, then cools the clot and, as a result, produces cottage cheese".

for the transition to semantics, it is possible to obtain the following rules that provide a one-step transition:

- the grammatical direct object of the initial text is mapped to the action;
- the grammatical attribute of the initial text is mapped to the object.

The final result is shown in Figure 6.

VII. Integration into the knowledge base

The agent of merging structures in the knowledge base integrates the structure obtained as a result of analysis of the text into the knowledge base. The process involves searching for and resolving contradictions.

As an example, we will present a situation when there is a fragment in the knowledge base that describes an exposure that is of the same type as shown in Figure 7.

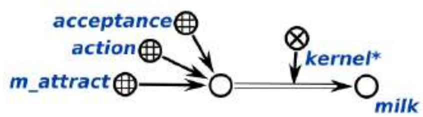


Figure 7. The construction that was present in the knowledge base before merging.

In this case, the models are merged. The resulting construction is shown in Figure 8.

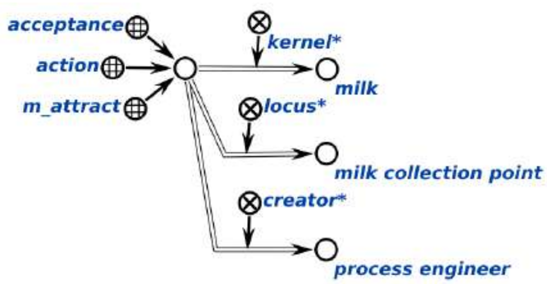


Figure 8. The result of merging constructs.

VIII. Conclusion

The article proposes a new approach to the machine understanding of texts in natural language (Natural Language Understanding, NLU), based on the formalization of the World Model using algorithms of the Theory for Automatic Generation of Knowledge Architecture (TAPAZ-2) and the immersion of the obtained semantic formalisms into the software environment using Open Semantic Technology for Intelligent Systems (OSTIS) that operates with original dynamic graph models – semantic networks in the form of specially oriented taxonomies and ontologies represented in the SC-code (Semantic Computer Code). The resulting taxonomic and ontological set is universal and can be used for machine understanding of collections of texts of various subject domains in various natural languages. The advantages of the approach are:

- decoding the meaning of signs and sense of sentences through decoding the patterns of the World Model, which provides the ability to support analytical activities and solve inventive problems not only by analogy [44, p. 39], [46, p. 192], [61, p. 16];
- standard dynamic graph representation of any type of knowledge within a single knowledge base, regardless of the platform or system [65]–[67];
- a unified top-level algebraic ontology adapted to the semantization of the Internet;
- machine-friendly parsing that provides a straightforward transition to automatic semantic markup of content;
- Semantic Classifier, Role List of Individuals and Knowledge Graph, significantly superior to their analogues in terms of the capacity of semantic calculus [11]–[28], [68], [69];
- mathematical, semantic and software algorithms that can significantly increase the accuracy and speed of operation of search engines;
- compatibility with statistical methods and any types of machine learning that scale the obtained results and reduce the complexity and labor intensity of the knowledge base development.

Acknowledgment

The authors thank the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics, the Department of Theory and Practice of the Chinese Language of the Minsk State Linguistic University for the help and valuable comments.

References

[1] A. Hardzei. Attributive-Predicative Syntagm as a Language Universal. *Inostrannyye yazyki v vysshey shkole* [Foreign languages in Higher School]. Ryazan, Ryazan State University Publ., 2019, iss. 4(51), pp. 7–19. (In Russian).

[2] A. Hardzei. Foundations of Combinatory Semantics. *Collection of Papers on Lexicography — Slovo i slovar' = Vocabulum et vocabularium* [The Word and the Lexicon]. Grodno, Grodno State University Publ., 2005, pp. 32–35. (In Russian).

[3] A. Hardzei. Linguistic Propaedeutics. *Materialy IV Respublikanskoy nauchnoy konferentsii "Belarus' v sovremennom mire"* [Proceedings of IV Republican Research Conference "Belarus in the Modern World"]. Minsk, The Republican Institute of Higher School Publ., 2005, pp. 226–229. (In Russian).

[4] A. Hardzei. *Printsipy ischisleniya semantiki predmetnykh oblastei* [The Principles of Evaluating the Semantics of Subject Domains]. Minsk, Belarusian State University Publ., 1998, 156 p. (In Russian).

[5] Yu. V. Knorozov. On the Classification of Signaling. *Osnovnyye problemy afrikanistiki* [Basic Problems of African Studies]. Moscow, Science Publ., 1973, pp. 324–334. (In Russian).

[6] E. V. Romanovskaya. The Theory of Fascination in the Context of the Problems of Literary Translation (on the Example of Translations of the Chinese Classical Poetry "Shi"). *Izvestiya Gomel'skogo gosudarstvennogo universiteta imeni F. Skoriny* [Proceedings of Gomel State University named after Francisk Skorina]. Gomel, Gomel State University Publ., 2019, no. 1(112), pp. 125–131. (In Russian).

[7] A. Hardzei. About Declarative and Procedural Knowledge Representation. *Inostrannyye yazyki v vysshey shkole* [Foreign languages in Higher School]. Ryazan, Ryazan State University Publ., 2021, iss. 3(58), in print. (In Russian).

[8] M. Minsky. A Framework for Representing Knowledge. *MIT-AI Laboratory*, memo no 306, June, 1974, pp. 1–81. [Online]. Available at: https://dspace.mit.edu/bitstream/handle/1721.1/6089/AIM306.pdf?sequence_2, (accessed 29.07.2021).

[9] V. V. Martynov. *V tsentre soznaniya cheloveka* [In the Center of Human Conscious]. Minsk, Belarusian State University Publ., 2009, 272 p. Available at: https://dokumen.tips/documents/-55cf9b12550346d033a49d65.html, (accessed 29.07.2021).

[10] Z. S. Harris. Co-occurrence and Transformation in Linguistic Structure. *Language*. Baltimore, 1957, vol. 33, no 3, pp. 283–340.

[11] Ch. J. Fillmore. The Case for Case. *Universals in Linguistic Theory*. New-York, Harper and Row, 1968, pp. 1–88.

[12] Ch. J. Fillmore. Types of Lexical Information. *Studies in Syntax and Semantics*. Dordrecht, Reidel, 1969, pp. 109–137.

[13] Ch. J. Fillmore. Verbs of Judging: An Exercise in Semantic Description. *Studies in Linguistic Semantics*. New York, Holt, Rinehart and Winston, 1971, pp. 272–289.

[14] Ch. J. Fillmore. Verbs of Judging: An Exercise in Semantic Description. *Studies in Linguistic Semantics*. New York, Holt, Rinehart and Winston, 1971, pp. 272–289.

[15] Ch. J. Fillmore. The Case for Case Reopened. *Syntax and Semantics*. New-York, Academic Press, 1977, vol. 8, pp. 59–81.

[16] Ch. J. Fillmore. Frame Semantics. *Linguistics in the Morning Calm*. Papers Prepared for the 1981 Seoul International Conference on Linguistics. Seoul, Hanshin Publishing Co., 1982, pp. 111–137.

[17] Ch. J. Fillmore. Frames and the Semantics of Understanding. *Quaderni di Semantica*. 1985, vol. 6, no. 2, pp. 222–254.

[18] Ch. J. Fillmore. "Corpus Linguistics" vs. "Computer-aided Armchair Linguistics". *Directions in Corpus Linguistics*. Proceedings from a 1991 Nobel Symposium on Corpus Linguistics. Stockholm, Mouton de Gruyter, 1992, pp. 35–66.

[19] Ch. J. Fillmore. Encounters with Language. *Computational Linguistics*. 2012, vol. 38 (4), pp. 701–718.

[20] Ch. J. Fillmore, B. T. S. Atkins. Towards a Frame-based Lexicon: The Semantics of RISK and Its Neighbors. *Frames, Fields and Contrasts*. Hillsdale (NJ), Lawrence Erlbaum Associates, 1992, pp. 75–102.

[21] Ch. J. Fillmore, B. T. S. Atkins. Starting where the Dictionaries Stop: The Challenge for Computational Lexicography. *Computational Approaches to the Lexicon*. Oxford, Oxford Univ. Press, 1994, pp. 349–393.

[22] Ch. J. Fillmore, B. T. S. Atkins. *Polysemy: Theoretical and Computational Approaches*. Oxford, Oxford Univ. Press, 2000, pp. 91–110.

[23] T. A. Stowell. *Origins of phrase structure*. Thesis (Ph. D.) Massachusetts Institute of Technology, Dept. of Linguistics and Philosophy, 1981, 496 p. Available at: http://www.ai.mit.edu/projects/dm/theses/stowell81.pdf, (accessed 29.07.2021).

[24] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web: A New Form of Web Content that is Meaningful to Computers Will Unleash a Revolution of New Possibilities. *Scientific American*. 2001, vol. 284, iss. 5, pp. 34–43.

[25] T. Berners-Lee, N. Shedbolt, W. Hall. Semantic Web Revisited. *IEEE Intelligent Systems*. 2006, vol. 21, iss. 3, pp. 96–101.

[26] Activity Vocabulary. W3C Recommendation. Available at: https://www.w3.org/TR/activitystreamsvocabulary/#dfn-activity, (accessed 29.07.2021).

[27] Semantic Web. Available at: https://www.w3.org/standards/semanticweb, (accessed 29.07.2021).

[28] Schema.org Vocabulary, version 8.0. Available at: https://schema.org, (accessed 29.07.2021).

[29] S. P. Rippa, O. M. Lyashenko. Semantic platforms of knowledge bases software in informatics. *Pratsi Odes'koho politekhnichnoho universytetu* [Proceedings of Odessa Polytechnic University]. Odessa, Odessa Polytechnic University Publ., 2013, vol. 1(40), pp. 91–96. (In Russian).

[30] V. V. Martynov. *Osnovy semanticheskogo kodirovaniya. Opyt predstavleniya i preobrazovaniya znaniy* [Foundations of Semantic Coding. Experience of Knowledge Representation and Transformation]. Minsk, European Humanitarian University Publ., 2001, 140 p. (In Russian).

[31] D. G. Bobrow, T. Winograd. An Overview of KRL, a Knowledge Representation Language. *Cognitive Science*. 1977, vol. 1 (1), pp. 3–46.

[32] A. Vežbicka. From the Book "Semantic Primitives". *Semiotika. Antologiya* [Semiotics. Anthology] / comp. by Yu. S. Stepanov. Moscow, Academichesky Proekt; Yekaterinburg, Delovaya Kniga, 2001, pp 42–270. (In Russian).

[33] Ashtadhyayi or Sutrapath of Panini. Available at: https://sanskritdocuments.org/doc_z_misc_major_works/aShTAdhyAyI.pdf, (accessed 29.07.2021).

[34] V. N. Toporov. On some analogies to the problems and methods of modern linguistics in the works of ancient Indian grammarians. *Kratkiye soobshcheniya Instituta narodov Azii* [Briefs of the Institute of the Peoples of Asia]. Moscow, USSR Academy of Sciences Publ., 1961, iss. 57, pp. 123–133. (In Russian).

[35] O. A. Voloshina. On the structure and linguistic terminology of Panini's grammar and its influence on European linguistics. *Vestnik RGGU. Seriya: Istoriya. Filologiya. Kul'turologiya. Vostokovedeniye* [Bulletin of the Russian State University for the Humanities. Series: History. Philology. Culturology. Oriental studies]. 2010, no. 9 (52), pp. 161–172. (In Russian).

[36] V. V. Martynov. *Semiologicheskie osnovy informatiki* [Semeiological Foundations of Computer Science]. Minsk, Science and Technology Publ., 1974, 192 p. (In Russian).

[37] V. V. Martynov. *Universal'nyy semanticheskiy kod. Grammatika. Slovar'. Teksty* [Universal Semantic Code. Grammar. Dictionary, Texts]. Minsk, Science and Technics Publ., 1977, 191 p. (In Russian).

[38] V. V. Martynov. *Universal'nyy semanticheskiy kod: USK-3* [Universal Semantic Code: USC–3]. Minsk, Science and Technics Publ., 1984, 134 p. (In Russian).

[39] V. V. Martynov. *Universal'nyy semanticheskiy kod: USK–4* [Universal Semantic Code: USC–4]. Preprint, no 2 / Institute of Linguistics of Academy of Science of BSSR. Minsk, Science and Technics Publ., 1988, 30 p. (In Russian).

[40] V. V. Martynov. *Universal Semantic Code: USC–5*. Minsk, Minsk State Linguistic University, 1995, 21 p. (Pre-print. MSLU; no. 4).

[41] A. Hardzei. Procedural Semantics and Evaluation of Subject Domains. *Materialy I Mezhdunarodnoy nauchnoy konferentsii "Yazyk: semantika, sintaktika, pragmatika"* [Proceedings of I International Research Conference "Language: semantics, syntactics, pragmatics". Minsk, Minsk State Linguistic University Publ., 1995, pt. 1, pp. 16–17. (In Russian).

[42] A. Hardzei. *Deduktivnaya teoriya yazyka* [Deductive Theory of Language]. Minsk, Belarussian Science Publ., 1998, 46 p. (In Russian).

[43] A. Hardzei. Theory for Automatic Generation of Knowledge Architecture (TAPAZ-2) and Further Minimization of Semantic Calculus. *Materialy IV Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii "Otkrytyye semanticheskiye tekhnologii proyektirovaniya intellektual'ykh system (OSTIS-2014)"* [Proceedings of IV International Scientific and Technical Conference "Open Semantic Technologies for Intelligent Systems (OSTIS-2014)"]. Minsk, Belarussian State University of Informatics and Radioelectronics Publ., 2014, pp. 49–64.

[44] A. Hardzei. *Theory for Automatic Generation of Knowledge Architecture: TAPAZ-2* / Transl. from Rus. I. M. Boyko. Rev. English edn. Minsk, The Republican Institute of Higher School Publ., 2017, 50 p. Available at: http://tapaz.by, (accessed 18.04.2021).

[45] A. Hardzei, A. Udovichenko. Graph of TAPAZ–2 Semantic Classifier. *Open Semantic Technologies for Intelligent Systems (OSTIS)*. Minsk, Belarussian State University of Informatics and Radioelectronics Publ., 2019, iss. 3, pp. 281–284.

[46] A. Hardzei. Plagiarism Problem Solving Based on Combinatory Semantics. *Communications in Computer and Information Science (CCIS)*. Switzerland, Springer Nature Switzerland AG, 2020, vol. 1282, pp. 176–197. Available at: https://link.springer.com/book/10.1007%2F978-3-030-60447-9), (accessed 29.07.2021).

[47] V. Golenkov, N. Gulyakina, N. Grakova, I. Davydenko, V. Nikulenka, A. Eremeev, V. Tarasov. From Training Intelligent Systems to Training Their Development Tools. *Open Semantic Technologies*

for Intelligent Systems (OSTIS). Minsk, Belarussian State University of Informatics and Radioelectronics Publ., 2018, iss. 2, pp. 81–98.

[48] T. N. Domnina, O. A. Khachko. *Nauchnyye zhurnaly: kolichestvo, tempy rosta* [Scientific Journals: Quantity, Growth Rates]. Available at: http://www.benran.ru/SEM/Sb_15/sbornik/ 83.pdf, (accessed: 29.07.2021). (In Russian).

[49] T. V. Chernigovskaya. *Kak internet izmenil nash mozg* [How the Internet Has Changed Our Brain]. Available at: https://golbis.com/pin/neyrolingvist-tatyana-chernigovskaya-kakinternet-vliyaet-na-nash-mozg-8, (accessed: 29.07.2021). (In Russian).

[50] S. Ivanov. *Kolichestvo publikatsiy nauchnykh rabot v mire yezhegodno uvelichivayetsya, prichem osnovnoy protsent rosta prikhoditsya imenno na razvivayushchiyesya strany* [The number of Scientific Publications in the World Is Increasing Every Year, with the Bulk of the Growth Accounted for by Developing Countries]. Available at: https://hightech.fm/2018/12/24/science-2, (accessed: 29.07.2021). (In Russian).

[51] V. Golenkov, N. Guliakina, I. Davydenko, A. Eremeev. Methods and tools for ensuring compatibility of computer systems. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem* [Open Semantic Technologies for Intelligent Systems (OSTIS)]. Minsk, Belarussian State University of Informatics and Radioelectronics Publ., 2019, iss. 3, pp. 25–52.

[52] A. Hardzei. Parts of Language Instead of Parts of Speech. *Yazyk. Glagol. Predlozheniye* [Language. Verb. Sentence]. Smolensk, Smolensk Pedagogical State University Publ., 2000, pp. 258–271. (In Russian).

[53] A. Hardzei. The Paradigm of Parts of the Language. *Materially VIII Mezhdunarodnoi konferentsii —Slovoobrazovaniye i nominativenaya derivatsiya v slavyanskikh yazykakh* [International Research Conference —Word-Building and Nominative Derivation in Slavic Languages]. Grodno, Grodno State University Publ., 2003, pp. 173–179. (In Russian).

[54] A. Hardzei. Parts of Language and the Procedures of Its Delineation. *Puti Podnebesnoi* [The Paths of the Middle Kingdom]. Minsk, Belarusian State University Publ., 2006, iss. 1, pt. 1, pp. 69–75. (In Russian).

[55] A. Hardzei. The Meaning of Logical and Semantic Paradoxes for Language Theory. *Karpovskiye nauchnye chteniya* [Readings from V. A. Karpov]. Collection of Research Papers in 2 parts. Minsk, Belarussian State University Publ., 2011, pt. 1, pp. 9–18. (In Russian).

[56] A. Hardzei, Sun Kewen. Typological Aspects of Word Formation. *Puti Podnebesnoi* [The Paths of the Middle Kingdom]. Minsk, The Republican Institute of Higher School Publ., 2011, iss. 2, pp. 40–53. (In Russian).

[57] A. Hardzei. Virtual String as a Syntactical Code of a Sentence (by the Examples of the Chinese Language). *Materialy mezhdunarodnoy nauchnoy konferentsii "Yazyk, obshchestvo i problem mezhkul'turnoy kommunikatsii"* [Proceedings of International Research Conference "Language, Society and Problems of Intercultural Communication"]. Grodno, Grodno State University Publ., 2007, pt. 2, pp. 349–358. (In Russian).

[58] A. Hardzei. About the Combinatorics of Numerical Signs in the Chinese Language. *Nauchnyye chteniya, posvyashchennyye Viktoru Vladimirovichu Martynovu* [Scientific readings dedicated to Viktor Vladimirovich Martynov]. Collection of Research Papers. Minsk, The Republican Institute of Higher School Publ., 2015, iss. 2, pp. 12–14.

[59] A. Hardzei. Dynamic Syntax: A Semantic View. *Inostrannyye yazyki v vysshey shkole* [Foreign languages in Higher School]. Ryazan, Ryazan State University Publ., 2017, iss. 4 (43), pp. 26–34. (In Russian).

[60] A. Hardzei. *Teoreticheskaya grammatika vostochnykh yazykov: lektsionnyy kurs* [Theoretical Grammar of Oriental Languages: Lecture Course]. Electronic data. Minsk, 2007, 1 electronic optical disc (CD-ROM).

[61] A. Hardzei. Semantic Markup of the Event and its Display by Means of the Chinese and Russian Languages. *Inostrannyye yazyki v vysshey shkole* [Foreign languages in Higher School]. Ryazan, Ryazan State University Publ., 2021, iss. 2 (57), pp. 5–26. (In Russian).

[62] M. I. Svyatoshchik. Translating Multi-Object Semantics into SingleObject Semantics as a Method of Minimizing Semantic Calculation. *Uchonyye zapiski VGU im. P. M. Masherova: sb. nauch. trudov [Scientific notes of Vitebsk State University named after P. M. Masherov*: collection of scientific papers]. Vitebsk, *Vitebsk State University Publ.*, 2020, vol. 31, pp. 180–184. (In Russian).

[63] A. Hardzei. Cognitive Approach to Teaching of Signs. *Komp'yuternyye programmy v obuchenii belorusskomu i inostrannym yazykam: Tezisy dokladov nauchauchnoy konferentsii* [Computer Programs in Teaching Byelorussian and Foreign Languages: Abstracts of the Research Conference]. Minsk, Minsk State Linguistic University Publ., 1995, pp. 18–20. (In Russian).

[64] A. Hardzei. Procedural Semantics and Calculus of Subject Domains. *Yazyk: semantika, sintaktika, pragmatika: Tezisy dokladov nauchnoy konferentsii* [Language: Semantics, Syntactics, Pragmatics: Abstracts of the Research Conference]. Minsk, Minsk State Linguistic University Publ., 1995, pt. 1, pp. 16–17. (In Russian).

[65] I. T. Davydenko. Semantic Models, Method and Tools of Knowledge Bases Coordinated Development Based on Reusable Components. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem* [Open Semantic Technologies for Intelligent Systems (OSTIS-2018)]. Minsk, Belarussian State University of Informatics and Radioelectronics Publ., 2018, iss. 2, pp. 99–118.

[66] D. Shunkevich. Agent-oriented Models, Method and Tools of Compatible Problem Solvers Development for Intelligent Systems. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem* [Open Semantic Technologies for Intelligent Systems (OSTIS-2018)]. Minsk, Belarussian State University of Informatics and Radioelectronics Publ., 2018, iss. 2, pp. 119–132.

[67] I. T. Davydenko. Ontology-Based Knowledge Base Design. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem* [Open Semantic Technologies for Intelligent Systems (OSTIS-2017)]. Minsk, Belarussian State University of Informatics and Radioelectronics Publ., 2017, iss. 1, pp. 57–72.

[68] R. Jackendoff. *Semantic Interpretation in Generative Grammar*. Cambridge, MA, MIT Press., 1972, 400 p.

[69] R. Jackendoff. *X–bar Syntax: A Study of Phrase Structure*. Cambridge, MA, MIT Press., 1977, 249 p.

[70] V. V. Golenkov, N. A. Gulyakina. A Project of an Open Semantic Technology for the Component Design of Intelligent Systems. Part 1: Principles of Creation. *Ontologiya proyektirovaniya* [Ontology of Design]. Samara, Samara National Research University Publ., 2014, no. 1, pp. 42–64. (In Russian).

[71] V. V. Golenkov, N. A. Gulyakina. A Project of an Open Semantic Technology for the Component Design of Intelligent Systems. Part 2: Unified Design Models. *Ontologiya proyektirovaniya* [Ontology of Design]. Samara, Samara National Research University Publ., 2014, no. 4, pp. 34–53. (In Russian).

Обработка и понимание естественного языка интеллектуальной системой

Гордей А. Н., Святощик М. И.,
Бобёр Е. С., Никифоров С. А.

Статья посвящена обработке данных на естественном языке в парадигме Теории автоматического порождения архитектуры знаний (TAPAZ-2) и погружении полученных семантических формализмов в программную среду посредством Открытой семантической технологии для интеллектуальных систем (OSTIS). Особенностью подхода является формализация семантики естественного языка с опорой на модель мира и сочетание семантического кодирования с онтологией и таксономией семантических сетей.

Received 01.06.2021

Conversational speech analysis based on the formalized representation of the mental lexicon

Vadim Zahariev, Sergei Nikiforov, Elias Azarov
Belarussian State University of
Informatics and Radioelectronics
Minsk, Belarus
Email: zahariev@bsuir.by, nikiforov.sergei.al@gmail.com, azarov@bsuir.by

***Abstract*—This article is dedicated to the study of the possibilities of formalization and modeling with the help of modern approaches in the field of artificial intelligence such a structure of human consciousness as "the mental lexicon". According to experts in the field of psycholinguistics [1], [39], it is a large graph of concepts and lexemes of a language, related to each other by semantic connections, which plays an essential role in the processes of generation and perception of speech by a human [2], [40].**

The usage of semantic networks for modeling this structure allows using the advantages of this form of representation in full for solving problems of natural speech understanding by technical systems. This approach also implies the need to develop methods of analysis and processing of the speech signal, which precede the stages of semantic processing, closer integration of these stages and, in a sense, the erasing of the boundaries between them, taking into account how this occurs in the process of speech perception by the human brain [41]–[43]. This article is concerned with the possibilities to show the advantages of such a complex approach to the analysis of speech messages through its implementation based on the Open Semantic Technology for Intelligent Systems (OSTIS).

***Keywords*—mental lexicon, semantic network, voice assistant, intelligent personal assistant, spoken language understanding**

I. INTRODUCTION

Currently, the systems of intelligent personal assistants have become widespread and popular. This is proved by the large number of products related to the usage of artificial intelligence methods in the field of speech technologies, which are appearing on the market [3], as well as the attention that leading technology companies pay to the development of this direction [4].

As a rule, modern assistants have two main modes of conducting a dialogue: command and free dialogue on random topics ones [5].

The command mode is based on the search for a certain keyphrase called "intent" in the user speech and the implementation of a response action, depending on the type of this intent. This mode allows, through the speech interface, getting important for the user information, performing some useful actions related to a certain class of intent (ordering a taxi, turning on and off the music or some device in the house, requesting a location or weather for a certain date, searching for the nearest interesting for the user place, etc.) [5]. If the system cannot precisely classify the intent of the user, it transmits it in text form to an Internet search engine, which should give some relevant answer based on a combination of keywords.

A random mode, when the system tries to simulate conducting a dialogue with a human, keeping up a conversation on general topics, depending on the initial assumptions contained in the user utterance. As a rule, in this case, the recognition of input phrases is carried out with an unlimited dictionary, when it is assumed that the user can ask anything and the system, based on its internal "knowledge" and the model of conducting a dialogue, should "understand" the utterance and formulate the most probable answer from the point of view of a human. It should be noted that modern systems add various stylistic language techniques, emotions, moods or even humor to the conversation for making a dialogue realistic. To prevent errors in case of incorrect interpretation of the input phrase, it may ask the user to repeat one or more of the last phrases [5].

By default, the system starts in command mode. Switching between modes is performed automatically by some key "intents", when the system understands that the user just wants to "talk" but not give the system concrete target designations.

The modern voice assistant is a distributed software and hardware system that consists of two main parts: a client and a server ones (fig. 1).

From the point of view of the implementation of the architecture modules, modern speech assistants consist of the following main modules implemented with the help of such libraries:

- automatic speech recognition (ASR) – "Deep-Speech", "Kaldi", "Vosk" [6]–[8];
- natural language understanding (NLU) and dialogue managment (DM) – "Wit.AI", "Dialogflow", "Rasa NLP", [9], [10];
- natural language generation (NLG) – "GPT2", "BERT", "GPT3" [11], [12];
- text-to-speech synthesis (TTS) – "WaveNet", "Taca-troon", "Nvidia Nemo TTS" [13]–[15].

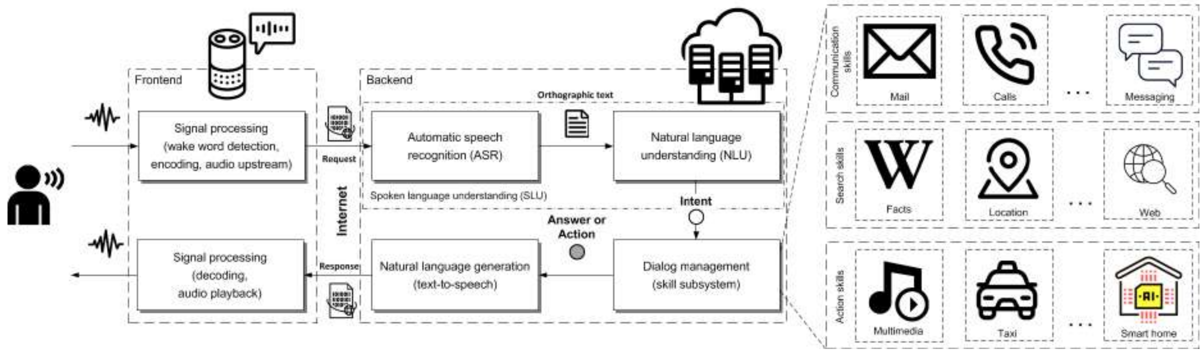


Figure 1. The intelligent speech assistant system architecture

In the current implementations of speech assistants, most of these modules are fundamentally based on neural network models and information processing methods. There are a lot of open source libraries and frameworks (including those listed above) that allow implementing the functionality necessary for the operation of these modules. Also, major technology companies have their products that are being developed on a proprietary basis [4].

It is an established matter that the currently dominant neural network approaches to solving this problem allow achieving top results relative to classical machine learning models, rule-based systems and other classical methods of building dialogue systems [16]. However, these approaches also have their peculiarities and limitations, primarily related to the strong dependence of the final characteristics of the models on the quantity and quality of training data, sample size, choice of network architecture, the complexity of formalization and understanding of the processes that occur inside in the learning process, computational constraints of using large neural network models on end devices with limited resources. Therefore, it seems quite logical and promising to use data-based methods, which include neural networks together with other artificial intelligence methods that allow effectively formalizing the processes of storing and processing information inside the dialogue assistant.

The further development of intelligent personal assistants will be quite difficult without extensive analysis of the nature of the speech message, the peculiarities of its perception at all stages, including at the semantic and even pragmatic levels.

One of the fundamental structures of consciousness, called in the psycholinguistic literature “mental lexicon” [17], [18], [20], has a great significance in the acts of speech message perception and understanding. Scientists describe this structure as an internal dictionary, which is a network of connected concepts (words), which are used by a human when enabling speech communication. In addition to the mental lexicon, there is also the concept of a “mental grammar” of some add-in over the “mental lexicon”, where the rules that we use when generating and perceiving word forms and even whole sentences are stored [53].

From our point of view, semantic networks and intelligent agents that run in connection with them can serve as a good abstraction for modeling such structures of human consciousness, the properties and purpose of which for computer systems can largely overlap with the purpose of the “mental lexicon” units for a human, as it is considered by specialists in psycholinguistics [21].

The question of displaying this structure in the framework of an intelligent system with a natural-language interface for building dialogue systems of a new class, the building of which will be discussed in this article, seems particularly intriguing.

II. Problem definition

The mental lexicon is a term that denotes, in a wide sense, how words are represented and systematized in the human consciousness [17], [18]. Most scientists agree that the mental lexicon can be described as a giant network of concepts (fig. 2), where words that are close in meaning and similar in sound (writing, in the case of written speech) are connected [19]. The affinity and organization of these concepts, primarily by semantic and phonetic markers, has been proved by a series of various experiments [22].

The mental lexicon is one of the most important components of a language ability of a human, in which notions about the world and their lexical representation are fixed. The units of the mental lexicon are connected into a single complex dynamic system that can be rebuilt, depending on the situation. The mental lexicon has a complex multi-sided and multi-level structure with intra- and interlevel connections (fig. 3) [19], [44], and in its organization, it is possible to distinguish the core and peripheral areas, for example, based on the criteria of the frequency of usage of concepts [45], [46].

Taking into account the volatility of the lexicon, it can be stated that the boundaries between the core and the periphery cannot be invariant because these structures are in relations of dynamic connection. The core changes dynamically during human life since new units are assimilated by the individual and some, on the contrary, may fall outside the core as a result of a change of professional activity, place of residence, social status, etc. Nevertheless, the analysis of research has shown that vocabulary that reflects national, social, professional, age and other realias of the outside world of the individual is prevalent in the core [47].

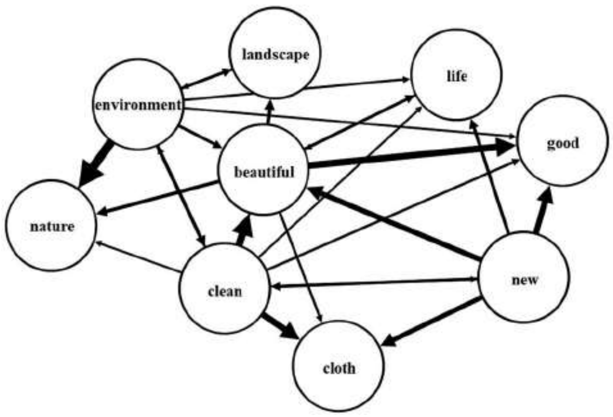


Figure 2. A fragment of the mental lexicon that includes the semantic neighborhood of the concept "environment" [19]

The agility and mobility of the units of the mental lexicon in the direction from the core to the periphery and vice versa is provided by the type of connections between them, based on the semantic similarity of the units. In the mental lexicon, those connections of units are active, which are reflected in integrations considered in traditional semantics, such as a lexical-semantic field, a lexical-semantic group, a lexical-thematic group, a thematic group, etc.

Modern models of mental language representation differ not only in their attitude to the oppositeness of lexicon and grammar but also in their directivity to the speech generation or perception as well as compatibility with languages of different structures. Although the mental and neurophysiological mechanisms for enabling speech activity are the same, the procedures vary significantly depending on the type of language [23], [24].

Due to this fact, for languages of different types, the description of the mental lexicon in the form of a single model is hardly possible. The actual models of the mental language representation are grouped into four classes [23]:

- single-system models that do not separate lexicon and grammar;
- two-system models that separate the lexicon as an inventory of units from grammatical rules;
- hybrid models that allow, along with the integrated one, elementwise storage of word forms and the usage of rules for combining morphemes in word forms;
- models without a mental lexicon.

An essential part of the research of the mental lexicon by psycholinguists correlates with hybrid models focused on speech perception [48].

Hybrid models assume both gestalt (perceived as an integral whole) storage of word forms and the decomposition of a word form into morphemes; at least productive affixes can be stored as independent units and, accordingly, act as operational units when planning an utterance. In this case, the storage of units and their operation occurs due to three levels of representation: the notion of morphemes, the notion of a complete word form, the notion of a lemma. The word form is extracted from the mental lexicon as a gestalt, although its elementwise extraction is possible both for recognition when perceiving speech and generating an utterance. The lemma is interpreted in different ways, but all interpretations are unified by its relation to semantics; rather, the lemma mediates the transition from formal representations (the sound/graphic image of a word form) to the node of the semantic network [25]. The lemma represents the general meaning of the lexeme, thus providing access to the concept as an item of the semantic network [26].

Current models of the mental lexicon, along with the generalization of the results of experiments in standardized methods, often explicitly or implicitly rely on the concepts of early psycholinguistics. The possibility of ambivalent comprehension of a derivative (and its word forms) both entirely and on the basis of the elementwise analysis is proved in the following papers [25], [49]. If a word form represents a storage unit in almost all models of the mental lexicon, then in the process of searching for access to it, it is possible to operate with other units: morphemes (quasi-morphemes) for perception and lemmas for speech generation. The storage unit may not coincide with the operational one – the unit that the speaker or listener operates with [50].

The mental lexicon is often represented as an associative-verbal network [51], as a "dynamic functional system that organizes itself due to the constant interaction between the process of processing and ordering speech experience and its products" [52].

In this case, linguistic (verbal) units can act as a storage unit in the mental lexicon. The representation of morphemes, integral word forms and lemmas in the mental lexicon at its different levels within the framework of the hybrid model suggests the ability to assess the accuracy of word form recognition based on grammatical features as well as segmental and suprasegmental phonetic characteristics. Moreover, the variability of the sound image of a word form in the conversation suggests the presence of such a mechanism of speech perception that would allow correlating a variable acoustic signal with the perceptual standard of a language item stored in consciousness and associated with the lemma level. The operation of this mechanism is based on probabilistic

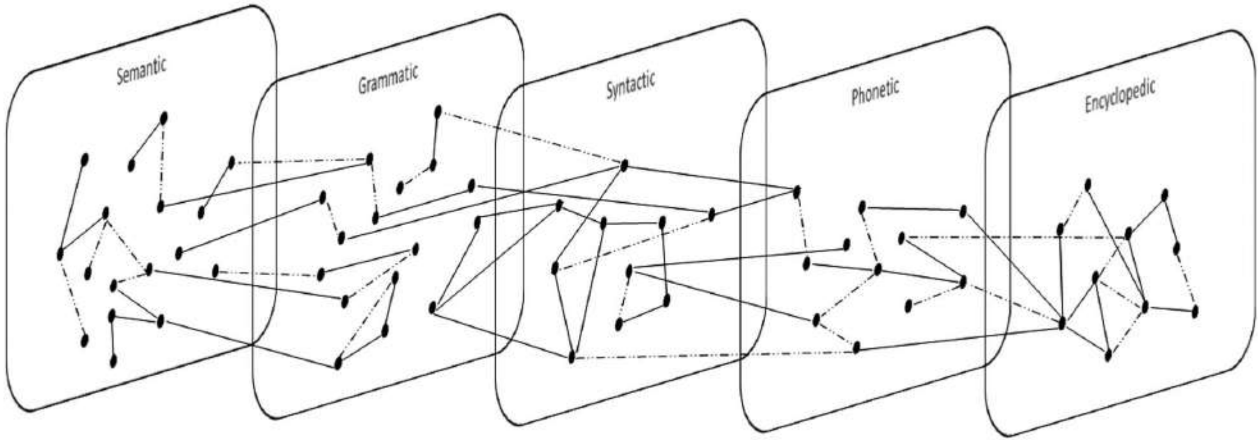


Figure 3. Possible layers of the mental lexicon [19]

forecasting of the subsequent signal of the speech chain based on various features [50]. In essence, the study of speech perception in this aspect is an analysis of the mechanisms of access to the units of the mental lexicon.

In the memory of a native speaker, the standards of the sound image of words and the main syntactic constructions (including the intonation contour) that correspond to the sentence are stored. The complex of linguistic features characterizes the expression plane of language units and, thanks to this, mediates access to a perceptual standard correlated with the content plane, that is, language semantics.

Perceptual standards combined in the perceptual language base are correlated with the units of the mental lexicon: perceptual standards of the lexical level are a unit of access to the word-form level, correlated with lemmas and thus with semantic representations. An actual linguistic problem is the modeling of the perceptual language base as a hierarchical probabilistically organized structure of multidimensional matrices of language units of different levels, permeated with multiple intersecting perceptually significant linguistic features. Characterizing the form of language units, the complex of features mediates access to the perceptual standard and intromission to the semantic level [2], [53].

The significance of the mental lexicon for the processes of speech perception and understanding, indicated above, demonstrates the relevance of the research of this psycholinguistic concept as well as the importance of the problem of implementing structures like the described one as part of intelligent systems with a natural-language interface. This can lead, including due to the synergetic effect, to an improvement in the quality of the model operation in the current implementations of the systems as well as confer them new useful properties. Thus, the purpose of this article is to study the possibilities of a modeling and implementation of the mental lexicon and the potential benefits that can enable its implementation as part of a new generation of speech assistants.

To achieve this purpose, it is necessary to solve the following main problems:

- to perform modeling and formalization of the network part of the mental lexicon as a complex system of hierarchical concepts connected by semantic, perceptual, frequency, etc. features that allow a human to carry out the processes of conceptualization, formulation, perception and articulation in the process of speech communication;
- to choose and implement an adequate model of speech perception mechanisms that is related as closely as possible to the notion of this process and its connection with the mental lexicon.

It should be noted that in the case of modeling the mental lexicon for its implementation as part of technical systems, other important and relevant from a scientific and practical points of view problems can be identified. However, the two points listed above are of the highest priority in terms of their significance and those useful properties that can be enabled by the implementation of the mental lexicon as part of an intelligent system. Approaches to their solution are the subject matter of this article.

III. Proposed approach

To solve the two abovementioned problems, we propose two main approaches for consideration:

- the usage of the semantic network appliance for modeling the structure of the mental lexicon and the connections between its elements as one of the similar models of the representation and organization of knowledge implemented in technical systems;
- taking an approach based on semantic-acoustic analysis, represented by us in a series of previous articles [30]–[32], which offers a solution to how it is possible to make a direct connection between the acoustic image of a word and its meaning in

the semantic network (in our case, in the network of the mental lexicon), as it occurs in the process of perception of speech information by a human, and how to connect the acoustic image of a word with its concept in the semantic network in the most direct way.

IV. Semantic level

A semantic network is a model for representing knowledge in a formalized form, which allows describing any information in the form of a graph structure [54], [55]. Nodes in such networks are concepts connected by semantic relations that act as edges of the graph [27]. Each node of the network has an important property from the point of view of semantics, namely, each element is associated with some of its "meaning" and "sign" (in the classification according to Ch. S. Peirce), i.e., the direct meaning of what this node means, as well as how this concept is represented in the area of perception, is its external form [57]. As an example, it is possible to introduce the concept "house" and a picture of a house as a visual image of this concept. In the case of a mental lexicon, an analogy with the acoustic or orthographic image of a word and its meaning (the place of the concept) in the mental lexicon associated with this sign can be drawn.

The usage of semantic networks for knowledge representation has the following main advantages [58]–[60]:

- they allow quickly and easily adding information, drawing conclusions based on the connections between different concepts;
- they ensure coherence and the absence of duplication of concepts;
- the information in the network can be easily interpreted by both a machine and a human;
- they can be used equally effectively to describe any subject domain.

The semantic network reflects the structure of the subject domain, the model of which it is, in an "almost isomorphic" way. So, the structure of any subject domain can be represented in the form of a semantic network quite simply, "almost isomorphically" (in the sense specified above). In other words, the correlation between the subject domain and the corresponding semantic network is quite transparent and has no unnecessary complications due not to the essence of the matter, not to the structure of the subject domain but to the peculiarities of the description language. Within the framework of this correlation, there is a fairly simple semantic interpretation of various types of elements of the semantic network (nodes, bindings, incident relations of bindings, key nodes) [61].

To an extent, semantic networks can also include the forms of information representation that humanity has been dealing with for a long time. The generally accepted ways of representing schematic electrical diagrams, logical circuits, flow diagrams are also nothing more than semantic networks represented in various alphabets. It is also obvious that semantic networks also include such ways of representing information as cognitive maps, knowledge maps and much more [28].

Such a seemingly simple and intuitive form of information representation allows formalizing knowledge from any subject domain quite fully, placing it in a model called an ontology, which has important properties for searching and extracting this information [29]. Something similar occurs in the process of human ontogenesis, when all the acquired knowledge and experience sum up in a complex system of concepts connected by nonlinear relations, where the search is carried out on the basis of semantic, associative, frequency and many other complex criteria for information search. The representation of such a structure at the linguistic level of human functioning, in fact, is the mental lexicon, the computer model of which will be nothing more than the knowledge base of the intelligent system.

At the same time, let us emphasize that the semantic network as an abstract mathematical structure should be clearly distinguished from different variants of its implementation and representation in computer memory, including graphical visualization. Various semantic networks can have different alphabets of elements (different sets of labels on the elements of semantic networks) [28], [62]. Each of these implementations has its characteristics, strengths and limitations. One of the platforms that have proven their effectiveness in the development of intelligent systems based on semantic networks is the OSTIS Technology. The features of using this technology and its advantages for modeling the mental lexicon will be considered below.

V. Acoustic level

In terms of modeling the process of speech perception, it should be noted once again that the study of speech recognition in this aspect is an analysis of the mechanisms of access to the units of the mental lexicon [2], [25], [48].

As was already mentioned above, the standards of the sound image of words and the main syntactic constructions (including the intonation contour) that correspond to the sentence are stored in the memory of a native speaker. As storage units in the mental lexicon, linguistic (verbal) units can serve. The representation of morphemes, integral word forms and lemmas in the mental lexicon at its different levels within the framework of the hybrid model implies the ability to assess the accuracy of word form recognition based on grammatical features as well as segmental and suprasegmental phonetic characteristics.

Separately, it should be emphasized that if the question is about the perception of oral communication, the nodes of the lexicon are connected directly with their corresponding acoustic images and not their analogues in

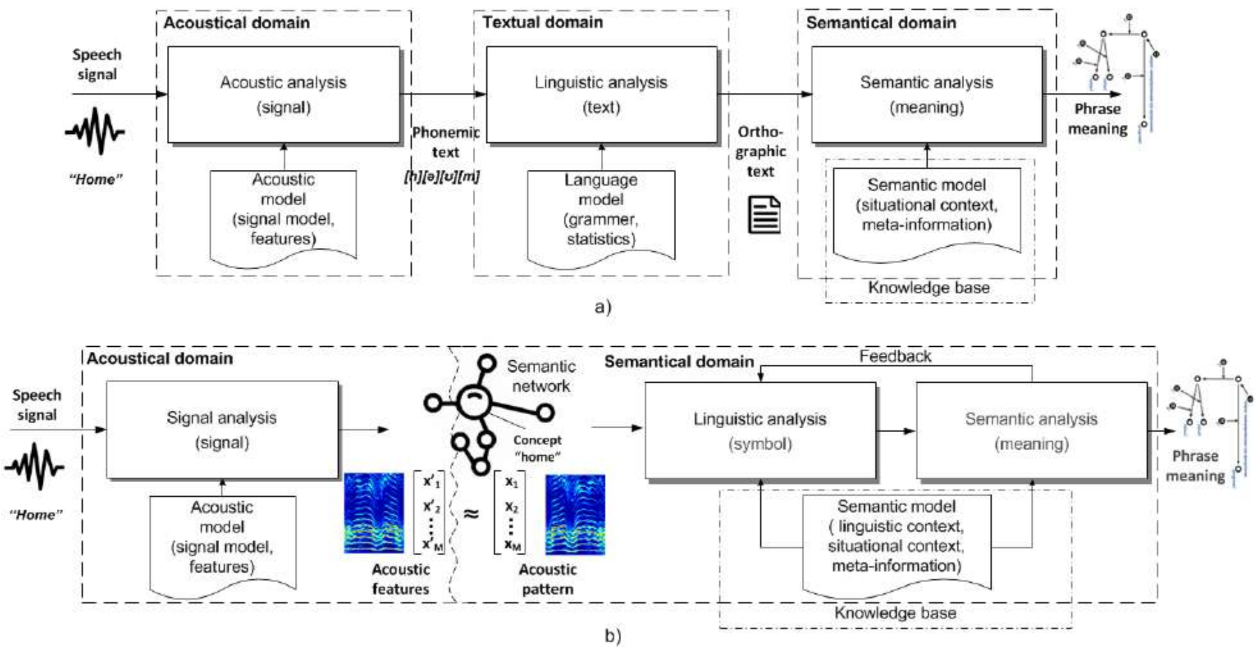


Figure 4. The conceptual architecture of the system: a) the default 3-tier approach; b) an approach based on semantic-acoustic analysis

the form of a text or other representation. Text notations associated with lexicon units are used in the process of perception and transmission of speech in written, not oral, form. There is no contradiction here because several signs of different types can correspond to the same semantic concept. For example, the concept "house" coincides with an acoustic image of the word "house", its text entry in the corresponding language, a visual image in the form of an image of a house. All these signs are equivalent in terms of the efficiency of their usage for accessing the nodes of the mental lexicon.

This fact of the organization of processes of processing auditory information differs significantly from the principles, on which modern speech recognition and understanding systems run when there is a rigid hierarchy of recognition stages. At the initial stage, the basic acoustic units of the language, phonemes and allophones, are transmitted (recognized) into their textual representation, in the future, at higher levels of language information processing (syntactic and semantic ones), only their textual representation is used.

Thus, the acoustic image of the concept is subordinated to the text image, which is not present in real life but is important for the current level of development of recognition systems, including as part of speech assistants, since it reduces the complexity of implementing current solutions. Therefore, we consider it essential to create methods of signal analysis that would allow more naturally describing the fact of the interrelation of verbal signs in acoustic form with the nodes of the mental lexicon.

In this article, we propose to use the method of semantic-acoustic analysis. The ideas of this method were proposed and considered by us in the papers [30]–[32]. It allows performing the primary parsing and parameterization of a speech message using special signal processing techniques. In the course of their application, certain "acoustic images" that correspond in our case to the concepts of the mental lexicon are separated from the conversation, which in turn will correspond to certain nodes (signs of concrete entities or concepts) in the semantic network.

In this article, at the current level of development of the proposed approach, it is recommended to extract and analyze sound images of concepts according to the principle of audio dactyloscopy systems ("audio fingerprinting") [36]. This method involves comparing the selected audio fragments of the signal, represented in parametric form, with a dictionary of standards and determining the proximity measure of the selected "acoustic image" to the standard of this image – associated with the corresponding node of the semantic network of the mental lexicon. The degree of confidence measure, in this case, is rather a proximity measure of the selected signal fragment to the reference one in the selected parametric space.

In general, the following information may be required to carry out this kind of analysis within the framework of the proposed approach:

- a set of standards for correlation with fragments of a speech signal and their specification;
- the context of the analyzed message (from whom the message was received, in what external conditions, what other sounds are present in the background,

etc.);

- a set of rules for the transition from fragments of a speech message to semantically equivalent constructions in the knowledge base;
- a semantic specification of the concepts included in such constructions.

For speech analysis, a model based on a hybrid representation of a speech signal will be used, which allows representing any fragments of a speech signal of different nature of sound generation in the most adequate form [33].

Vocalized and non-vocalized fragments of the signal are described by periodic (harmonic) and aperiodic (noise) components, which are important for the perception and differentiation of various acoustic images of concepts and are described by various components of the model. Details of the model implementation will be presented below during the description of the system architecture.

From our point of view, such a joint implementation of the abovementioned approaches focused on modeling the mental lexicon using semantic networks and semantic-acoustic analysis will allow acquiring of new useful properties by the intelligent personal assistant: additional flexibility and adaptability, the ability to overcome the limitations inherent to the systems presented on the market at the moment. All this will eventually allow such systems to come up to a whole new level.

VI. Practical application

It would be desirable to consider the possibilities of using the presented approaches on a practical example of one of the most relevant fields of application of speech assistants. At the moment, such a direction is the systems of semi-automation of the infrastructure of accommodation units, which are called "smart home" systems in common terms.

All these systems are built according to the same scheme for recognizing a selective list of commands transmitted by the user from an application with a graphical interface or through a voice control interface, which the system implements, and this scheme has a set of parameters embedded in it that are configured during the programming and deployment of the system.

For the expansion and reprogramming of such systems, special knowledge and skills are required to add certain new nodes to the system, to ensure their correct configuration depending on the operating conditions. Often, all these actions require a highly qualified user or a house-call of high-paid specialists.

In this context, speech assistants act as part of the system, providing speech input, interpretation and transmission of commands to the smart home controller as well as notification of the current state of the system. Due to the lack of flexibility and limitations of the current implementation, they do not allow the user of speech interface tools to perform "fine-tuning" of the system, changing the system parameters just-in-time and adapting the system to changing external conditions: the appearance of new users of the system who are not familiar to it, fine-tuning the system depending on changes in the weather during the day or the time of day.

As a cover of our theses, let us present several scenarios for using the smart home system that are not available in current system implementations, where improved adaptability of the system to changing conditions is necessary:

- *Example 1.* Reprogramming of the access control system just-in-time. Imagine that guests came to the owners of a detached home for dinner. A smart home equipped with the option of controlling access to rooms by identifying the user by voice or face image (this option is considered in detail in the article "Semantic analysis of the video stream based on neuro-symbolic artificial intelligence") from microphones or watching cameras of a smart home. As part of the standard implementation of the smart home system, new rules would have to be added by reconfiguring the system. Having a flexible system based on the principles described above, it would be possible to "introduce" new users to the system and order it to issue access permissions to new persons for a specified period immediately at the moment of their first appearance. The system would formulate the rules independently, according to which it can provide guests with access to the rooms of the house and adhere to them during the allotted time.
- *Example 2.* Adaptive fuzzy lighting control. During the day, significant fluctuations in the intensity of natural light can be detected within a short time. This is especially true for countries located in the midland with an oceanic or moderate-continental climate during the mid-seasons of the year when there is often a change in atmospheric phenomena. The settings of the lighting parameters of sensors and controllers are calibrated to some average values, without taking into account the complex dependencies of changes in external parameters. However, during the day, there may be situations when it is necessary to adjust the overall picture of the lighting of the rooms of the building, related to the daily round: the presence or absence of humans, work that requires lighting or not, rest schedule, etc. In modern systems, there is no functional ability to make the light in a certain room "brighter" or "more subdued" without specifying the exact intensity value. Moreover, it is often difficult for the user to give a clear command: "Make the lighting intensity 15% more". Quite often, the user does not know about the acceptable scale of intensity values and its dynamic

range. It would be much easier, in this case, to "ask" the system to "make the lighting a little brighter" or "a little more subdued", and how much is "a little", taking into account the current level of illumination of the room and the lighting parameter profile, the system should be able to determine and implement itself.

Thus, the implementation of the system based on the modeling of the mental lexicon with the help of semantic networks and speech analysis using the semantic-acoustic approach would give additional flexibility to the semantic core of the system, provide the possibility of forming new rules in the knowledge base of the system directly, by means of the language interface in the interaction process. The architecture of such a system and the analysis of its specific components are presented below.

A. Algorithm of the system operation

The general algorithm of the operation of an intelligent personal assistant that includes the implementation of a mental lexicon model (in the form of a knowledge base of an intelligent system built using the OSTIS Technology) for conducting a dialogue with the user can be described in the form of a sequence of the following actions.

- The user utters a speech message or command for the smart home system.
- Further, the acoustic module implemented within the framework of the OSTIS platform as an Agent of transition of speech into the semantic representation performs the procedure of semantic-acoustic analysis and transmits its results directly to the knowledge base of the system in the form of concept identifiers, thus displaying a natural language phrase in the semantic space of the computer "mental lexicon".
- Then the semantic analysis module, which includes the knowledge base of the system and the Message processing agent, Agent of decomposition of messages into atomic ones, Agents of allocation of entities and relations, Message classification agent and Agent of logical inference, parses, analyzes and interprets the received user message using information from the knowledge base (about the profiles of the system users, its current state, environmental factors, etc.). With the help of the Command classification agent and Agent of the control of executive devices, it executes a user command or a request.
- As a result, the response generation module, in which, based on the response structure in the knowledge base, the result of the work of the system is created in text form by means of the Message generation agent, which is then voiced by the system through the Agent of text-to-speech transition, which refers to the text-to-speech synthesis engine. As a result of the reasons for the typical implementation of the last module, consideration of its features is beyond the framework of reference of this article. The Message classification agent classifies the received message based on the rules present in the knowledge base.

Further, the features of the implementation of the two main modules, acoustic and semantic ones, will be considered in more detail to demonstrate how the designated problems of modeling the mental lexicon can be implemented.

B. Acoustic module

For speech analysis, a model based on a hybrid representation of a speech signal will be used, which allows representing any fragments of a speech signal of different nature of sound generation in the most adequate form [33]. Vocalized and non-vocalized signal fragments belong to separate parts of the model: periodic (harmonic) and aperiodic (noise) ones.

Mathematically, the main idea of the given model can be formalized in the following form:

$$s(n) = h(n) + r(n), \quad n = \overline{0,..,N-1} \tag{1}$$

where $s(n)$ is the input speech signal, $h(n)$ – a harmonic component, $r(n)$ – a noise component of the signal, n and N – the current signal sumple number and the total duration of the analysis fragment, respectively. The harmonic component can be represented by the following expression:

$$h(n) = \sum_{k=1}^{K} G_k(n) \sum_{c=1}^{C} A_k^c(n) \cos_k^c n + \phi_k^c(0)) \tag{2}$$

where G_k is an amplifier gain determined based on the spectral envelope, c – the number of sinusoidal components of the signal for each harmonic curve, A_k^c – the instantaneous amplitude of the c-th component of the k-th harmonic curve f_k^c and $\phi_k^c(0)$ – frequency and initial phase of the c-th component of the k-th harmonic curve, e_k – an actuating signal of the k-th harmonic curve. The amplitudes A_k^C are normalized to provide the sum of the harmonic energy equal to $\sum_{c=1}^{C}[A_k^c]^2 = 1$ for k = 1,...,K.

In this case, the aperiodic component is modeled over the entire frequency band, as it is demonstrated in the spectrum of a real speech signal [34]. This effect is reached by applying the technique of signal analysis through synthesis and diminution of the harmonic part from the original signal:

$$r(n) = \begin{cases} \max(s(n), h(n)) - h(n), & s(n) > 0 \\ \min(s(n), h(n)) - h(n), & s(n) < 0 \end{cases} \tag{3}$$

Thus, for one signal frame with the number m and the duration of N samples, a characteristic vector is formed, which includes the coefficients of the model $\mathbf{x_m} = [G_k, A_k^c, f_k^c, K, C]$. The acoustic image of a single word is a sequence of such characteristic vectors: $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_M})^T$.

It is proposed to evaluate the model parameters using the method of instantaneous harmonic analysis, which allows significantly increasing the accuracy of determining the parameters of the periodic component [63]. Its application allows obtaining a high temporal and frequency resolution of the signal as well as a clearer spectral pattern of the localization of energy at the corresponding frequencies (fig. 5) and, as a result, performing a more accurate assessment of the signal parameters (on average by 10–15%).

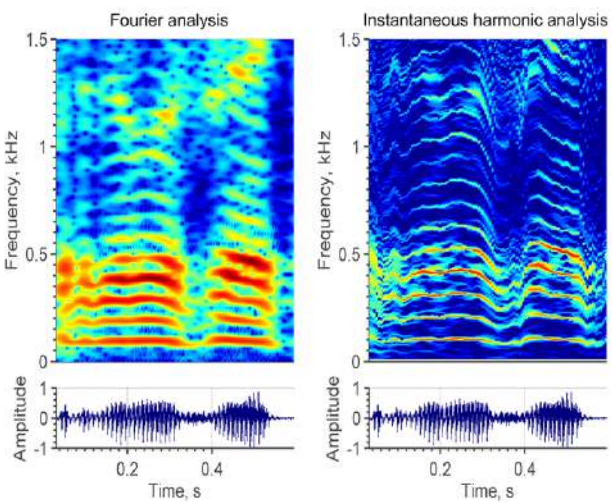


Figure 5. STFT- and IHA-based spectrograms

Unlike the classical methods of signal analysis used in modern speech recognition systems based on the determination of Mel-frequency cepstral coefficients (MFCC) or linear prediction coefficients (LPC) [64], [65], the method based on instantaneous harmonic analysis allows getting a high temporal and frequency resolution of the signal as well as a clearer spectral pattern of the localization of energy at the corresponding frequencies. Unlike classical methods, which are based on the discrete Fourier series transformation or the definition of the autocorrelation function of a signal on a short fragment, the method under consideration does not impose strict constraints related to adherence to the conditions of stationarity of the signal parameters on the analysis frame. At the same time, the parameters of the harmonic model, if necessary (for example, to describe the spectral envelope), can be relatively easily converted to other methods of representation, such as classical Mel-frequency cepstral or linear prediction coefficients.

Algorithms that implement the signal processing method described above based on instantaneous harmonic analysis have a reference implementation as part of the GUSLY audio signal analysis and synthesis framework [35].

Thus, after obtaining the parameters of the model, we can generate a spectrogram of the signal. A spectrogram is a visual representation of the frequency content of a signal as a function of time (fig. 6). The spectrogram of any audio signal can be considered unique, but this representation has too high a dimensionality to be used as a kind of fingerprint of the "acoustic image" associated with a specific node of the mental lexicon. Therefore, a more compact representation of the acoustic images of the mental lexicon is required. Similarly, this procedure is implemented in such services as, for example, the music recognition application "Shazam" [37].

The spectrogram of the signal of a phoneme, lexeme, word combination or even an entire phrase that sounds in speech can be considered its unique signature. Therefore, to determine whether two acoustic images are the same, it is possible to compare their spectrograms. However, the spectrogram is a rather large three-dimensional array (frequency, time, amplitude) and, therefore, requires a significant memory amount.

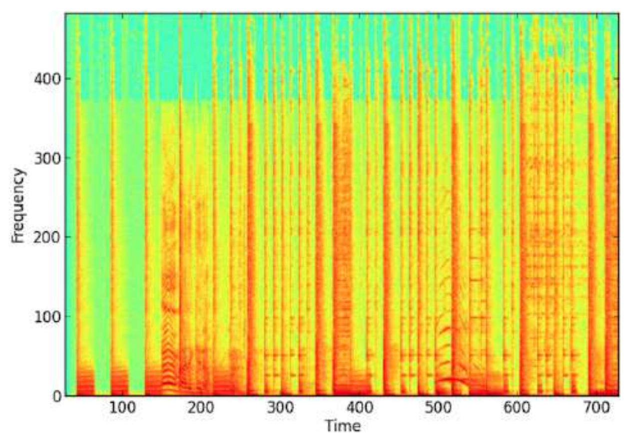


Figure 6. A spectrogram of the speech signal

Physical storage and computational comparison of unique signatures in the form of spectrograms for millions of nodes of a semantic network would be an unsolvable problem. Therefore, it is important to find methods that will allow highlighting only the most significant information from the spectrogram and finding a way to present it in tabloid form. One of the techniques of such compression involves the creation of what is called a "constellation map", i.e. an array of key points of the spectrogram, which is formed by finding local peaks in the signal spectrum (fig. 7).

The next step is to obtain a compressed representation of this array of points and create an acoustic image fingerprint that denotes an audible concept. We use a method, in which the frequency of a local peak is combined with the frequency of another local peak in its neighborhood and the time difference between the frequencies is calculated [36].

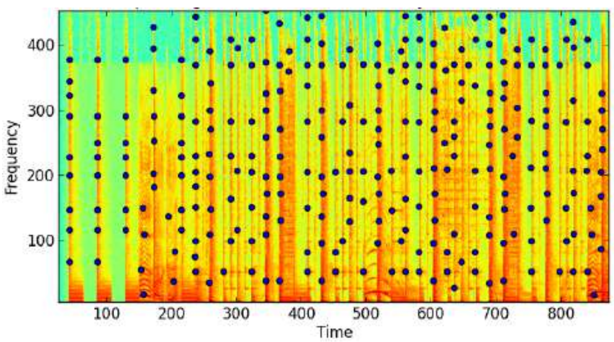


Figure 7. Highlighting of peaks on the spectrogram

For each local peak frequency (anchor), we have a set of the nearest frequencies (targets) and their time deltas. By forming sets of such triples of parameters for each peak frequency in this way, we can preserve the unique features of the analyzed signal fragment. The constellation map (fig. 8) is the information used to generate an acoustic image fingerprint by transmitting it to the hashing function. The hashing function receives data of indeterminate length and generates output data of fixed size (called a hash). In addition, hashing functions will always produce the same hash for the same input. The result of the hashing function is a sound fingerprint of the acoustic image of the spoken word, which we can compare with the standard associated with the node of the semantic network of the mental lexicon.

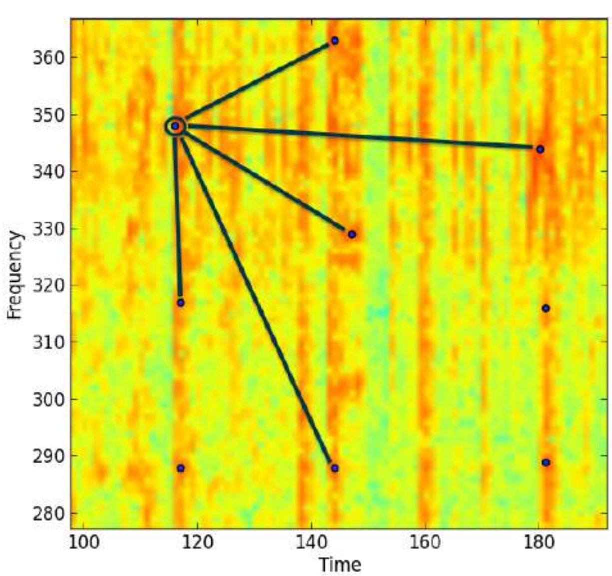


Figure 8. The constellation map

To determine the proximity between two images, we will use the Jaccard distance, which allows measuring the similarity between finite sets and is defined as the size of the intersection divided by the size of the union of these sets:

$$J = \frac{|H_sig \cap H_ref|}{|H_sig \cup H_ref|} = \frac{|H_sig \cap H_ref|}{|H_sig| + |H_ref| - |H_sig \cap H_ref|} \tag{4}$$

where J is the value of the Jaccard distance, H_sig – a hash of the acoustic image of the input signal, H_ref – a hash of the acoustic image of the standard associated with the network node.

This approach allows us to compare signals that can have different lengths, which is a very common phenomenon since words of a language can be and are often pronounced at different speeds. Another important property of the proposed algorithm is its low computational complexity, which allows it to be used in nearly real-time systems. This is especially true for creating speech assistants where the speed of the system response to user input is important.

C. Semantic module

As a technological basis for the implementation of the proposed semantic part of the system, it is proposed to use the OSTIS Technology [66]. The main advantages of the implementation of intelligent systems in the context of solving the problem of modeling the mental lexicon as a semantic network of language concepts have already been identified by us in the section "Problem definition". Let us focus on some practical aspects of the usage of the technology and its features that provide these advantages.

The systems built on the basis of the OSTIS Technology are called ostis-systems, respectively, the module for understanding speech messages, the prototype of which is considered in this article, will be built as a reusable component that will be integrated into various ostis-systems in the future, if necessary.

As a formal basis for encoding various information in the knowledge base, the SC-code [66] is used, the texts of which (sc-texts) are written in the form of semantic networks with a basic set-theoretic interpretation. The elements of such networks are called sc-elements (sc-nodes, sc-arcs). The focus of this work on the OSTIS Technology is due to its following main advantages:

- within the framework of this technology, unified means of representing various types of knowledge, including meta-knowledge, are proposed, which makes it possible to describe all the information necessary for analysis in one knowledge base in a unified manner [38];
- the formalism used within the framework of the technology allows specifying in the knowledge base not only concepts but also any files external from the point of view of the knowledge base (for example, fragments of a speech signal), including the syntactic structure of such files;
- the approach proposed within the framework of the technology to the representation of various types of knowledge [38] and models of their processing [29]

ensures the modifiability of ostis-systems, i.e., allows easily expanding the functionality of the system by introducing new types of knowledge (new systems of concepts) and new models of knowledge processing;
- the above advantages cumulatively make it possible to perform acoustic, syntactic and semantic analysis of messages in the same memory using unified processing tools, which, in turn, allows adjusting the analysis processes at any stage using various information from the knowledge base. In turn, the developed module for speech message understanding is itself built as an ostis-system and has an appropriate architecture.

D. Knowledge base

The basis of the knowledge base of any ostis-system (more precisely, the sc-model of the knowledge base) is a hierarchical system of subject domains and their corresponding ontologies. The upper level of the hierarchy of the part of the knowledge base related directly to speech assistants is shown below.

Voice assistant knowledge base
⇐ *section decomposition**:
{
- *Section. Subject domain of messages*
- *Section. Subject domain of dialogue*
  ⇐ *section decomposition**:
  {
  - *Section. Subject domain of dialogue control*
  - *Section. Subject domain of dialogue participants*
  }
- *Section. Subject domain of a smart home*
}

The knowledge base of the ostis-based system has already been partially described in [32], but it has been expanded and has undergone some changes, which will be discussed in more detail below.

Next, let us take a closer look at some of the above subject domains.

E. Subject domain of messages

Below is the upper level of the updated and refined message classification according to various criteria that do not depend on the subject domain.

atomic message
⇒ *decomposition**:
{
- *interrogative message*
- *imperative message*
- *declarative message*
}
⇒ *decomposition**:
{
- *message without an emotional coloring*
- *message with an emotional coloring*
}
⇒ *decomposition**:
{
- *message about the past*
- *message about the present*
- *message about the future*
}

An atomic message is a message that does not include other messages.

interrogative message
⊃ *information request message*

imperative message
⊃ *wish message*

declarative message
⇒ *decomposition**:
{
- *informational message*
- *neutral message*
  ⊃ *greeting message*
  ⊃ *valedictory message*
}

A neutral message is a declarative message that does not carry new information and does not confirm or deny formerly known information. An example of a neutral message is a message that contains information known to the system, but that is not an answer to a question asked to confirm/deny this information.

informational message
⇒ *decomposition**:
{
- *informing message*
- *message of denial of information*
- *message of confirmation of information*
}

message with an emotional coloring
⇒ *decomposition**:
{
- *message with a negative emotional coloring*
- *message with a neutral emotional coloring*
- *message with a positive emotional coloring*
- *message with an undefined emotional coloring*
}

A message with a negative emotional coloring is an atomic message that expresses a negative emotion; negative emotions include anger, fear, hate, fright, etc. A message with a neutral emotional coloring is an atomic message that expresses a neutral emotion; neutral emotions include curiosity, surprise, indifference, etc. A message with a positive emotional coloring is an atomic message that expresses a positive emotion; positive emotions include pleasure, exultation, love, etc. A message with an undefined emotional coloring is an atomic message that has an undefined emotional coloring, on the basis of which it is difficult to determine an emotion; this type of message can appear when a person demonstrates several, usually conflicting, emotions.

F. Subject domain of dialogue participants

To represent information about the participants of the dialogue, an appropriate model of the subject domain and ontology are created. The structure of this subject domain is shown below:

Subject domain of dialogue participants
⇒ *private subject domain**:
- *Subject domain of biography*
  ⇒ *private subject domain**:
  - *Subject domain of organizations*
  - *Subject domain of territorial entities*
  - *Subject domain of living beings*
  - *Subject domain of awards*
  - *Subject domain of education*
- *Subject domain of personal characteristics*
  ⇒ *private subject domain**:
  - *Subject domain of mental states*
    ⇒ *private subject domain**:
    - *Subject domain of emotions*
    - *Subject domain of mood*
  - *Subject domain of personality types*

The subject domain of biography contains means of describing factual data from the biography of the interlocutor, such as:

- participation in any organizations and their characteristics;
- relations that connect them with territorial entities (such as the *place of birth**);
- information about relatives and marital status;
- awards received (including honorary titles);
- educational qualifications.

The subject domain of personal characteristics contains means of describing the mental state of the interlocutor as well as their personality type. The need to store this information is due to one of the goals of the system – to maintain a good mood in the interlocutor, which results in the need to take into account their current state during the dialogue. Within the framework of this subject domain, the following classes of mental states are distinguished:

mental state
⇐ *decomposition**:
{• *superficial mental state*
• *deep mental state*
}
⇐ *decomposition**:
{• *conscious mental state*
• *unconscious mental state*
}
⇐ *decomposition**:
{• *personality-related mental state*
• *mental state caused by the situation*
}
⇐ *decomposition**:
{• *positive mental state*
• *negative mental state*
• *neutral mental state*
}
⇐ *decomposition**:
{• *short-term mental state*
• *long-term mental state*
• *medium duration mental state*
}

Figure 9 shows a fragment of the description in the knowledge base of a specific user known to the system.

The above description contains both long-term information about the user, which will be saved after the end of the dialogue (gender, name, etc.) and short-term one, which can be updated with each new dialogue – information about the age, date of the last visit, mood, etc.

Each element of the *beginning* set is a class of temporary entities (i.e., entities with temporal characteristics: duration, initial time, final point, etc.), which have the same moment of the beginning of their existence. The concrete value of this parameter can be either an exact value or a discreet/interval one.

Each element of the *completion* set is a class of temporary entities that have the same final moment of their existence (the moment of the end of existence). The specific value of this parameter can be either an exact value or a discreet/interval one.

G. Subject domain of a smart home

For the usage of the system within the framework of the "smart home", a corresponding subject domain was introduced, which contains the means of describing both the building itself and the devices used in it as well as their operation status (the state of lighting, the presence of humans, etc.).

H. Problem solver

The problem solver of any ostis-system (more precisely, the sc-model of the ostis-system problem solver) is a hierarchical system of knowledge processing agents in semantic memory (sc-agents) that interact only by specifying the actions they perform in the specified memory.

In comparison with **??**, the structure of the problem solver has been revised to make it possible to use it within the framework of the smart home system. The top level of the updated hierarchy of agents of the problem solver of the speech assistant in the SCn-language looks like follows:

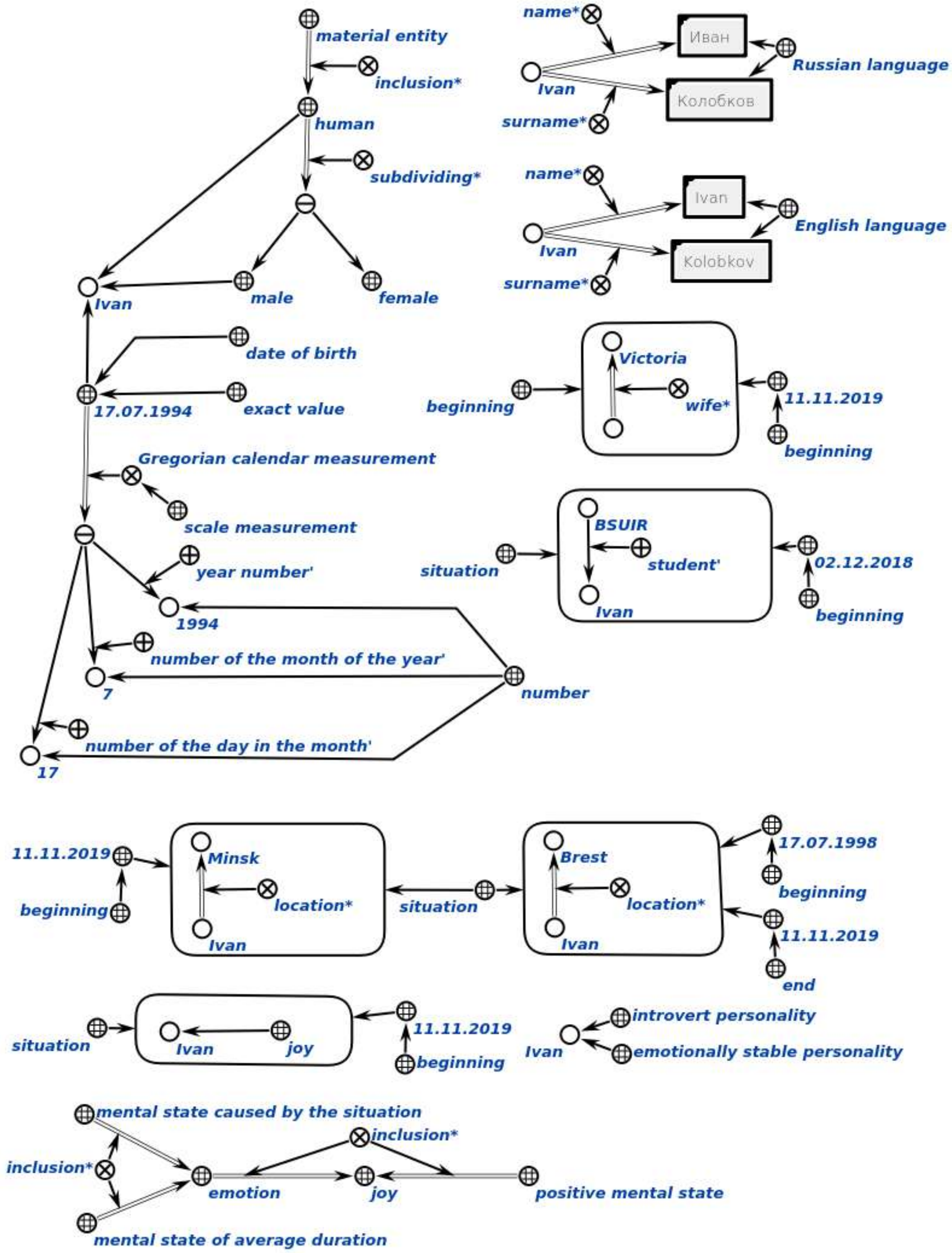


Figure 9. The first part of the model description

Speech assistant problem solver

⇐ *decomposition of an abstract sc-agent**:

{
- *Agent of logical inference*
- *Message processing agent*
  - ⇐ *decomposition of an abstract sc-agent**:
    {
    - *Agent of decomposition of a non-atomic message into atomic ones*
    - *Agent of allocation of entities*
    - *Agent of allocation of relations*
    - *Message classification agent*

    }
- *Smart home control agent*
  - ⇐ *decomposition of an abstract sc-agent**:
    {
    - *Command classification agent*
    - *Agent of the control of executive devices*

}
- *Message generation agent*
- *Agent of speech-to-text transition*
- *Agent of text-to-speech transition*

}

A *smart home control agent* is designed to distinguish the necessary actions depending on the events that have occurred and their performance.

The *Message processing agent* is designed to distinguish the meaning of input natural-language messages. Depending on the result of processing the received message, in particular, its class, in the future, either only the Message generation agent can be called or the Smart home control agent can be called in addition to it. In the second case, the receiving of a message that is a command (for example, to turn on lighting) is processed by the Smart home control agent as well as any other event (for example, user recognition in the frame of any of the cameras or the fall of a certain time of day). Thus, the Message generation agent also serves to generate notifications about the performance of actions.

The Agent of logical inference applies logical rules and is used by various agents, including the Message generation agent and the Smart home control agent.

The knowledge base contains the specification of these agents. As an example, the specification of the Agent of text-to-speech transition is shown in figure 10.

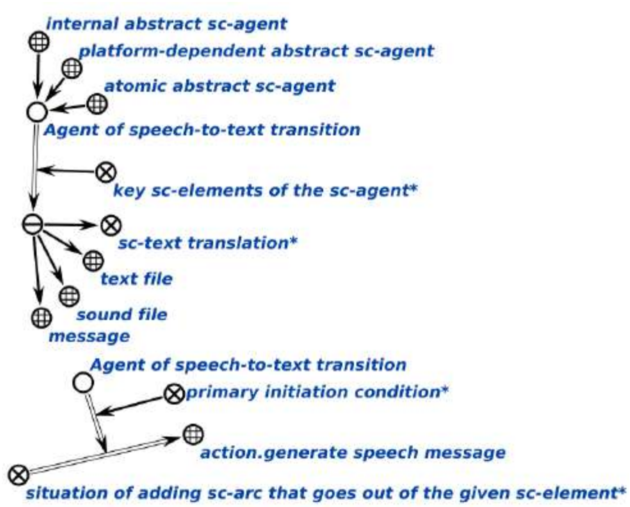


Figure 10. A fragment of the specification of the Agent of text-to-speech transition.

VII. Example of working

As an example, it is possible to give a scenario that includes the following stages:

- the user comes to the front door, the system recognizes them and unlocks the door;
- the user enters the living room, the system turns on the light in this room;
- the user instructs to increase the brightness by 30%.

At the first stage, the following rules are applied: first, according to the rule shown in figure 11, the Smart home control agent opens the door, then, according to the rule shown in figure 12, the Message generation agent generates a message notifying about the door opening. In this case, the text of the message generated by this rule is created according to the template associated with this rule by the *message text pattern** relation.

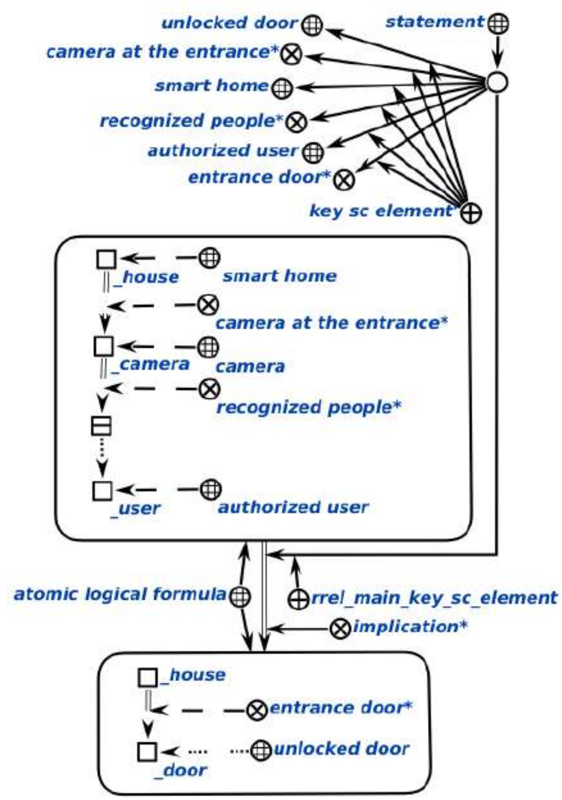


Figure 11. A rule for performing the "opening of the door" action.

To turn on the lighting in the room, if a user has entered it, the rule shown in figure 13 is applied. The message that notifies about the performed action is generated according to the rule similar to what was shown in figure 12.

An example of a rule that takes into account a user message that has been received (and processed by the message processing agent) can be the rule shown in figure 14.

VIII. Conclusion

In the article, one of the possible approaches to the implementation of the psycholinguistic concept of the mental lexicon is proposed, which plays an essential role in the process of human communication as part of intelligent personal assistants. In our opinion, the research of the possibilities of translation of this structure of human consciousness as part of intelligent systems is an important area of research in the field of AI. From a theoretical

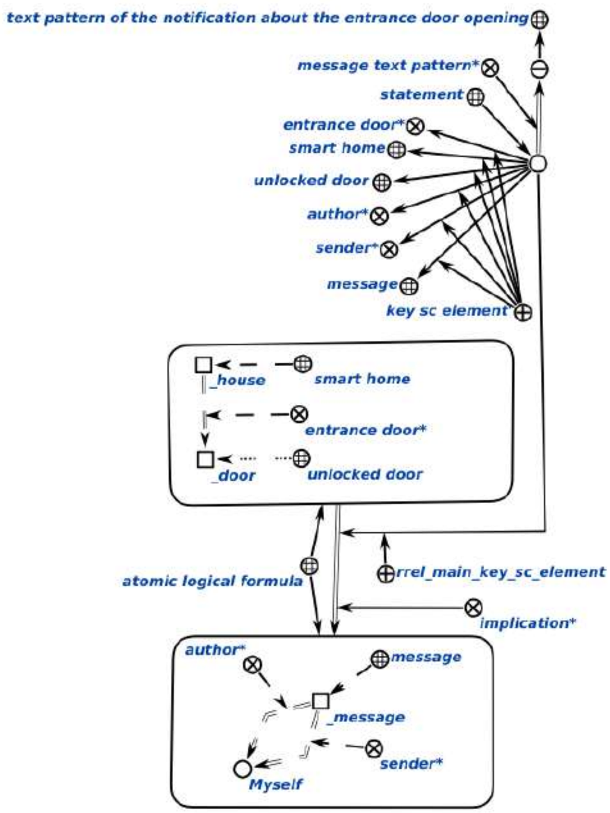


Figure 12. The rule for generating a message that notifies about the opening of the door.

point of view, this will increase the level of understanding of the processes that occur in the human consciousness when perceiving and understanding speech messages, and from a practical point of view, it will make it possible to create intelligent systems with a speech interface that have new qualitative capabilities for understanding speech messages, flexibility and adaptability, the ability to learn directly in the process of interaction with the user.

It is proposed to model the mental lexicon using the semantic network appliance, intelligent agents and knowledge bases and their practical implementation within the framework of the OSTIS Technology, which includes modern implementations of all these aspects of the semantic part of the system. To model the speech recognition process, it is proposed to use the method of semantic-acoustic analysis, which allows direct transiting from the space of acoustic images of words to the space of semantic network concepts that correspond to this image. Thus, a significant part of information processing can be carried out immediately in the semantic domain, bypassing the preliminary stages of speech-to-text transformation that are characteristic of all modern systems.

Thus, the implementation of the intelligent personal assistant system using the mental lexicon model and its implementation within the framework of the OSTIS Technology as well as an approach based on semantic-acoustic speech analysis will give it new useful properties, ensure additional flexibility and adaptability. It will make it possible to overcome some of the limitations that are characteristic of the solutions currently presented on the market and will allow intelligent systems with a speech interface to reach a qualitatively new level.

Acknowledgment

The authors would like to thank the Departments of Intelligent Information Technologies, Control Systems and Electronic Computing Facilities of the Belarusian State University of Informatics and Radioelectronics for the help and valuable comments.

References

[1] Gonia J. et al. The Mental Lexicon: Core Perspectives. Gonia Jarema, Gary Libben. BRILL, Jul 1, 2007. Language Arts & Disciplines. 246 p.
[2] Eisner, F., McQueen J. M. Speech perception. Hoboken, Wiley, 2018, Vol. 3, pp. 1–46.
[3] Hoy M. B. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. Medical reference services quarterly, 2018, Vol. 37, No. 1, pp. 81–88.
[4] Global Voice Assistant Market By Technology, By Application, By End User, By Region, Competition, Forecast & Opportunities, 2024. Available at: https://www.businesswire.com/news/home/20190916005535/en/Global-Voice-Assistant-Market-Projected-Grow-1.2. (accessed 2021, May)
[5] MacTear M., Callejas Z., Griol D. The Conversational Interface: Talking to Smart Devices. Springer, 2016, 422p.
[6] Hannun A. et al. Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567, 2014.
[7] Kaldi ASR. Available at: https://kaldi-asr.org. (accessed 2021, May).
[8] VOSK Offline Speech Recognition API. Available at: https://alphacephei.com/vosk/. (accessed 2021, May).
[9] Biswas M. Wit. ai and Dialogflow. Beginning AI Bot Frameworks. Apress, Berkeley, CA, 2018, pp. 67–100.
[10] Bocklisch T. et al. Rasa: Open source language understanding and dialogue management. arXiv preprint arXiv:1712.05181, 2017.
[11] Ham D. et al. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 583–592.
[12] Floridi L., Chiriatti M. GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 2020, vol. 30, No. 4, pp. 681–694.
[13] Oord A. et al. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.
[14] Wang Y. et al. Tacotron: Towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135, 2017.
[15] Stan A., Lőrincz B. Generating the Voice of the Interactive Virtual Assistant. Virtual Assistant, IntechOpen, 2021.
[16] Deriu J. et al. Survey on evaluation methods for dialogue systems. Artificial Intelligence Review. 2021, vol. 54, No. 1, pp. 755–810.
[17] Treisman, A. M. Contextual cues in selective listening. Quarterly Journal of Experimental Psychology 12, 1960, pp. 242–248.
[18] Oldfield, R. C. Things, words and the brain. Quarterly journal of experimental psychology 18, no. 4, 1966, pp. 340–353.
[19] Kovács L. et al. Networks in the mind–what communities reveal about the structure of the lexicon. Open Linguistics. 2021, vol. 7, No. 1, pp. 181–199.
[20] Cuyckens, H., Dirven R., Taylor J. R. Cognitive approaches to lexical semantics. Walter de Gruyter, 2009, Vol. 23.
[21] Zhang, X., Nannan, L. I. U. Exploring the Second Language Mental Lexicon With Word Association Tests. Cross-Cultural Communication. 2014, Vol. 10, No. 4, pp. 143–148.

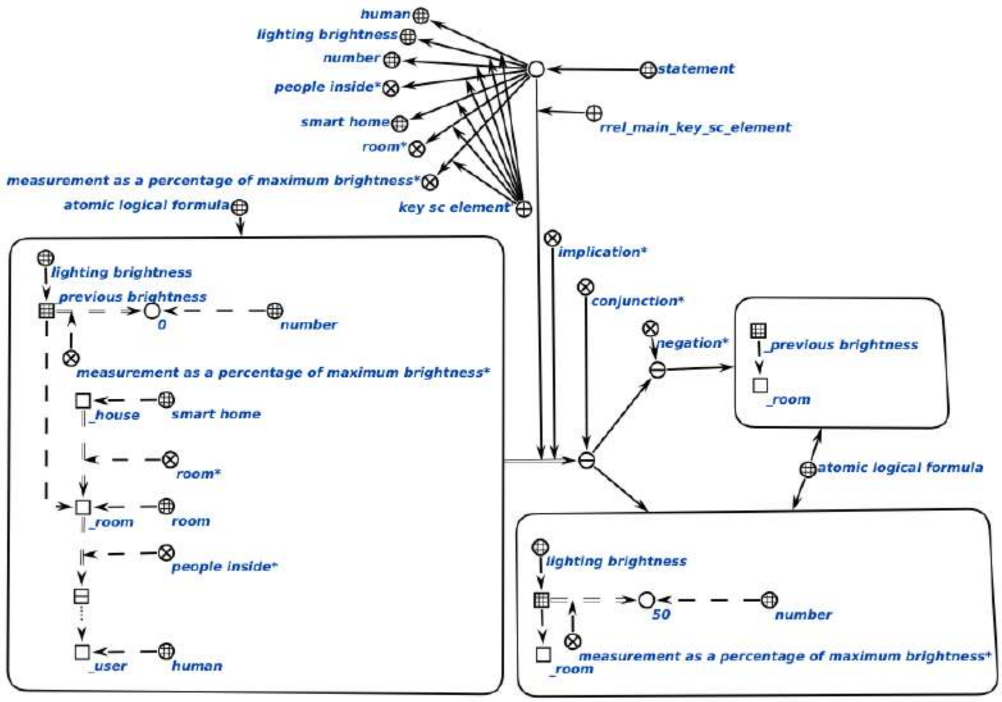


Figure 13. The rule for turning on the lighting in the room when the user enters it.

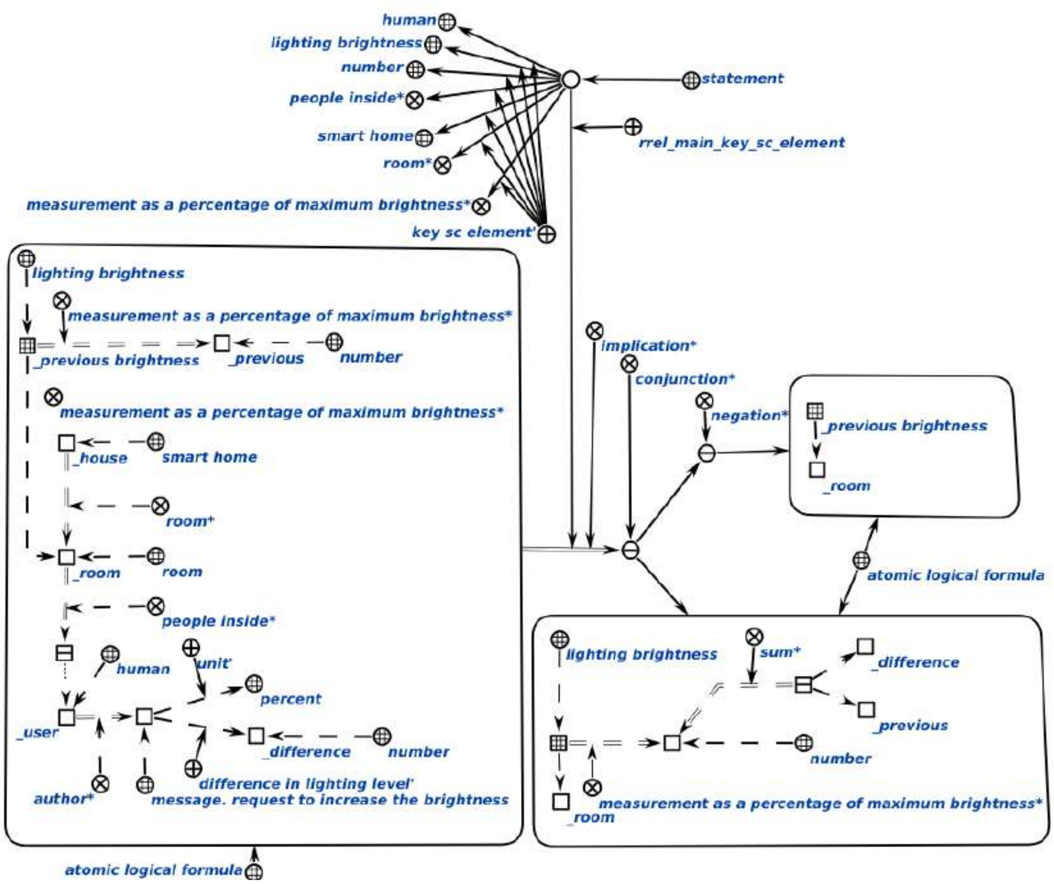


Figure 14. The rule for adjusting the brightness of lighting according to the received user message.

[22] Levelt, W. J. M. et al. Levelt, W. J., Schriefers, H., Vorberg, D., Meyer, A. S., Pechmann, T., & Havinga, J. The time course of lexical access in speech production: A study of picture naming. Psychological review, 1991, Vol. 98, No. 1, 122 p.

[23] Baayen, R. H. Experimental and psycholinguistic approaches to studying derivation. Handbook of derivational morphology. 2014, pp. 95–117.

[24] Pirrelli, V. et al. Psycho-computational modelling of the mental lexicon. Word Knowledge and Word Usage. De Gruyter Mouton, 2020, pp. 23–82.

[25] Gaskell, G., Marslen-Wilson W. Inference processes in speech perception. Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society. Routledge, 2019, pp. 341–345.

[26] Helfer, K. S., Merchant, G. R., Wasiuk, P. A. Age-related changes in objective and subjective speech perception in complex listening environments. Journal of Speech, Language, and Hearing Research. 2017, Vol. 60, No. 10, pp. 3009–3018.

[27] Sowa, J. F. Principles of semantic networks: Explorations in the representation of knowledge. Morgan Kaufmann, 2014, 594 p.

[28] Han, J. et al. Semantic networks for engineering design: A survey. Proceedings of the Design Society. 2021, Vol. 1, pp. 2621-2630.

[29] Shunkevich, D. V. Ontology-based design of knowledge processing machines. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]. Minsk, BSUIR, 2017, pp. 73–94.

[30] Zahariev V.A., Azarov E.S., Rusetski K.V. An approach to speech ambiguities eliminating using semantically-acoustical analysis. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]. Minsk, BSUIR, 2018, pp. 211–222.

[31] Zahariev V. A., Lyahor T., Hubarevich N., Azarov E.S. Semantic analysis of voice messages based on a formalized context. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]. Minsk, BSUIR, 2019, pp. 103–112.

[32] Zahariev V., Shunkevich D., Nikiforov S. and Aarov E. "Intelligent voice assistant based on open semantic technology," in-OpenSemantic Technologies for Intelligent System, ser. Communicationsin Computer and Information Science, V. Golenkov, V. Krasno-proshin, V. Golovko, and E. Azarov, Eds. Springer, Cham, 2020,pp. 121–145.

[33] Stylinou, Y. Applying harmonic plus noise model in concatenative speech synthesis. IEEE Trans. on Speech and Audio Processing. 2001, Vol. 9, No. 1, pp. 21–29.

[34] Serra, X. A system for sound analysis / transformation / synthesis based on deterministic plus stochastic decomposition. PhD thesis. Stanford. 1989, 178 p.

[35] Azarov E., Vashkevich M., Petrovsky A. Instantaneous harmonic representation of speech using multicomponent sinusoidal excitation. INTERSPEECH 2013: proceedings of 12th Annual Conference of the International Speech, Lyon, France, 2013, pp. 1697–1701.

[36] Lerch, A. An introduction to audio content analysis: Applications in signal processing and music informatics. Wiley-IEEE Press, 2012, 259 p.

[37] Froitzheim, S. A Short Introduction to Audio Fingerprinting with a Focus on Shazam. Available at: https://hpac.cs.umu.se/teaching/sem-mus-17/Reports/Froitzheim.pdf (accessed 2021, May)

[38] Davydenko I.T. Ontology-based knowledge base design. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]. Minsk, BSUIR, 2017, pp. 57–72.

[39] Leont'ev A. A. Osnovy psikholingvistiki [Fundamentals of psycholinguistics]. Smysl, NPF Smysl, 2005, 310p. (in Russian)

[40] Vygotskii, L. C. Myshlenie i rech'. Psikhologicheskie issledovaniya. [Thinking and speaking. Psychological research] – Natsional'noe obrazovanie, 2015, 368 p. (in Russian)

[41] Chernigovskaja T. V. Chto delaet nas ljud'mi: pochemu nepremenno rekursivnye pravila? Razumnoe povedenie i jazyk [What makes us human: why recursive rules necessarily? Reasonable behavior and language]. 2008, No. 1, pp. 395–412. (in Russian)

[42] Chernigovskaja T. V. Jazyk, mozg i komp'juternaja metafora [Language, Brain and Computer Metaphor]. Chelovek [Human]. 2007, Vol. 2, pp. 63–75. (in Russian)

[43] Chernigovskaja T. V. Eshhe raz o mozge i semiozise: mozhno li najti tochku v nejrosetjah? [Once again about the brain and semiosis: is it possible to find a point in neural networks?] Voprosy filosofii [Philosophy questions]. 2021, No. 6, pp. 5–13. (in Russian)

[44] Zalevskaja A. A. Slovo v leksikone cheloveka: psiholingvisticheskie issledovanija [A word in the human lexicon: psycholinguistic research]. M. M. Kopylenko Eds. Voronezh, VGU, 1990, 204 p. (in Russian)

[45] Zalevskaja A. A. Mental'nyj leksikon s pozicij raznyh podhodov [Mental'nyj leksikon s pozicij raznyh podhodov]. Aktual'nye problemy sovremennoj lingvistiki [Actual problems of modern linguistics]. L. N. Churilina Eds. 4-e izd., 2009, pp. 311–327. (in Russian)

[46] Popova Z. D. et al. Popova i Polevye struktury v sisteme jazyka [Popova and Field Structures in the Language System]. Z. D. Popova Eds. Voronezh, 1989, 198 p. (in Russian)

[47] Zalevskaja A. A. Psiholingvisticheskie issledovanija. Slovo. Tekst: izbran. trudy [Psycholinguistic research. Word. Text: selected. labors]. Moskva, Gnozis, 2005, 542 p. (in Russian)

[48] Vencov A.V., Kasevich V.B. Problemy vosprijatija rechi [Speech perception problems]. Moskva, Izdatel'skaja gruppa URSS, 2003, 240 p. (in Russian)

[49] Saharnyj L. V. Psiholingvisticheskie aspekty teorii slovoobrazovanija: ucheb. posobie [Psycholinguistic aspects of the theory of word formation: textbook. allowance]. L., LGU, 1985, 408 p. (in Russian)

[50] Stern A. S. Perceptivnyj aspekt rechevoj dejatel'nosti: jeksperimental'noe issledovanie [Perceptual aspect of speech activity: an experimental study]. Sankt-Peterburg, Izd-vo S.-Peterb. un-ta, 1992, 236 p. (in Russian)

[51] Karaulov J. N. Aktivnaja grammatika i associativno-verbal'naja set' [Active grammar and the associative-verbal network]. M., Institut russkogo jazyka RAN, 1999, 180 p. (in Russian)

[52] Zalevskaja A. A. Vvedenie v psiholingvistiku : ucheb. dlja studentov vuzov, obuch. po filol. spec. [Introduction to psycholinguistics: textbook. for university students, training. by philol. specialist.] M., RGGU, 1999, 381 p. (in Russian)

[53] Prokopenja V. K., Sljusar' N. A., Petrova T. E., Chernova D. A. & Chernigovskaja T. V. Jeksperimental'nye issledovanija grammatiki: ustanovlenie anaforicheskih otnoshenij v processe recheponimanija [Experimental studies of grammar: the establishment of anaphoric relations in the process of comprehension]. Voprosy jazykoznanija [Linguistic issues]. 2018, No. 1, pp. 76–90. (in Russian)

[54] Golenkov V. V. Principy postroenija massovoj semanticheskoj tehnologii komponentnogo proektirovanija intellektual'nyh sistem [Principles of building mass semantic technology for component design of intelligent systems]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]. Minsk, BGUIR, 2011, pp. 21–58. (in Russian)

[55] Skorohod'ko J. F. Semanticheskie seti i avtomaticheskaja obrabotka teksta [Semantic networks and automatic word processing]. Kiev, Nauk. dumka, 1983. (in Russian)

[56] Skrjegg G. Semanticheskie seti kak modeli pamjati [Semantic networks as memory models]. Novoe v zarubezhnoj lingvistike [New in foreign linguistics]. No. 12, M., Raduga, 1983, pp. 228–271. (in Russian)

[57] Peirce Ch. S. Logicheskie osnovanija teorii znakov [Logical foundations of the theory of signs]. SPb., Izd-vo Sankt-Peterb. un-ta: Aleteja. 2000, 352 p. (in Russian)

[58] Sapatyj P.S. Ob jeffektivnosti strukturnoj realizacii operacij nad semanticheskimi setjami [Ob jeffektivnosti strukturnoj realizacii operacij nad semanticheskimi setjami]. Tehn. kibernet. 1983, No. 5, pp. 128–134. (in Russian)

[59] Shunkevich, D. V. Vzaimodejstvie asinhronnyh parallel'nyh processov obrabotki znanij v obshhej semanticheskoj pamjati [Interaction of Asynchronous Parallel Knowledge Processing Processes in Shared Semantic Memory]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open se-

mantic technologies for intelligent systems]. Minsk, BGUIR, 2016, pp. 137–144. (in Russian)

[60] Harlamov A. A. Semanticheskie seti kak formal'naja osnova reshenija problemy integracii intellektual'nyh sistem. Formalizm avtomaticheskogo formirovanija semanticheskoj seti s pomoshh'ju preobrazovanija v mnogomernoe prostranstvo [Semanticheskie seti kak formal'naja osnova reshenija problemy integracii intellektual'nyh sistem. Formalizm avtomaticheskogo formirovanija semanticheskoj seti s pomoshh'ju preobrazovanija v mnogomernoe prostranstvo]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]. Minsk, BGUIR, 2011, pp. 87–96 (in Russian)

[61] Golenkov V. V., Guljakina N. A. Strukturizacija smyslovogo prostranstva [Structuring the semantic space]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]. Minsk, BGUIR, 2014, pp. 65–78. (in Russian)

[62] Golenkov V. V., Guljakina N. A. Intellektual'nye sistemy. Problemy i perspektivy [Intelligent systems. Problems and Prospects]. Informacionnye tehnologii i sistemy 2016 (ITS 2016). Minsk, BGUIR, 2016, pp. 13–20. (in Russian)

[63] Azarov I.S., Petrovskii A.A. Mgnovennyi garmonicheskii analiz: obrabotka zvukovykh i rechevykh signalov v sistemakh mul'timedia [Instant harmonic analysis: processing of sound and speech signals in multimedia systems]. LAP Lambert Academic Publishing, Saarbrucken. 2011, 163 p. (in Russian)

[64] Aificher E., Dzhervis B. Tsifrovaya obrabotka signalov: prakticheskii podkhod [Digital Signal Processing: A Practical Approach]. 2-e izd. [Digital signal processing: a practical approach. 2nd ed.]. M., Williams, 2004, 992 p. (in Russian)

[65] Rabiner L. R., Schafer R. W. Cifrovaja obrabotka rechevyh signalov [Digital processing of speech signals]. M.V. Nazarova, Ju. N. Prohorova Eds. M., Radio i svjaz', 1981, 496 p. (in Russian)

[66] Golenkov V. V., Guljakina N. A. Semanticheskaja tehnologija komponentnogo proektirovanija sistem, upravljaemyh znanijami [Semantic technology of component design of knowledge-driven systems]. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]. Minsk, BGUIR, 2015, pp. 57–78. (in Russian)

Анализ диалоговой речи на основе формализованного представления ментального лексикона

Захарьев В.А., Никифоров С.А., Азаров И.С.

В работе предложен один из возможных подходов по реализации психолингвистического концепта ментального лексикона, играющего существенную роль в процессе человеческого общения, в составе интеллектуальных речевых ассистентов. Изучение возможностей воплощения данной структуры человеческого сознания в составе интеллектуальных систем, по-нашему мнению, является важным направлением исследований в области ИИ. Подобные исследовании с теоретической точки зрения позволят повысить уровень понимания процессов происходящих в человеческом сознании при восприятии и понимании речевых сообщений, а с практической точки зрения, позволят создавать интеллектуальные системы с речевым интерфейсом обладающие новыми качественными возможностями понимания речевых сообщений, гибкостью и адаптивностью, способностью обучаться непосредственно в процессе взаимодействия с пользователем.

Предлагается осуществлять моделирование ментального лексикона с применением аппарата семантических сетей, интеллектуальных агентов и баз знаний, и их практического воплощения в рамках технологии OSTIS, включающей в себя современные реализации всех данных аспектов смысловой части системы. Для моделирования процесса распознавания речи предлагается использовать метод семантико-акустического анализа, позволяющий осуществить прямой переход из пространства акустических образов слов, в пространство понятий семантической сети, соответствующих данным образом. Таким образом существенная часть обработки информации может вестись сразу в семантической области миную предварительные этапы преобразования речи в текст.

Это позволит преодолеть некоторые ограничения, характерные для представленных в текущий момент времени на рынке решений, позволит обеспечить выход интеллектуальных систем с речевым интерфейсом на качественно новый уровень.

Received 28.05.2021

Ontological Approach for Generating Natural Language Texts from Knowledge Base

Longwei Qian, Wenzu Li
Department of Intelligent Information Technology
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
qianlw1226@gmail.com, wzzggml@gmail.com

***Abstract*—The computer systems are expert in dealing with structured data. However, when computer systems attempt to impart information to the end-users, generating natural language text that expresses structured data is a significant challenge. Currently graphical knowledge representations as a kind of forms to represent structured data are gradually becoming universal in computing. In this paper, we present a unified semantic model for generating fluent, multi-sentence, appropriate natural language text (e.g., Chinese language text) from knowledge base to the end-users. This article describes the development of semantic model for natural language generation, and the optional linguistic ontologies which may be used in the processing of generation. The main novelty is that it is possible to integrate different approaches and linguistic knowledge to generate natural language text from the structured data of the computer systems represented in graph form. For the ordinary end-users it will be an easier access to the information in the computer systems.**

***Keywords*—natural language generation, ontology, knowledge-driven, knowledge base, OSTIS**

I. Introduction

A. Objective and Relevance of the Work

The goal of this article is to use ontology-based approach as a basis to develop a unified semantic model for generating natural language text from the knowledge base. Generating natural language text is considered as a part of natural language interface that is based on the technology of ontology-based design of intelligent system user interface [1]. The technology of ontology-based design of intelligent system user interface itself is a part of Open Semantic Technology for Intelligent Systems Design (OSTIS Technology). Natural language interface focuses on processing of natural language text, taking into account methods and principles of user interface design. The proposed approach involves the development of natural language interface formal ontology. One of the advantages of this technology is the possibility to introduce various knowledge for generating natural language text.

Natural language processing is considered as a cognitive problem in the field of artificial intelligence. Many researchers have tried to address the problem of natural language understanding, in contrast, there are not many researches on issues of natural language generation (NLG). Although the number of works in NLG is fewer, their content varies due to the difficulty to precisely define NLG. Now everybody agrees that the output of NLG should be natural language text, but the exact form of the input can vary substantially. Generally speaking, the input can be divided into unstructured textual forms, semi-structured textual forms and structured forms (e.g., tabular data, knowledge base). According to this point, the machine translation, text summary, data-text generation, as well as generation from visual input such as image or video entirely are instances of NLG. The difficulties in terms of NLG usually come from following two aspects:

- Analysis and processing of the input information forms;
- Analysis of features of the target natural language text that was generated by intelligent system.

Nowadays the application of intelligent system is ubiquitous. One of the most important research directions for designing intelligent system is the development of knowledge-based system, in particular after the concept of knowledge graph was proposed by Google [2]. In order that this kind of intelligent system can interact with users more conveniently, it's necessary for the system to have the capacity to understand the knowledge base and generate grammatically reasonable natural language text for the users. It had become possible with the increase of computing power and the proposal of the novel models. However, due to the various graphical structure and the optional multilinguality for the output, the generation of natural language text is still an open challenge.

B. Problems, Need to be Solved

Firstly, our goal is the development of natural language interface. Secondly, there are lots of models, methods and means to generate natural language text from structured data. They are successfully applied as components of natural language interface of intelligent system. Nevertheless, the following problems still need to be considered:

- Absence of unification of development process of natural language interface, thus development process of natural language interface cannot be achieved in parallel, as well as we cannot reuse the already developed component;
- The difficulties in analysis and processing of knowledge representation structure. The knowledge is not stored discretely in the intelligent system; it is interconnected to form a structured data in the system;
- There is no ability to use unified tools to represent various kinds of knowledge (e.g., the linguistic knowledge), which is necessary to provide useful information for NLG;
- Developed problem solvers are not flexible at each stage of NLG, e.g., for generating text in a new language it's difficult to change the already existing components or to extend NLG component to adapt.

The relevance of these problems will be explained in detail by analyzing related work to natural language generation.

C. *Analysis of related work for natural language generation*

In this paper, we focus on the NLG part of natural language interface. The NLG part can be considered as a separated system, or as a component of user interface. But it's difficult to achieve the development of separated NLG system that can be reused into the component of user interface due to absence of unified principle for user interface design. The NLG part in this paper tends to generate natural language text from structured form, in particular, from knowledge base.

Early in the application domain, the successful NLG system includes the weather reporting, "robo-journalist" and so on, which convert the tabular data or data of information box into reasonable natural language text by filling placeholders in a predefined template text [3]. In this situation, the text generated on the base of these predefined template is very simple and inflexible. The rule-based approaches convert data into resulted text by a serious of grammar and heuristic rules. The factors influencing these approaches to generate natural language text are the linguistic rules. For analysis of natural language, linguists proposed many linguistic theories that focus on interpreting linguistic formalism, for example, the dependency grammar is used for the interpretation of the syntactic structure. However these approaches lack supporting of unified basis for representing various linguistic knowledge. Moreover, traditional rule-based approaches focus not on the semantic of natural language text, but rather on the syntax.

There is a classic pipeline architecture [4] for NLG. Based on this pipeline, generally, the six tasks are frequently found in NLG system (Fig. 1). In fact, the classic pipeline architecture can be considered as the modular approach to solve the NLG problem. Different modules in the pipeline incorporate different subsets of the tasks described above. However, the biggest problem for the ordering of the modular approach is the generation gap [5] that refers to the mistakes of early tasks in the pipeline passed further downstream.

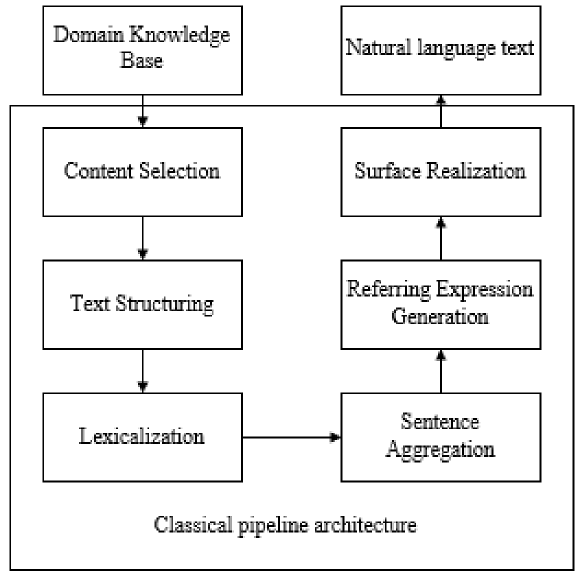


Figure 1: Classic pipeline of natural language generation

The pipeline continues influence on alternative architectures that were proposed in recent years for NLG. The proposed approaches often end up blurring the boundaries between modules. These approaches tend to emphasize statistical methods that is data-driven. From the pipeline perspective, there are three simplified steps:

- content selection;
- content planning;
- surface realization.

In the subtask of SemEval AMR-to-English generation, the abstract AMRs to be converted into syntactic structures by a symbolic generator, then the syntactic structures are linearized with an off-the-shelf tools (e.g. statistical linearizers) [6].

With the advent of deep learning, the most influential architecture for NLG is the Encoder-Decoder [7]. The encoder encodes various kinds of input (e.g., natural language text, structured data, image, video and knowledge base) into a low dimension vector representing the semantic of the input. The decoder generates natural language text from the vector embedding. In practice, the knowledge representation languages like RDF and others are widely used as a kind of input of various modern neural generation models for knowledge-based system constructed by the W3C standards. Currently the

W3C standards are widespread used for development of knowledge-based system. The standard to specify knowledge is RDF, a formal language based on semantic networks. The WebNLG [8] is a project oriented on developing technologies that give humans easy access to machine-readable data of Web, usually this data is in form of RDF. The WebNLG challenge tends to develop neural generator by deep learning models. The training data of WebNLG challenge consists of Data/Text pairs where the data is a set of triples extracted from DBpedia [9] and the text is a verbalization of these triples. However the neural generators usually ignore the distinction between text planning and realization that causes difficulties in controlling over the generated text structure [10]. Another problem is dataset acquiring, because the high quality aligned training data is the core for neural generator. Currently the WebNLG supports the English and Russian datasets. There are also high quality Chinese knowledge bases, such as CN-DBpedia [11], zhishi.me [12] and others, but the aligned Data/Text pairs datasets are difficult to obtain.

Nowadays the mainstream methods consist in applying the neural network model to achieve NLG with the help of the high quality dataset, recently rare works on researching the ruled-based method for NLG. The system NaturalOWL [13] proposed to construct generation resource to verbalize OWL. The system in fact is a symbolic generator; manually constructed resources of this system determine the quality of the generated text. The single model cannot solve the efficiency and quality of generated texts simultaneously, especially the quality dataset is absent for certain languages. The integration of various methods and the execution efficiency of various methods is the main problems for NLG.

As can be noted, the above-mentioned approaches can solve partial problems for development of NLG part of natural language interface. The effective integration of different approaches is still unsolved. Through the discussion of related work, for developing compound NLG part of natural language interface, the integrated approaches will be needed to solve the various problems of NLG.

D. The Proposed Approach

We propose to use ontological approach to develop the unified representation for linguistic ontologies and to design problem solvers having ability to integrate various approaches for NLG solution within OSTIS Technology framework [14]. In this paper, natural language generation is developed as a component of natural language interface that is a part of user interface of the ostis-system. The ostis-system is a hybrid intelligent system being developed on OSTIS Technology. The OSTIS Technology is aimed at knowledge processing and various knowledge presentation for intelligent systems, it's focused on the development of knowledge-driven computer systems. Each ostis-system consists of sc-model of knowledge base, sc-model of problem solver and sc-model of user interface. The sc-model of knowledge base is based on several basic principles (e.g. the hierarchical system of subject domain and ontologies) that provides the ability to represent knowledge of various types in the knowledge base [15]. The sc-model of problem solver is based on the principle that a hierarchical system of agents react to situation and events in sc-memory and interact with other corresponding agents in the sc-memory [16]. The sc-model of user interface is based on some principles to resolve specific interface tasks [1].

As a basis for knowledge representation in the framework of OSTIS Technology, a unified version of coding any kind of information named SC-code is used. The SC-code is a semantic network language with set-theoretic interpretation. Several universal variants of visualization of SC-code [14], such as SCg-code (graphic variant), SCn-code (nonlinear hypertext variant), SCs-code (linear string variant) will be shown below.

We propose the approach for generating natural language text based on OSTIS Technology. The following main advantages of this technology are considered:

- Use of unified semantical models and tools for structuring and managing the knowledge base. When faced on the new language the unified tools to design new linguistic ontologies allow to decrease laboriousness and time;
- In the user interface design using OSTIS Technology, the syntax and semantics of external natural language are described using SC-code with the appropriate ontology. Hence, the translation mechanisms between the intelligent system and the external natural language are not depend on external language. In the processing of interaction between human beings and system, for new specialized language the specification of the syntax and semantics of the language is only need;
- For the development of different components of the problem solvers in the user interface of ostis-system, the focus is on the features of different components, making any changes to the ostis-system is unified;
- The technology provides designed component oriented on natural language generation modifiability, i.e. the ability to extend its functionality or to improve its performance.

E. Tasks to be Solved for Proposed Approach Implementation

The clarification of the generation of natural language text from knowledge base is, in fact, the clarification of constructing natural language generation component using the principle of ontology-based user interface design.

Taking into account the features of natural language generation component and the design principle of ontology-based user interface, the following tasks should be solved for proposed approach:

- To develop the sc-model of natural language interface knowledge base;
- To develop the sc-model of natural language interface problem solvers used for natural language generation. According to design principle of problem solvers within OSTIS Technology, problem solvers are presented as a hierarchical system of agents.

II. The General Structure for Designing the Natural Language Interface

One of the principles of ontological method to design user interface is to treat the user interface as a specialized ostis-system oriented on the interface tasks solution. From the perspective of OSTIS Technology, as a part of user interface, natural language interface is also designed based on general principle. The design of natural language interface involves the development of the ontological model of knowledge base of natural language interface and the ontological model of problem solvers oriented on natural language interface. As a part of natural language interface, NLG component generating natural language text from knowledge base of specific ostis-system is designed underlying the principles of designing natural language interface.

The knowledge base of natural language interface provides the necessary linguistic knowledge for generating natural language text. The knowledge base provides different levels of linguistic knowledge from basic word to syntactic and semantic structure of natural language text. Some specific words that can serve as predicates have predicate-argument structures according to specific language. In the OSTIS Technology, these predicate-argument structures can be encoded into representations of knowledge base through SC-code. When the sc-structure that needs to be converted into natural language text is determined, the syntactic structure of the resulted text can be determined through the argument structure of the predicates in the sc-structure. Further, the sc-link of each sc-element in the sc-structure can be appropriately filled to generate the text, which satisfy syntax of certain language. For certain specific sc-structure, the representation form doesn't have much flexibility, the template-based method is the best choice. Templates for generating natural language text corresponding to the syntactic and semantic structure can be constructed as a logical statement in the logical ontology.

The development of problem solver of natural language interface is based on the classical pipeline architecture for realizing NLG. Based on OSTIS Technology, a group of sc-agents need to be developed to achieve the function of each part in the pipeline architecture with the help of linguistic knowledge. According to the principles of ontology-based problem solver design, the realization of the function of each part can adopt various suitable approaches. In addition, due to the complexity of sc-structure, there are a series of prepossessing of sc-structure to convert sc-structure into message triples, which are easier to express as natural language text. The multiple message triples can be transferred into a sentence. The message triples are not randomly combined to generate a sentence. Even different orderings of each element in a message triple will correspond different syntactic structures. In this situation, aggregation between these message triples need to be considered. When multiple triples are aggregated into a compound sentence, the use of referring expression can make the sentence more natural and fluent. For generating resulted text, the inflectional forms of the word stored in sc-link have to be provided to achieve the function of surface realization.

III. The Structure of the Knowledge Base of the Natural Language Interface

The knowledge base of the natural language interface has two parts (Fig. 2). Moreover, the knowledge base involves own problem solvers oriented on natural language generation from the specific subject domain based on linguistic ontologies. The language part is the ontological model of knowledge base of specific linguistics, e.g. sc-model of knowledge base for Chinese language. The subject part is sc-model of knowledge base for specific domain (e.g. History, Movies).

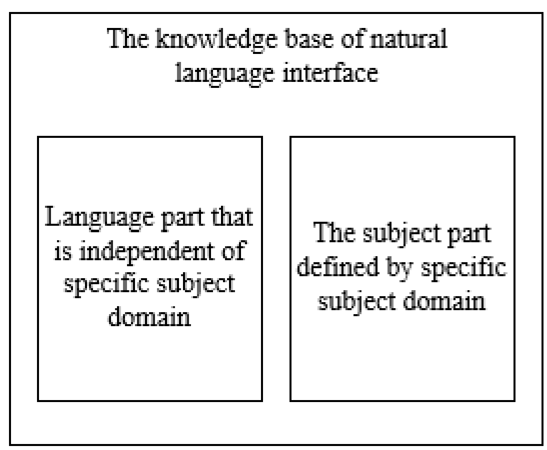


Figure 2: The knowledge base of the natural language interface

Within OSTIS Technology, the knowledge base structure of any ostis-system is described by a hierarchy of subject domains (SD) and the corresponding ontologies [15]. Subject domain is one of the key concepts in determining the structure of knowledge base. Each subject domain is the description of connections about the relevant class of research objects. The structure of knowledge base is an interconnection of various subject domains that allows to consider objects of research on different levels of detail. Each ontology is a specification of subject domain, i.e. specification of a system of concepts used in this subject domain. Each ontology provides a conceptualization of a knowledge domain by defining the classes and subclasses of the domain's individuals (entities), the types of possible relations between them etc. According to the properties of concept researched in the subject domain, every subject domain is represented by the following distinguished ontologies: structural specification, set-theoretical ontology, logical ontology and so on [14].

Any ostis-system created on OSTIS Technology is considered as a child system of **IMS Metasystem** [17]. IMS Metasystem is an ostis-system about ostis-system design automation. The knowledge base of IMS Metasystem includes a system of top-level formal ontologies used for formal description of a sense knowledge representation SC-code, such as subject domain of entities, subject domain of connections and relations and so on. They ensure quality of internal language for knowledge representation of intelligent system. Hence, from perspective of semantic network that is formal basis for SC-code, elements in SC-code is named sc-elements, such as nodes – sc-nodes, connections – sc-connectors (sc-arcs, sc-edges).

The *Subject Domain of linguistics* represents the language part. To provide formal representation for the various kinds of linguistic knowledge, a number of subject domains and their ontologies need be developed. The general structure of *Subject Domain of Linguistics* is presented in SCn-language.

Subject domain of Linguistics
⇒ *private subject domain**:
- ***SD of Chinese language texts***
- ***SD of English language texts***
- ***SD of Russian language texts***

In order to design universal translation mechanisms from the internal to the external natural language and back, specification of the syntax and semantics of the language is the core need. The syntax and semantics of specific language texts become the main object of research in *Subject Domain of linguistic*. Look at a general structure of *SD of Chinese language texts* represented in SCn.

SD of Chinese language texts
⇒ *particular SD**:
- ***SD of Chinese language syntax***
- ***SD of Chinese language semantics***

The *SD of Chinese language syntax* and *SD of Chinese language semantics* describe specification of a system of concepts from the syntactic aspect and the semantic aspect of the Chinese language, respectively [18].

In the Fig. 3, a logical statement in the logical ontology represented in SCg indicates a simple heuristic rule used for generating natural language text. The heuristic rule can be used to generate a simple declarative sentence based on template. When the role of each element of a triple in the sentence is determined, e.g., identifier of a element is "binomial theorem" served as the subject of sentence, identifier of another element is "observation" served as the object. The relation between them is membership. Based on the template the resulted sentence "The binomial theorem is a kind of observation" is generated.

Subject Domain of specific domain represents the subject part, which is dependent from the ostis-system. For example, in the knowledge base of OSTIS intelligent tutoring system for Discrete Math, the subject part contains various information about the discrete math domain, such as the type of theorem, inclusion relation of graph theory and so on.

In the knowledge base of ostis-system, name of each sc-element is an arbitrary unique string stored in the sc-link. There are three main commonly used options for naming sc-elements: system identifier, main identifier, and identifier. Among them, it is recommended to use the system identifier as the name for sc-elements. The system identifier is unique for a sc-element within the entire knowledge base of a given ostis-system. We follow the convention that how to construct a system identifier of sc-elements with specific label. However, there are obvious distinguish between the system identifier and the name of the sc-elements used in natural language texts. For words corresponding to system identifiers that the systems wish to use in the resulted texts, the lexical unit (lexicon entry) has to be provided, which specifies properties of that word (e.g., inflectional forms of word in English), as well as other information.

Lexical unit is a fragment of linguistic ontologies, it represents the words that need to be used in this subject domain. Strictly speaking, it belongs to **SD of natural language word**. The lexical unit consists of closed-class words and opened-class words. The lexical units of closed-class words, like determiners and prepositions, are domain-independent. The lexical units of opened-class words usually are basic for constructing system identifier of sc-elements in specific subject domain.

Fig. 4 shows the linguistic knowledge for lexical unit (in Chinese). The lexical units for adjectives and others are similar.

The sc-structure specifies that a lexical unit whose system identifier is "I_ch_to_be". The role relations "arg0" and "arg1" indicate predicate-argument relation in the linguistic theory FrameNet. Notice that in addition to describing the inflectional forms of the lexical unit (when it's for English or Russian language), there are also other

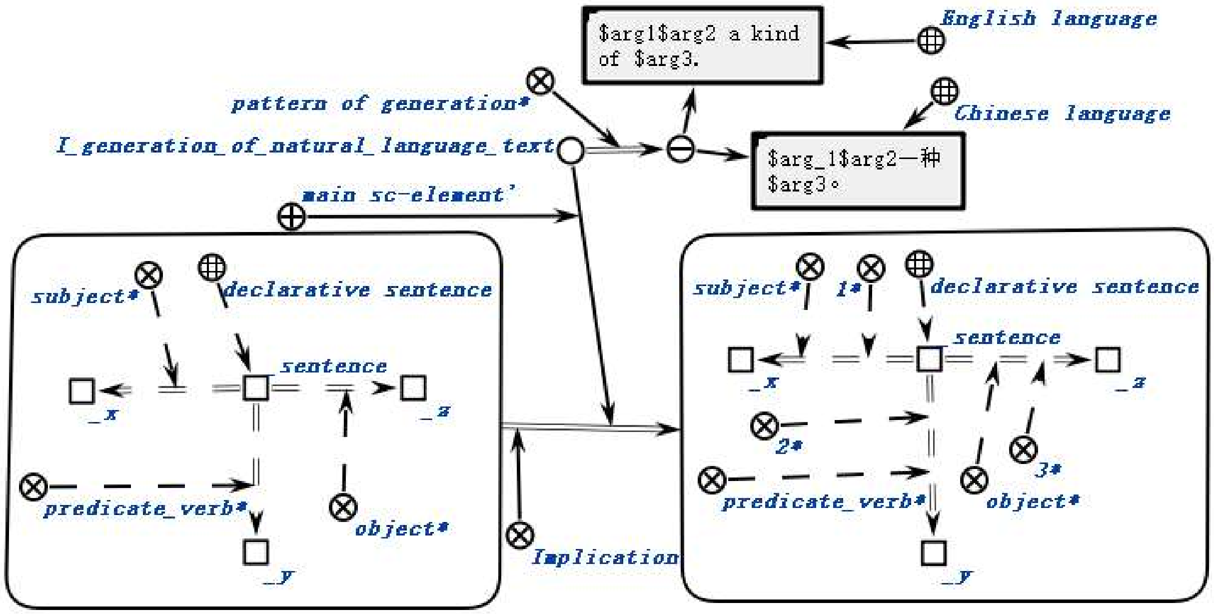


Figure 3: Logical statement about generating natural language text

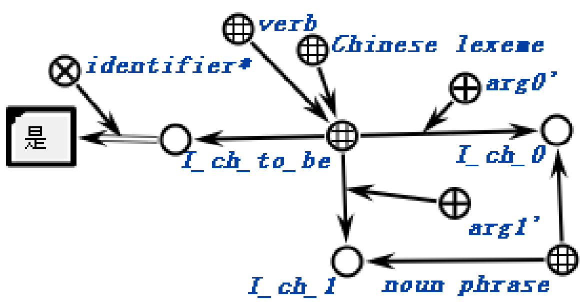


Figure 4: linguistic information for lexical unit "to be"

information about syntactic level and semantic level of the lexical unit to be specified, for example, the part-of-speech of the lexical unit. These information allows to generate most appropriate words in the resulted generated texts.

The information about semantic level of lexical units is essential for constructing syntactic structure. We provided the information about semantic level in *the subject domain of natural language semantic*. Among them, the most important core part is the semantic role frame of predicate. We note that there are many useful linguistic resources for constructing ontologies of natural language. For construction of ontologies of lexical units, some general-purpose lexical unit would be to exploit, such as Chinese WordNet [19], Chinese part of ConceptNet [20] and so on. In *the Subject domain of Chinese language semantic*, with the help of the CPB, Mandarin VerbNet [21], and other linguistic resource, we will consider the concepts, such as semantic frame of predicate, semantic role and so on. The Subject domain of Chinese language semantic provides the semantic information for certain specific lexical units. The general-purpose lexical units, however, often do not cover the highly technical concepts of domain knowledge base. Sometimes it is necessary to tailor or design lexical units from general-purpose lexical units that are suitable for development of a specific domain knowledge base.

IV. The Problem Solver Structure of the Natural Language Interface

Within the OSTIS Technology, the development of problem solvers actually comes down to development of its knowledge base. The constructed knowledge base includes the own problem solvers having program agents. The multi-agent approach is used as a basis for problem solvers design. The interaction of agents will be performed exclusively in the semantic memory (sc-memory), which stores the SC-code constructions. Such approach provides the flexibility and modularity of developed problem solvers, as well as provides the ability to integrate various methods to problem solutions. In the term of implementation, agent programs can implement logical reasoning based on a hierarchy of statements comprised in the logical ontology, as well as the data-driven learning algorithms using various programming language.

Abstract sc-agent generating external texts from the knowledge base is a group of sc-agents that implement the mechanisms of natural language generation from knowledge base, i.e. given a sc-structure containing a set of sc-sentences, sc-agents generate a corresponded fluent natural language text. One of the possible approaches for implementation of this abstract sc-agent is to build a collection of simpler abstract sc-agents by the relation abstract sc-agent decomposition*. The following is general structure for problem solver of natural language interface represented in SCn-language.

Abstract sc-agent of natural language interface
⇐ *decomposition of an abstract sc-agent**:
  {
  - *Abstract sc-agent translating external texts into the knowledge base*
  - *Abstract sc-agent generating external texts from the knowledge base*
    ⇐ *decomposition of an abstract sc-agent*:*
      {
      - *Abstract sc-agent for content selection*
      - *Abstract sc-agent text planning*
      - *Abstract sc-agent for micro-planning*
      - *Abstract sc-agent for surface realization*
      }
  }

In the technology framework of OSTIS, a sc-agent is some entity that can perform actions in the sc-memory. The notions of sc-agent and related concepts are specified in the subject domain of abstract sc-agent and the corresponding ontologies. The abstract sc-agents is a certain class of functionally equivalent sc-agents, various items of which can be implemented in different ways to specific tasks.

Implementing content selection from knowledge base includes a sc-agents of the following types:

Abstract sc-agent for content selection
⇐ *decomposition of an abstract sc-agent**:
  {
  - *Abstract sc-agent determining sc-structure*
  - *Abstract sc-agent dividing determined sc-structure into basic sc-structure*
  - *Abstract sc-agent transferring basic sc-structure into message triple*
  - *Abstract sc-agent determining the candidate sc-structures*
  - *Abstract sc-agent transferring candidate sc-structures into message triples*
  }

Abstract sc-agent for determining sc-structure - the groups of agents that provide the retrieval from the domain knowledge base sc-structures containing sc-sentences from which we will transform to natural language texts. This stage is to determine what we want to talk.

Abstract sc-agent dividing determined sc-structures into basic sc-structures – the agents that implement the mechanisms of decomposition of retrieved sc-structures into basic structures, which can be transferred into message triples. Sometime several message triples transformed from sc-structure are redundancy, so the system finally selects among the basic sc-structures the ones to be transferred.

Abstract sc-agent determining the candidate sc-structures – the agents that implement the mechanism of determination of appropriate candidate sc-structures that satisfy specific end-users. ***Abstract sc-agent transferring candidate sc-structures into message triples*** – the agents that implement the mechanism of conversion of candidate sc-structures to message triples.

Abstract sc-agent for text planning - the agents that implement the function that message triples are ordered, in effect the ordering of sentences in the resulting text. The following structure of this abstract sc-agent is considered:

Abstract sc-agent for text planning
⇐ *decomposition of an abstract sc-agent**:
  {
  - *Abstract sc-agent ordering message triples*
  - *Abstract sc-agent ordering entities of a message triple*
  }

Abstract sc-agent for miro-planning - the agents that implement the mechanism of transferring message triples to abstract sentence specifications that are varied accorss NLG system, for example, the simple text templates with slots, syntactic structures and so on. The structure of abstract sc-agent for micro-planning:

Abstract sc-agent for micro-planning
⇐ *decomposition of an abstract sc-agent**:
  {
  - *Abstract sc-agent constructing sentence plan*
  - *Abstract sc-agent for sentence aggregation*
  - *Abstract sc-agent generating the referring expression*
  }

Abstract sc-agent for surface realization - the group sc-agents that implement the mechanisms of concatenating the sc-links to generate natural language text, i.e. retrieving resulting forms of lexical units according to rules, as well as filling and concatenating the sc-links.

The above listed problem solvers are not fixed. It is possible to adjust and make extensions for the already developed abstract sc-agents, i.e. the structure of problem solver is flexible and changeable. This kinds of features for designing problem solver are due to the advantages of multi-agent approaches by OSTIS Technology: adding a new agent or removing (deactivation) of one or more existing agents usually does not lead to changes in other agents; agents often work in parallel and independently from each other and so on.

V. Implementation of Chinese Language Generation

With the help of previously constructed linguistic ontologies and problem solvers in the ostis-system, the following example shows the processing stage for implementing Chinese language generation.

Several constrains are defined:

- The input of system is completed and has sense;
- The input given sc-structure formally represented in the knowledge base of specific SD (Discrete Math);
- The output of system is a simple narrative Chinese sentence;
- The knowledge base includes entity signs with Chinese identifier, which will be used in the resulted sentence

For generating natural language text from ostis-system specific subject, we propose that the realization of natural language generation is roughly divided into two steps: rule-based symbolic generator converting structure of knowledge base to message triples; rule-based approached or statistical generator (when high quality aligned datasets is accessible.) translating message triples to natural language text. Unfortunately, the high quality aligned datasets is relatively difficult to access. The example in this article is just to generate a simple narrative Chinese sentence. The rule-based approach is used to illustrate the process.

Step 1 We are given a sc-structure formally represented in SCg (Fig. 5).

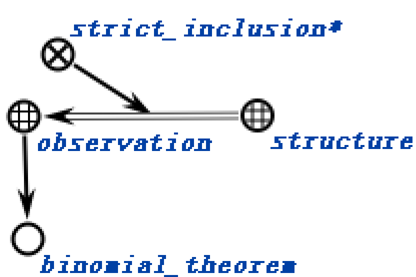


Figure 5: Sc-structure of knowledge base of Discrete Math

Step 2 The given sc-structure is transferred to a set of message triples, the identifier of each sc-element of message triples corresponds to a certain lexical unit of specific language. Description of lexical units that is stored in the linguistic knowledge base had been mentioned above. In the example, we just consider one of the set of message triples. Moreover, the predicate "to be" will be used for the independent property "instanceOf" (Fig. 6).

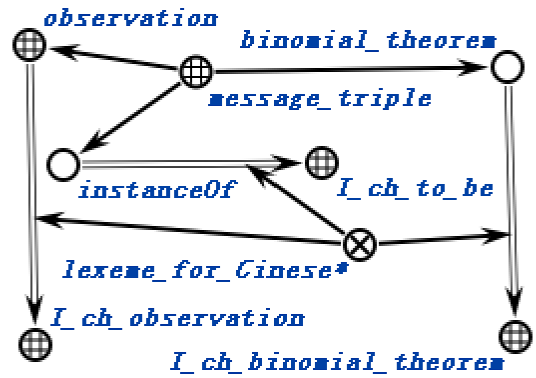


Figure 6: Message triple for part of sc-structure

Step 3 For each property of message triples, a suitable predicate-argument structure from the linguistic ontologies is matched. For this example, the specific template shown in the Fig. 3 will be used.

Step 4 The agent fills the sc-links of corresponded sc-elements of message triples, i.e. a certain sc-link needs to be filled with the result of a certain inflection form of a lexical unit. For Chinese language, there is not a certain inflection form for lexeme, e.g., the result of a lexical unit "binomial theorem" in Chinese (Fig. 7). Therefore, the processing of this step is relative simpler.

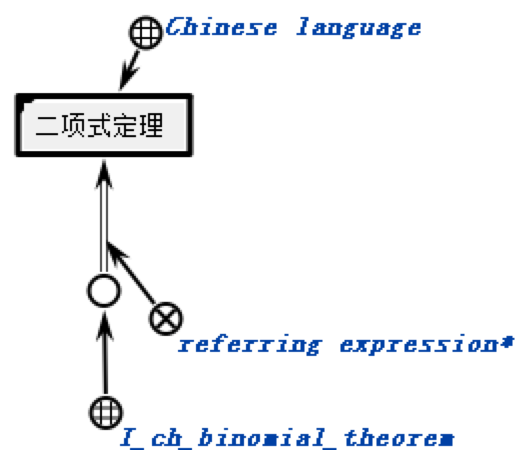


Figure 7: The sc-link filled by the Chinese lexical unit "binomial theorem"

Step 5 Previous sc-links are concatenated to generate the resulted Chinese sentence according to valid ordering (Fig. 8).

In the knowledge base there are different identifiers for each lexical unit. In this example just the simple declaration sentence is considered, the referring expression of the lexical unit "binomial theorem" is itself in Chinese language. Based on the template, expression referring

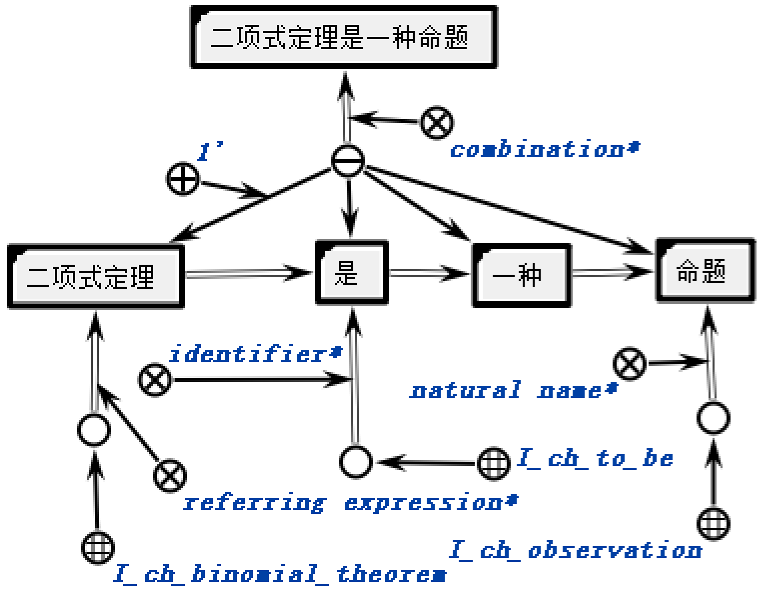


Figure 8: Generation of Chinese sentence corresponding to sc-structure

to the lexical unit "binomial theorem" is served as the subject. The lexical unit "observation" is served as object in Chinese language. The template is predefined with fixed phrase, in the (Fig. 8) the sc-link without corresponding lexical unit is the fixed part.

When these steps are implemented, a fragment sc-structure of knowledge base of Discrete Math domain can be transferred into Chinese sentence with specific sense. This natural language text is easier to access to ordinary end-users.

Nowadays there are two types of methods for evaluating generated natural language text: automatic metrics and manual evaluation. The BLEU [22] and ROUGE [23] and METEOR [24] score are widely used as the automatic metrics. We begin by comparing the natural language text, which is generated by the system, with the natural language text described by human using the common automatic metrics BLEU. Notice that we couldn't compare the performance of system with other state-of-the-art system due to the absence of the system for processing complex semantic structure in knowledge base.

Generally speaking, when describing the performance of a text generation system, the cumulative score from BLEU-1 to BLEU-4 is usually reported. Table. I shows that performance of different approaches for generating text. The templates are usually predefined by human, therefore, the text generated according to templates can get high scores. The BLEU-4 score "0.00" indicates the predefined templates is not flexible. However, for multiple sc-sentences, these approaches couldn't obtain satisfactory scores. The system still has certain deficiencies when generating longer natural language text.

Table I: Evaluation of Approaches for Text Generation

	Template-based for one basic sc-sentence	Rule-based for one basic sc-sentence	Template-based/rule-based multiple basic sc-sentences
BLEU-1	1.00	0.78	0.64
BLEU-2	1.00	0.62	0.53
BLEU-3	1.00	0.53	0.41
BLEU-4	0.00	0.32	0.23

VI. Conclusion

This article has proposed an ontological approach to design a unified semantic model for natural language generation from knowledge base. The use of this approach offers the following advantages:

- The model for natural language generation is applied to generate fluent, coherent, and multi-sentence natural language texts appropriating for end-users;
- The semantic structure that processed by this model is more complex than simple tabular or triples structure;
- as a part of the model, the Chinese linguistic knowledge base is designed to generate grammatically similarity texts. The knowledge base make the model more explanatory than statistical model.

We discussed the structure of knowledge base of natural language generation component of natural language interface, the processing stage for Chinese language generation, the optional linguistic knowledge base used in each stage. Relying on the component to automatically produce texts from knowledge base makes the information of intelligent system easily accessible not only to computer programs, but also to end-users.

We are currently working towards implementing the model for Chinese language generation in certain trial, in order to verify the practicality of the model and to evaluate the quality of generated text. It would also be particularly interesting to explore the model's possibility to support multiple language generation as long as the corresponding linguistic ontology is added.

REFERENCES

[1] Sadouski M.E., Boriskin A.S., Koronchik D.N., Zhukau I.I., Khusainov A.F.: Ontology-Based Design of Intelligent Systems User Interface. In: 7th International Scientific and Technical Conference "Open Semantic Technologies for Intelligent Systems", pp. 95-106. Belarus Minsk (2017)

[2] Introducing the Knowledge Graph: Things, Not Strings. Available at: https://blog.google/products/search/introducing-knowledge-graph-things-not/

[3] Plachouras V., Smiley C., Bretz H., Taylor O., Leidner J. L., Song D., Schilder F.: Interacting with financial data using natural language. In Proc. SIGIR'16, pp. 1121-1124. Italy Pisa (2016)

[4] Reiter E.: Pipelines and Size Constraints. Computational Linguistics, 2000, vol.26 No 02, 251-259.

[5] Meteer M. W.: Bridging the generation gap between text planning and linguistic realization. Computational Intelligence, 1991, vol.07 No 04, 296-304.

[6] Simon M., Roberto C., Alicia B., Leo W.: FORGe at SemEval-2017 Task 9: Deep sentence generation based on a sequence of graph transducers. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 920-923. Canada Vancouver (2017)

[7] Sutskever I., Vinyals O., Le Q. V.: Sequence to sequence learning with neural networks. In Proc. NIPS'14, pp. 3104-3112. Canada Montreal (2014)

[8] Natural Language Generation for the Semantic Web. Available at: https://webnlg.loria.fr/pages/index.html/

[9] DBpedia. Available at: https://www.dbpedia.org/

[10] Amit M., Yoav G., Ido D.: Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2267-2277. USA Minneapolis (2019)

[11] Xu B. Xiao Y.H. CN-DBpedia: A Never-Ending Chinese Knowledge Extraction System. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp. 428-438. Springer, Cham, 2017.

[12] Niu X., Sun X., Wang H., Rong S., Qi G., Yu Y.: Zhishi.me - Weaving Chinese Linking Open Data. In: The Semantic Web – ISWC 2011. ISWC 2011. Lecture Notes in Computer Science, vol 7032. Springer, Berlin, Heidelberg, 2011.

[13] Androutsopoulos I., Lampouras G., Galanis D.: Generating Natural Language Descriptions from OWL Ontologies: the NaturalOWL System. Journal of Artificial Intelligence Research, 2013, vol.48 No 01, pp. 671-715.

[14] Golenkov V.V. Gulyakina N.A. Proekt otkrytoi semanticheskoi tekhnologii komponentnogo proektirovaniya intellektual'nykh sistem. Chast' 1 Printsipy sozdaniya [Project of open semantic technology of component designing of intelligent systems. Part 1 Principles of creation]. Ontologiya proektirovaniya [Ontology of designing], 2014, no 1, pp. 42-64. (in Russian)

[15] Davydenko I.T.: Ontology-based knowledge base design. In: 7th International Scientific and Technical Conference "Open Semantic Technologies for Intelligent Systems", pp. 57–72. Belarus Minsk (2017)

[16] Shunkevich D.V.: Ontology-based design of knowledge processing machines. In: 7th International Scientific and Technical Conference "Open Semantic Technologies for Intelligent Systems", pp. 73–94. Belarus Minsk (2017)

[17] (2021, May.) The IMS.OSTIS website. [Online]. Available: http://www.ims.ostis.net/

[18] Qian L., Sadouski M., Li W. Ontological Approach for Chinese Language Interface Design. In: Golenkov V., Krasnoproshin V., Golovko V., Azarov E. (eds) Open Semantic Technologies for Intelligent System. OSTIS 2020. Communications in Computer and Information Science, vol 1282. Springer, Cham, 2020.

[19] (2021, May.) The Chinese WordNet. [Online]. Available: https://lope.linguistics.ntu.edu.tw//cwn/

[20] (2021, May.) Chinese part of ConceptNet. [Online]. Available: http://conceptnet.io/c/en/chinese

[21] (2021, May.) Mandarin VerbNet. [Online]. Available: http://verbnet.lt.cityu.edu.hk/#/

[22] K. Papineni, S. Roukos, T. Ward, W.J. Zhu. BLEU: a method for automatic evaluation of machine translation, Proceedings of the 40th annual meeting on association for computational linguistics (2002), pp. 311-318.

[23] C.Y. Lin. Rouge: A package for automatic evaluation of summaries, Text Summarization Branches Out (2004).

[24] S. Banerjee, A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization (2005), pp. 65-72.

Онтологический подход к генерации естественного языка из базы знаний

Цянь Лунвэй, Ли Вэньцзу

В статье рассматриваются существующие методы разработки естественно-языкового интерфейса, а также методы к реализации генераций естественного языка из базы знаний как компантент естественно-языкового интерфейса. Был проведен анализ проблем, возникающих при генерации естественного языка из структурированных данных (в частности база знаний) в настоящее время.

На основании различных рассмотренных методов был предложен онтологический подход к генерации естественного языка, который позволяет интегрировать разные типы методов генерировать тексты естественного языка. Предложенный метод направлен на разработку семантической модели знаний о линквистике. Этапы реализации подхода были созданы лингвистические онтологии и решатели для генерации естественного языка. Лнигвистические онтологии выражает синтаксические и семантические знания конкретного языка, которые можно использовать решателями для генерации естественного языка.

Таким образом, для дальнейшей обработки и реализации части естественно-языкового пользовательского интерфейса в работе предлагается модель генерации естественного языка из базы знаний конкретного домена, основанная на знаниях. Более того, в качестве китайского языка, функции каждых этапов генерации и назначение лингвистических онтологий в процессе генерации проиллюстрированы, чтобы проверять практичность модели.

Received 14.05.2021

Development of a problem solver for automatic answer verification in the intelligent tutoring systems

Wenzu Li, Longwei Qian
Belarussian State University Informatics and Radioelectronics
Minsk, Belarus
lwzzggml@gmail.com, qianlw1226@gmail.com

***Abstract*—This article proposes an approach to developing a problem solver for automatic answer verification in intelligent tutoring systems constructed using OSTIS Technology. The problem solver is developed based on multi-agent technology. The developed problem solver automatically verifies the correctness and completeness of the user answer at the semantic level by calling the corresponding sc-agent sets according to the type of the questions (multiple-choice questions, fill in the blank questions, questions of definition interpretation, etc.). Sometimes there may be multiple standard answers for the question of definition interpretation, but the problem solver can automatically filter a semantic fragment of the standard answer in advance that best matches the user answer according to the semantic fragment of the user answer, and then continue to verify the correctness and completeness between them.**

***Keywords*—problem solver, answer verification, OSTIS Technology, intelligent tutoring systems, ontology, knowledge base**

I. Introduction

With the development of modern information technology, artificial intelligence as an important part of modern computer applications is rapidly integrating into the field of education. Applying artificial intelligence technology to the field of education can not only provide new teaching methods and tools, but also enable learners to spend less time acquiring more useful knowledge, effectively improving the accuracy of information retrieval and the efficiency of knowledge selection. At the same time, the combination of artificial intelligence technology and the educational process is also one of the important means to ensure the fairness of education, so that every learner has an equal opportunity to obtain knowledge. Especially since the outbreak of COVID-19, intelligent tutoring systems (ITS) have played an increasingly important role in distance education [3].

As more and more ITS used in different fields are developed, most developers believe that ITS need to meet at least the following basic functions:

- automatic verification of answers, if an error is found, the cause of the error needs to be analyzed and corrective measures taken;
- according to the learner's level and learning situation, automatically adjust the learning content and progress;
- automatic generation of various questions and exercises;
- have the ability to generate and understand natural language, and realize relatively free Human-Machine conversation;
- have the ability to explain teaching content;
- automatic problem solving based on understanding the teaching content.

As the most basic and critical function of ITS, automatic answer verification can quickly check the user's grasp of new knowledge, and can greatly improve the user's learning efficiency, allowing users to obtain the most knowledge in a limited time. Usually answer verification needs to solve the following basic tasks:

1) subjective question answer verification and objective question answer verification;
2) analysis of correctness and completeness of answers;
3) verification a type of question with multiple standard answers, and these standard answers do not satisfy logical equivalence (for example, the definition of a square);
4) whether the logic formula of standard answer and user answer meets the equivalence;
5) analysis of decision sequence and logic rationality when users solve problems;

According to the types of questions, automatic answer verification can be divided into: 1. objective question answer verification; 2. subjective question answer verification. Objective questions refer to a type of question that has a unique standard answer. In this article, objective questions include: multiple-choice questions, fill in the blank questions and judgment questions. Objective questions differ from subjective questions, which have more than one potential correct answer and sometimes have room for a justified opinion. Subjective questions in this article include: definition explanation questions, proof questions and theorem interpretation questions. Because the answers to objective questions are fixed and simple, ITS on the market basically have the function of objective question answer verification. However, since subjective question answer verification needs to involve natural language processing (NLP), linguistics and other aspects of knowledge, currently only some ITS have the function of subjective question answer verification [4], [7].

With the development of semantic web, deep learning and NLP technology, subjective question answer verification has become a very important research direction. Therefore, we have introduced in detail the existing subjective question answer verification approaches and their advantages and disadvantages in the literature [4], and on the basis of these approaches, we propose an semantic-based answer verification approaches (subjective question answer verification and objective question answer verification). The basic principle of this approach is to first decompose the standard answers and user answers in the form of semantic graphs into substructures according to the knowledge representation rules in the ostis-systems (system built using OSTIS Technology (Open Semantic Technology for

Intelligent Systems)), and then calculate the similarity between the semantic graphs by comparing the matching relationships of the decomposed substructures, finally, the correctness of the user answer is judged according to the similarity [1], [2], [6]. A semantic graph is a network that represents semantic relationships between concepts. In the ostis-systems, the semantic graph is constructed using SC-code (as a basis for knowledge representation within the OSTIS Technology, a unified coding language for information of any kind based on semantic networks is used, named SC-code). The user answer in natural languages is converted to SC-code using the natural language interface [5].

In literature [4], we only briefly introduced the process of using semantics to verify user answers at the theoretical level, and conducted a feasibility analysis of the proposed approach, but the article does not involve the specific process of using the program to implement each step. Therefore, in this article, a problem solver for answer verification in the ostis-systems is developed based on the answer verification approach proposed in [4]. One of the key components of each intelligent system is the problem solver, which provides the ability to solve various problems. The developed problem solver mainly solves the tasks (1), (2), (3) listed above, and for the solutions to the remaining more complex tasks, we will introduce them in detail in the subsequent articles. The discrete mathematics tutoring system developed using OSTIS Technology will serve as a demonstration system for the problem solver.

II. Existing approaches and problems

According to the type of knowledge in the knowledge base of the ITS, answer verification can be divided into:

- factual knowledge answer verification;
- logical knowledge answer verification.

Factual knowledge refers to knowledge that does not contain variable types, and this type of knowledge expresses facts. In the knowledge base of ostis-systems, objective questions and their answers are usually described using factual knowledge. Among them, the user answers to objective questions in the form of natural language have been aligned (entity alignment) with the existing knowledge in the knowledge base when they are transformed into SC-code through the natural language interface. Therefore, there is no need to consider the similarity between concepts or relations at the language level when calculating the similarity between the answers to objective questions. That is, SC-nodes that represent the same concept or relation in the knowledge base have a unique main identifier.

Logical knowledge usually contains variables, and there are logical relations between knowledge. In the ostis-systems SCL-code (a special sub-language of the SC language intended for formalizing logical formulas) is used to represent logical knowledge. The answers to subjective questions in the knowledge base are described in the form of logical formula using logical knowledge. Because, the semantic segment used to represent the answer to subjective question in the knowledge base contains variables (equivalent to the bound variable in the predicate logic formula), and there is a strict logical sequence between each sc-node in the semantic segment. Therefore, when using the problem solver to calculate the similarity between the semantic graph of standard answer and the semantic graph of user answer, it is necessary to establish the mapping relationship between the potential equivalent variables between the two semantic graphs according to the semantic structure and the position of the variables in the logical formula. Among them, establishing the mapping relationship of potential equivalent variables between semantic graphs is the most basic and critical step in subjective question answer verification. In this article, we can regard semantic graphs describing standard answers and user answers as partial fragment of the ontology (ontology is a type of knowledge, each of which is a specification of the corresponding subject domain, focused on describing the properties and relations of concepts that are part of the specified subject domain) [2], [4], [6]. Therefore, the approach to establishing a mapping relationship between semantic graphs is similar to the approach to establishing a mapping relationship between ontology. At present, there are many mature tools and approaches to establishing the mapping relationship between ontology, and we will consider them in detail next.

Ontology mapping

With the rapid development of the new generation of semantic web, the ontology as the foundation of the semantic web has become a research hot-spot, and many ontology libraries with rich semantic information and practical value have emerged. These ontology libraries have huge differences due to different developers, application purposes and application fields, and they cannot communicate and inter-operate effectively with each other. The ontology mapping is a key step to solve the heterogeneity of ontology and realize knowledge sharing and ontology inter-operation. The core idea of ontology mapping is to calculate the similarity between elements (concepts, attributes, and instances) in different ontology, and then establish the mapping relationship between the elements according to the similarity and mapping strategy.

Due to the rapid development of ontology mapping related technologies, many concepts with similar semantics and different names have emerged, such as **Ontology Mapping**, **Ontology Alignment**, **Ontology Matching**, **Ontology Integration**, **Ontology Fusion**, and **Ontology Merging**. It is generally believed that ontology mapping, ontology alignment and ontology matching are concepts with the same semantics, that is, the mapping relationship between elements in different ontology is established according to the mapping strategy. Ontology integration, ontology fusion, and ontology merging generally produce new ontology, and ontology mapping is their basic task [10], [11]. Ontology mapping involves multiple steps such as ontology preprocessing. Since this article focuses on the establishment of element mapping relationships between ontology, other steps will not be introduced in detail.

There are already many mature ontology mapping algorithms and mapping systems:

- a comprehensive similarity calculation algorithm that establishes semantic mapping relationships between elements (concepts, attributes, and instances) between RDFS ontology is introduced in literature [9]. Taking concepts between ontology as an example, the algorithm first uses Edit Distance (Levenshtein Distance) to calculate the similarity of names between concepts, and then calculates the similarity of the instances of the concept according to the ratio of the number of instances matched between the concepts and the number of all instances, and finally, the structural similarity of the concepts is calculated according to the relationship between the number of the same father and child concepts and the number of all adjacent concepts between the concepts. Combine the above three similarities and set different weights to get the final similarity of the concept. Finally, according to the comprehensive similarity between concepts and the mapping strategy, the mapping relationship of equivalent concepts between different ontology is established;
- with the rapid development of machine learning in recent years, many ontology mapping approaches based on machine learning frameworks have emerged. An approach

to alignment of entities between knowledge graphs based on machine learning is introduced in the literature [12]. The knowledge graph is regarded as a formal description of things and their interrelationships in the objective world. The approach consists of two parts: 1. knowledge representation learning; 2. learning of mapping relationships between entities; Knowledge representation learning refers to the use of machine learning algorithms to learn the semantic representation of entities and relationships in the knowledge graph. The learning of the mapping relationship between entities refers to learning the mapping relationship of entity pairs between knowledge graphs according to the manually labeled data sets [13];
- with the development of ontology mapping technology, many mature ontology mapping systems have emerged, among which the most representative ones are RiMOM and ASMOW. RiMOM is an ontology mapping system developed based on Bayesian decision theory. RiMOM uses similarity propagation theory and multiple heuristic rules to establish the mapping relationship between concepts. ASMOV is an automated ontology mapping tool developed by Jean-Mary et al. Its goal is to promote the integration of heterogeneous ontology. ASMOV uses an iterative calculation method to analyze multiple characteristics of elements to calculate the similarity of element pairs between ontology, and to establish mapping relationships between concepts, attributes, and instances in turn [15], [16].

Although the ontology mapping approaches introduced above have many advantages, they also have many problems:

- the traditional algorithm for calculating the similarity of element pairs between ontology requires iterative calculation of the similarity between the current element in the source ontology and each element in the target ontology. Therefore, when the amount of knowledge contained in the ontology is very large, it may take several minutes or more to establish the mapping relationship between the ontology, and real-time mapping cannot be performed;
- using machine learning algorithms for ontology mapping has improved the accuracy of ontology mapping to a certain extent, but it requires a huge amount of human resources to label matching element pairs between ontology;
- ontology mapping is a very complicated process, and no approach is perfect. Especially in recent years, the generation of big data has led to the generation of big ontology, but the existing ontology mapping system and approach cannot perform ontology mapping in a big data environment.

Establishing the mapping relationship of potential equivalent variable pairs between the semantic fragments of the answers to subjective question is a key step of logical knowledge answer verification. The part of the process of establishing the mapping relationship between potential equivalent variables pairs is similar to the establishment of the mapping relationship between the equivalent element pairs of the ontology. However, in the ostis-systems, the knowledge base is constructed using SC-code, and the knowledge in the knowledge base has a specific knowledge structure and knowledge representation approach, so the existing ontology mapping algorithms cannot be used directly. Therefore, based on the existing ontology mapping approach and OSTIS Technology, this article proposes an approach to establish the mapping relationship of the potential equivalent variables pairs between the semantic fragments of the answers to subjective questions based on the semantic structure.

III. Proposed approach

Based on the OSTIS Technology used to develop semantic intelligence systems and the corresponding platforms, tools and approaches, an approach to developing a problem solver for automatic answer verification is proposed in this article.

Each ostis-system for different application fields includes a platform for interpretation semantic models of ostis-systems, as well as a semantic model of ostis-systems using SC-code (sc-model of ostis-systems). At the same time, the sc-model of the ostis-systems includes the sc-model of the knowledge base, the sc-model of the problem solver and the sc-model of the interface (in particular, the user-oriented intelligent interface). The rules and methods for detailed design of the knowledge base and problem solver in the ostis-systems are described in [1].

Using the models, application tools and approaches provided by OSTIS Technology in the framework of this work will provide the following possibilities:

- the developed problem solver can be easily transplanted to the ostis-systems for different application fields;
- verifying the correctness and completeness of user answers at the semantic level;
- saving the user's test record;
- analyzing the user's test results from the semantic layer and logic layer, and give reference opinions;
- answer verification does not depend on the natural language (English, Russian, Chinese, etc.).

Next, we will introduce in detail the development process of the problem solver for automatic answer verification. In order to facilitate the explanation of the working principle of the problem solver, the illustrations and knowledge base fragments we choose in this article are all in English, but it needs to be emphasized that the problem solver developed does not depend on natural language.

IV. Development of problem solver

In the ostis-systems, the problem solvers are constructed based on a multi-agent approach. According to this approach, the problem solver is implemented as a set of agents called sc-agents. All sc-agents interact through common memory, passing data to each other as semantic network structures (sc-texts) [1], [2].

According to the requirements of the task, the developed problem solver for answer verification in this article needs to solve the following problems:

- the problem solver can not only verify the answer to objective question with the only correct option, but also verify the answer to objective question with multiple correct options (multiple-choice questions with multiple options and partially fill in the blank questions). If the user's answer is incorrect or incomplete, the correct standard answer needs to be displayed at the end;
- for subjective questions, it is necessary to verify the completeness and correctness of the user answer (for example, the answer is correct but incomplete, or the answer is partially correct, etc.). If the subjective question has multiple logically unequal standard answers (for example, the definition of triangle), the problem solver can automatically select the appropriate standard answer according to the user answer, and then verify the answer. Finally, if the user answer is incorrect (complete error or partial error), the problem solver also needs to find the incorrect part of the user's answer and display the correct standard answer.

The problem solver of any ostis-system is a hierarchical system of knowledge processing agents in semantic memory. In order to achieve the tasks listed above, some sc-agents are developed in this article. The hierarchy of the knowledge processing agents in the problem solver for automatic answer verification is shown in SCn-code (one of SC-code external visualization languages) [6] as follows:

Problem Solver For Automatic Answer Verification
$\Leftarrow$ *abstract sc-agent decomposition**:
{
- *Sc-agent for computing semantic similarity of factual knowledge*
- *Sc-agent for processing semantic similarity calculation results of factual knowledge*
- *Sc-agent for computing semantic similarity of logical knowledge*

}

The basic principle of automatic answer verification in the ostis-systems is to calculate the semantic similarity between the standard answer and the user answer. In the knowledge base, the answers to objective questions are described using factual knowledge, and the answers to subjective questions are described using logical knowledge [4]. Therefore, when using the problem solver to verify the answer, it realizes the automatic verification of the answer by calling different sc-agents according to the type of question. It should be emphasized that answer verification is only one of the main uses of the problem solver. The problem solver can also calculate the similarity between arbitrary semantic fragments constructed using sc-code by calling different sc-agents.

When verifying the answer to the objective question, the calling flow of sc-agents is as follows:

1) if the problem solver judges that the current question is an objective question, the sc-agent for computing semantic similarity of factual knowledge is called to start calculating the similarity between the semantic graph of standard answer and the semantic graph of user answer;
2) when the first step is completed, the problem solver automatically calls the sc-agent for processing semantic similarity calculation results of factual knowledge. The final verification result is given by this sc-agent according to the question type, characteristics (for example, multiple-choice questions with multiple correct options) and similarity between answers.

When verifying the answer to the subjective question, the calling flow of sc-agents is as follows:

- if the problem solver judges that the current question is a subjective question, the sc-agent for computing semantic similarity of logical knowledge is called.

It should be emphasized that the answers to subjective questions are described based on logical formula, so under certain conditions, the logical equivalence between the answers (equivalence judgment between the logic formulas) needs to be considered. Due to the limitation of the number of pages in this article, we will introduce the design approach and calling flow of sc-agent for verifying logical equivalence in the following article. Next, the specific functions and implementation process of each sc-agent will be introduced in detail.

A. Sc-agent for computing semantic similarity of factual knowledge

The basic function of the sc-agent for computing semantic similarity of factual knowledge is to calculate the similarity between semantic graphs described using factual knowledge. Because the answers to objective questions in the knowledge base are described using factual knowledge, the similarity between the answers can be calculated using this sc-agent. The similarity between answers is the basis for the objective question answer verification. When the user answer in natural language is converted to SC-code, it has been aligned with the existing knowledge in the knowledge base (such as coreference resolution, etc.), that is, elements with the same semantics have the same main identifier in the knowledge base. Therefore, in this article, the similarity between semantic graphs is calculated based on semantic and knowledge representation structures [4], [8].

The sc-agent for computing semantic similarity of factual knowledge needs to complete the following tasks:

1) according to the representation rules of factual knowledge, the standard answers and user answers in the form of semantic graphs are decomposed into substructures;
2) using formulas (1), (2), and (3) to calculate the precision P_{sc}, recall R_{sc} and similarity F_{sc} between semantic graphs.

$$P_{sc}(u, s) = \frac{|T_{sc}(u) \otimes T_{sc}(s)|}{|T_{sc}(u)|} \tag{1}$$

$$R_{sc}(u, s) = \frac{|T_{sc}(u) \otimes T_{sc}(s)|}{|T_{sc}(s)|} \tag{2}$$

$$F_{sc}(u, s) = \frac{2 \cdot P_{sc}(u, s) \cdot R_{sc}(u, s)}{P_{sc}(u, s) + R_{sc}(u, s)} \tag{3}$$

The main calculation parameters in the formulas include:

- $T_{sc}(u)$ — all substructures after the decomposition of the user answers u;
- $T_{sc}(s)$ — all substructures after the decomposition of the standard answers s;
- $\otimes$ — binary matching operator, which represents the number of matching substructures in the set of two substructures.

Next, we will introduce the working algorithm of this sc-agent in detail:

Algorithm 1 — The working algorithm of sc-agent for computing semantic similarity of factual knowledge

Input: The specific objective question and the corresponding semantic graph of standard answer and the semantic graph of user answer. The condition of the sc-agent response is that two semantic graphs that use factual knowledge to represent the answer appear in the sc-memory, and the similarity between them has not been calculated.

Output: The precision, recall and similarity between answers, and the sc-node used to record the matching status of substructures.

1) checking whether the standard answer and user answer exist at the same time, if so, go to step 2), otherwise, go to step 10);
2) according to the rules of factual knowledge representation (various types of sc-constructions), the semantic graphs of standard answers and user answers are decomposed into substructures [4];
3) iteratively traverse each substructure of the standard answer and user answer, classify them according to the type of substructure (three element sc-construction, five element sc-construction, etc.), and count the number of all substructures;
4) one type of substructure is randomly selected from the set of recorded standard answer substructure types;

5) according to the standard answer substructure type selected in step 4), a corresponding type of substructure is selected from the set of recorded user answer substructure types;
6) iteratively compare each substructure with the same substructure type between the standard answer and the user answer, and record the number of matched substructures and the matched substructures. The criterion for judging the matching of the same type of substructures is that the sc-nodes at the corresponding positions between the two substructures have the same main identifier. If the substructure contains sc-links, the contents of the sc-links at the corresponding positions must be also the same;
7) repeat step 4 — step 6 until all types of substructures have been traversed;
8) using formulas (1), (2), (3) calculate precision, recall and similarity, and generate semantic fragments for recording sc-agent running results;
9) removing all temporary sc-elements created while the sc-agent is running;
10) exit the program.

B. Sc-agent for processing semantic similarity calculation results of factual knowledge

The sc-agent for computing semantic similarity of factual knowledge only calculates the precision, recall and similarity between the standard answer and the user answer to the objective questions. However, because some objective questions have multiple correct options, it is necessary to comprehensively consider the precision, recall and similarity to fully judge the correctness of a question. Therefore, the sc-agent for processing semantic similarity calculation results of factual knowledge is developed in this article, its main function is to further judge the correctness and completeness of the current objective question based on the three information measurement parameters obtained in the previous step and the specific types and characteristics of the objective question [4].

The sc-agent for processing semantic similarity calculation results of factual knowledge needs to complete the following tasks:

1) judging the correctness and completeness of current question according to the precision, recall and similarity, as well as the evaluation strategies for the correctness and completeness of objective questions;
2) according to the correctness and completeness of the current question, generate some semantic fragments for prompting users;

The evaluation strategies for the correctness and completeness of objective questions mainly include the following:

- if the current question has the only correct option (multiple-choice questions with a correct option, judgment questions, and partially fill in the blank questions), then only the standard answer and the user answer exactly match, that is, the similarity is equal to 1 ($F_{sc} = 1$), the question is considered correct, otherwise the question is incorrect. The answer options in the semantic graph are described using the three element sc-construction;
- if the current question has multiple correct options (multiple-choice question with multiple correct options and partially fill in the blank questions), it can be subdivided into the following situations for judgment:
  - as long as the user answer contains the wrong option, the question is considered wrong. According to the definition of formulas (1), (2), (3), the similarity and precision are both less than 1 at this time ($F_{sc} < 1$ and $P_{sc} < 1$);
  - the all options included in the user answer are correct, but the number of correct options is less than the number of correct options in the standard answer, then the question is considered partially correct and incomplete. In this case, the precision is equal to 1, the similarity is less than 1, and the ratio of the number of all options in the user answer to the number of all options in the standard answer is the recall ($F_{sc} < 1$ and $P_{sc} = 1$);
  - if the options in the standard answer exactly match the options in the user answer, then the question is completely correct and complete. At this time, the similarity is equal to 1 ($F_{sc} = 1$);

Next, we will introduce the working algorithm of this sc-agent in detail:

Algorithm 2 — The working algorithm of sc-agent for processing semantic similarity calculation results of factual knowledge

Input: The semantic fragments of specific objective question, as well as the precision, recall and similarity between answers. The condition of the sc-agent response is that the similarity between the semantic graphs of the answers described using factual knowledge has been calculated, but the correctness and completeness of the answers have not been judged.

Output: The final answer verification result of a specific objective question, and the necessary semantic fragments used to display the answer verification result.

1) checking whether all input parameters used for sc-agent work meet the conditions, if so, go to step 2), otherwise, go to step 5);
2) according to the evaluation strategies for the correctness and completeness of objective questions, combined with the precision, recall, and similarity between answers, verify the correctness and completeness of specific objective question;
3) generating semantic fragments used to record the execution results of sc-agent;
4) removing all temporary sc-elements created while the sc-agent is running;
5) exit the program.

Combining sc-agent for computing semantic similarity of factual knowledge and sc-agent for processing semantic similarity calculation results of factual knowledge can verify the correctness and completeness of any type of objective question. Fig. 1 shows an example of using the problem solver to automatically verify the correctness and completeness of the answers to the multiple-choice questions in SCg-code (SCg-code is a graphical version for the external visual representation of SC-code) [1], [6].

C. Sc-agent for computing semantic similarity of logical knowledge

The basic function of the sc-agent for computing semantic similarity of logical knowledge is to calculate the similarity between semantic graphs described by logical knowledge. Because the answers to subjective questions in the form of semantic graphs in the knowledge base are described in the form of logical formula using logical knowledge (SCL-code), the similarity between the answers can be calculated using this sc-agent [1], [4]. Usually, the answers to subjective questions are not unique, so the similarity between answers becomes a key indicator for evaluating the correctness and completeness of user answers to subjective questions. Among them, user answers in natural language can be converted into SCL-code either manually or through natural language interfaces. Before

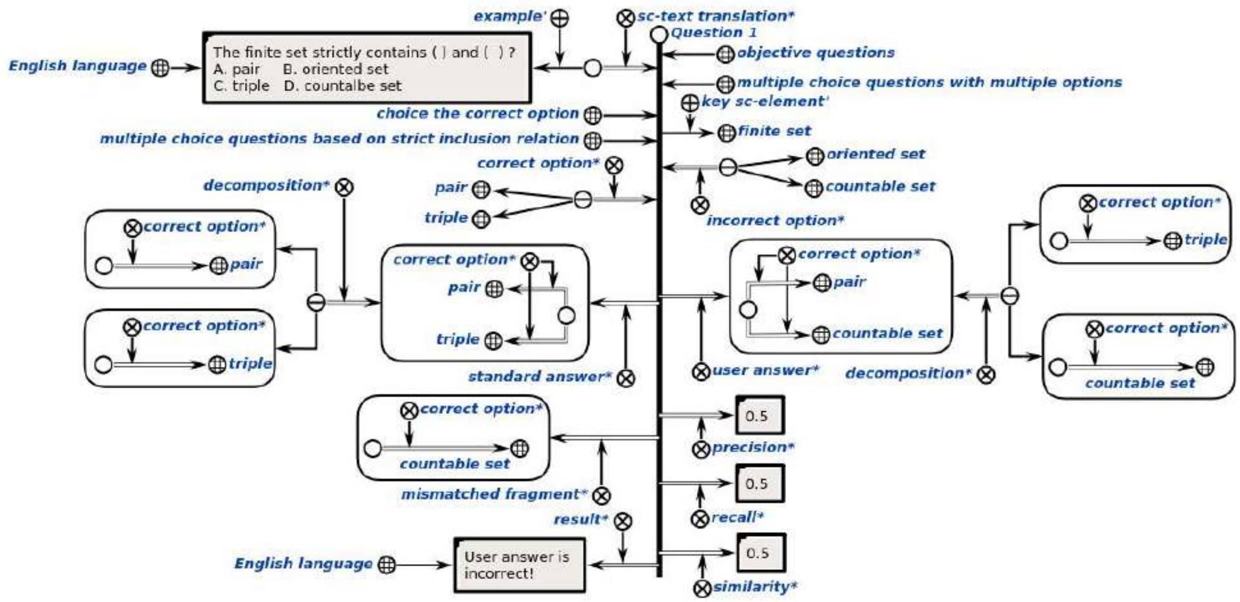


Figure 1. An example of automatic verification of answers to the multiple-choice questions

the subjective question answer verification, we assume that the factual knowledge contained in the user answer (for example, the constant sc-nodes used to represent concepts or relations) has been aligned with the existing knowledge in the knowledge base (through natural language interface) [5]. Therefore, in this article, the similarity between logical knowledge is calculated based on semantic and logical knowledge representation structures;

The sc-agent for computing semantic similarity of logical knowledge needs to complete the following tasks:

1) automatic selection of potential equivalent standard answer;
2) according to the representation rules of logical knowledge, the standard answer and user answer in the form of semantic graphs are decomposed into substructures;
3) establishing the mapping relationship of potential equivalent variable sc-node pairs between the semantic graph of the standard answer and the semantic graph of the user answer;
4) using formulas (1), (2), and (3) to calculate the precision P_{sc}, recall R_{sc} and similarity F_{sc} between semantic graphs.

Automatic selection of potential equivalent standard answer

Because some subjective questions usually have multiple standard answers (pre-stored in the knowledge base), and between the logic formulas used to formalize these answers do not satisfy logical equivalence. For example, the definition of equivalence relation: 1. in mathematics, an equivalence relation is a binary relation that is reflexive, symmetric and transitive; 2. for any binary relationship, if it is a tolerant relationship and is transitive, then it is an equivalence relation. Therefore, when calculating the similarity between answers, it is necessary to filter a standard answer that best matches the user answer from multiple possible standard answers in advance, and then calculate the similarity between them.

Because the answers to the subjective question in the knowledge base of the ostis-systems are described by logical knowledge in the form of logical formula. Therefore, if there are multiple standard answers to a question, and the logic formulas between them do not satisfy the equivalence, we find that the biggest difference between these answers is that the predicates used to describe them are different (that is, the constant sc-nodes used to represent concepts, relations, and elements in different answers are different) [10]. Therefore, this article proposes an approach to filtering the standard answer that best matches the user answer according to the similarity between all the predicates in the standard answer and all the predicates in the user answer.

The approach to filtering the standard answer that best matches the user answer according to the similarity between the predicates mainly includes the following steps:

1) if sc-agent judges that there are multiple standard answers to the current question, it will find all non-repeated predicates in each answer (the constant sc-nodes used to represent concepts, relations, and elements in the answer);
2) using formulas (1), (2), and (3) to iteratively calculate the similarity between all the predicates in the user answer and all the predicates in each standard answer. Here, the parameters in the formulas need to be given new meanings.
   - $T_{sc}(u)$ — all non-repeated predicates in the user answer u;
   - $T_{sc}(s)$ — all non-repeated predicates in the standard answer s;
   - $\otimes$ — binary matching operator, which represents the number of matching between all the predicates in the user answer and all the predicates in the standard answer.
3) choosing the standard answer with the greatest similarity to the user answer as the final standard answer.

Fig. 2 shows an example of filtering a standard answer in advance that best matches the user answer according to the predicate similarity between answers in SCg-code.

The establishment of mapping relationship of the potential equivalent variable sc-node pairs between answers

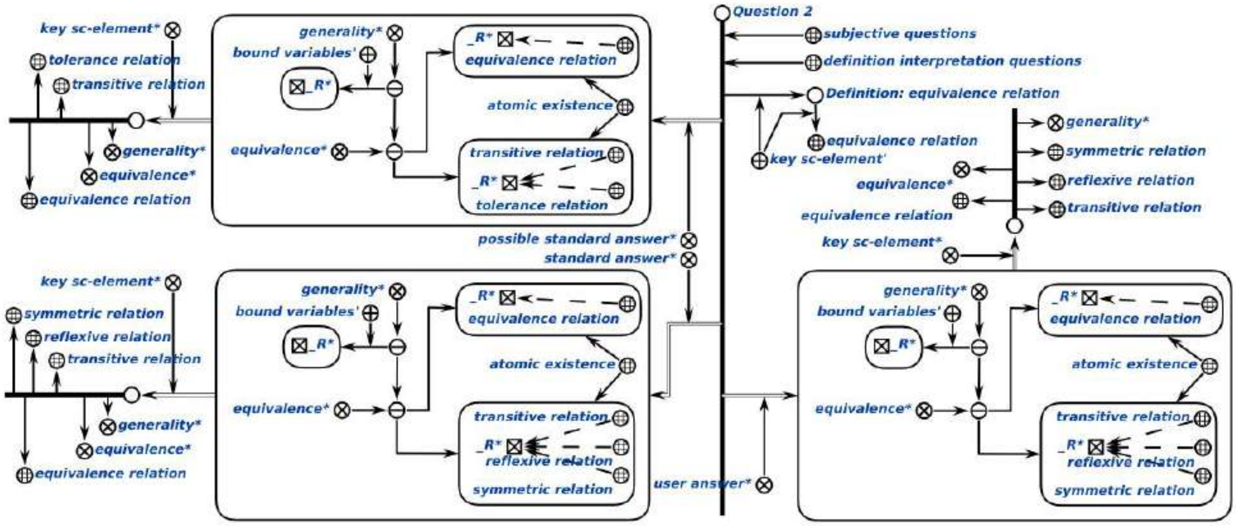


Figure 2. An example of filtering a standard answer that best matches the user answer according to the predicate similarity between answers

As we have already introduced, since the semantic graphs in the knowledge base used to describe the answers to subjective questions contain variables sc-nodes (equivalent to the bound variables in the predicate logic formula), when calculating the similarity between answers, the most critical step is to establish the mapping relationship (injection) of potential equivalent variable sc-node pairs between answers [11], [14]. Therefore, based on the existing ontology mapping methods, this article proposes an approach to establish the mapping relationship of potential equivalent variable sc-node pairs based on the semantic structures (various sc-constructions). In order to establish the mapping relationship of potential equivalent variable sc-node pairs, the following problems need to be solved first:

1) first, the position of the variable sc-nodes in the semantic graph needs to be determined;
2) it is necessary to determine the semantic connotation of the variable sc-nodes in the semantic graph.

Usually any predicate logic formula can be seen as consisting of two parts: 1. connective (such as negation ($\neg$) and implication ($\rightarrow$), etc.) used to describe logical relations, and quantifiers (universal quantifier ($\forall$) and existential quantifier ($\exists$)) used to carve the arbitrariness and existence of bound variables; 2. atomic predicate formula that uses predicates to describe variable attributes or relationships between variables. The predicate formulas used to formalize the answer in this article do not include free variables, so the variables in this article specifically refer to bound variables [4], [10]. Because the semantic graph used to describe the answer in the knowledge base is constructed strictly according to the logic formula. Therefore, the semantic graph used to represent the answer can also be regarded as composed of these two parts.

In the ostis-systems, the sc-construction composed of sc-tuple, relation sc-node, role relation sc-node and sc-connector is used to describe logical connectives and quantifiers, atomic predicate formula or multiple atomic predicate formulas that satisfy conjunctive relation are contained in the sc-structure and connected with the corresponding sc-tuple, and these sc-elements together constitute the semantic graph used to represent the answer. The atomic predicate formula is described using various sc-constructions. In the semantic graph, all sc-tuples and sc-connectors form a tree, which completely describes the logical sequence between connectives and quantifiers in the predicate formula. Because the sc-structure containing the atomic predicate formula is connected to the corresponding sc-tuple, as long as the position of each sc-tuple and sc-structure in the semantic graph is determined, the position of each variable sc-node in the semantic graph can be determined. In order to determine the position of each variable sc-node in the semantic graph, this article proposes an approach to numbering each sc-tuple and sc-structure in the semantic graph according to a depth-first search strategy (DFS). The approach mainly includes the following steps:

1) starting from the root of the tree structure composed of sc-tuples, each sc-tuple node in the tree is numbered in turn according to the DFS strategy (the numbering sequence starts from 0). If some nodes in the tree have multiple child nodes, the sub-trees with these child nodes as the root are sequentially numbered according to the node priority specified below;
   - if the child nodes and the parent node constitute the semantic structure that expresses the implication relation, then the priority of the conditional node (this node is connected with the parent node using a role relation " if' ") is greater than the priority of the conclusion node. That is, the node representing the condition and the corresponding sub-tree priority are numbered according to the DFS strategy;
   - if a node has multiple child nodes, and there is a node representing negative connective in the child nodes, then the priority of this node is higher than other nodes. If there are multiple nodes representing negative connective in the child nodes, the priority between them is related to the height of their corresponding sub-tree, and the higher the height of the sub-tree, the greater the priority;
   - in the current version, for other situations, a node is randomly selected for numbering.
2) according to the numbering sequence of sc-tuple, each sc-tuple in the tree is traversed from small to large, and the sc-structure connected to the current sc-tuple is numbered while traversing (the numbering sequence starts from 1). If there are multiple sc-structures connected

to the same sc-tuple, the sc-structure will be numbered according to the priority specified below.

- if there are multiple sc-structures connected to sc-tuple that represents universal quantifier or existential quantifier, the sc-structure that only contains variables is numbered preferentially;
- if there are multiple sc-structures connected to the sc-tuple that represents the implication relation, the sc-structure representing the condition is numbered preferentially;
- in the current version, for other situations, the numbering is based on the number of elements contained in the sc-structure. The fewer the number of elements contained in the sc-structure, the numbering will be given priority.

Because the atomic predicate formula or the conjunctive formula of the atomic predicate formula is included in the sc-structure, once the position of the sc-structure in the semantic graph is determined, the position of each atomic predicate formula in the semantic graph can be determined indirectly. In answer verification, if the standard answer and the user answer are exactly equal, it means that the atomic predicate formulas with the same semantics between the answers have the same position in the semantic graph (That is, the numbering sequence of sc-structure is the same). In the ostis-systems, the atomic predicate formula is expressed using various sc-constructions, so this article will establish the mapping relationship of potential equivalent variable sc-node pairs between the answers according to the matching relationship of the sc-constructions in the same position between the answers [4], [8]. The establishment of mapping relationship of the potential equivalent variable sc-node pairs between answers mainly includes the following steps:

1) according to the numbering sequence of the sc-structure in the semantic graph, each time a sc-structure pair with the same number is found from the standard answer and the user answer;
2) according to the priority order (from high to low) of the various types of sc-constructions used to describe the atomic predicate formula, it is determined in turn whether the current sc-structure pair contains this type of sc-construction at the same time. If the current sc-structure pair contains this type of sc-construction at the same time, then, according to the matching relationship of each sc-element between the current sc-construction in the standard answer and the current sc-construction in the user answer, the mapping relationship of the potential equivalent variable sc-node pairs between the current sc-construction pair is established. The priority order of various types of sc-constructions, and the criteria for judging whether the same type of sc-constructions match are as follows:
   - because there may be multiple sc-constructions in the same sc-structure, in order to ensure the uniqueness and accuracy of the mapping relationship of the potential equivalent variables sc-node pairs between sc-constructions, this article proposes to establish the mapping relationship between variables sc-nodes according to the priority order of sc-constructions. There are 14 types of sc-constructions in the current version, and the order of priority between them is determined by the number of sc-nodes contained in the sc-construction. The greater the number of sc-nodes, the higher the priority. If the number of sc-nodes is the same, it is determined according to the number of variable sc-nodes, the greater the number of variable sc-nodes, the higher the priority;
   - the criteria for judging the matching of the same type of sc-constructions are: 1. the constant sc-node at the corresponding position between them exactly matches; 2. there is a mapping relationship between the variable sc-nodes of the corresponding position, or there is no mapping relationship between the corresponding position variable sc-nodes, and there is no mapping relationship between these variables sc-nodes and other variables sc-nodes. If any pair of sc-constructions of the same type in the same position between the answers satisfy the above two conditions at the same time, the mapping relationship between the corresponding position variables sc-nodes between the sc-constructions is established (if there is already a mapping relationship between the two variable sc-nodes, it will not be created repeatedly).
3) repeat step 1 — step 2 until all potential equivalent variable sc-node pairs between semantic graphs have established a mapping relationship.

Fig. 3 shows an example of establishing the mapping relationship of potential equivalent variable sc-node pairs between semantic graphs according to the numbering order of sc-structures in SCg-code.

In Fig. 3, the definition of the inclusion relation is described in the form of a semantic graph $(\forall A \forall B(A \subseteq B) \Longleftrightarrow (\forall a(a \in A \rightarrow a \in B)))$.

When the mapping relationship between the potential equivalent variable sc-node pairs between the semantic graphs is established according to the positions of sc-tuples and sc-structures in the semantic graphs, the similarity between the answers can be calculated using formulas (1), (2), and (3). The criteria for judging the matching of substructures are: 1. the constant sc-nodes in the corresponding position between substructures have the same main identifier in the knowledge base or the same number in the semantic graphs (sc-tuple and sc-structure); 2. there is a mapping relationship between the variable sc-nodes at the corresponding position between the substructures.

Next, we will introduce the working algorithm of this sc-agent in detail:

Algorithm 3 — The working algorithm of sc-agent for computing semantic similarity of logical knowledge

Input: The specific subjective question and the corresponding semantic graph of standard answer and the semantic graph of user answer. The condition of the sc-agent response is that two semantic graphs that use logical knowledge to represent the answer appear in the sc-memory, and the similarity between them has not been calculated.

Output: The precision, recall and similarity between answers, and the sc-node used to record the matching status of substructures.

1) checking whether all input parameters used for sc-agent work meet the conditions, if so, go to step 2), otherwise,go to step 12);
2) checking whether the current question has multiple standard answers, if so, automatically select a standard answer that best matches the user answer according to the approach introduced earlier;
3) according to the rules of logical knowledge representation, the semantic graphs of standard answers and user answers are decomposed into substructures;
4) the sc-tuples and sc-structures in the semantic graph of the standard answer and the semantic graph of the

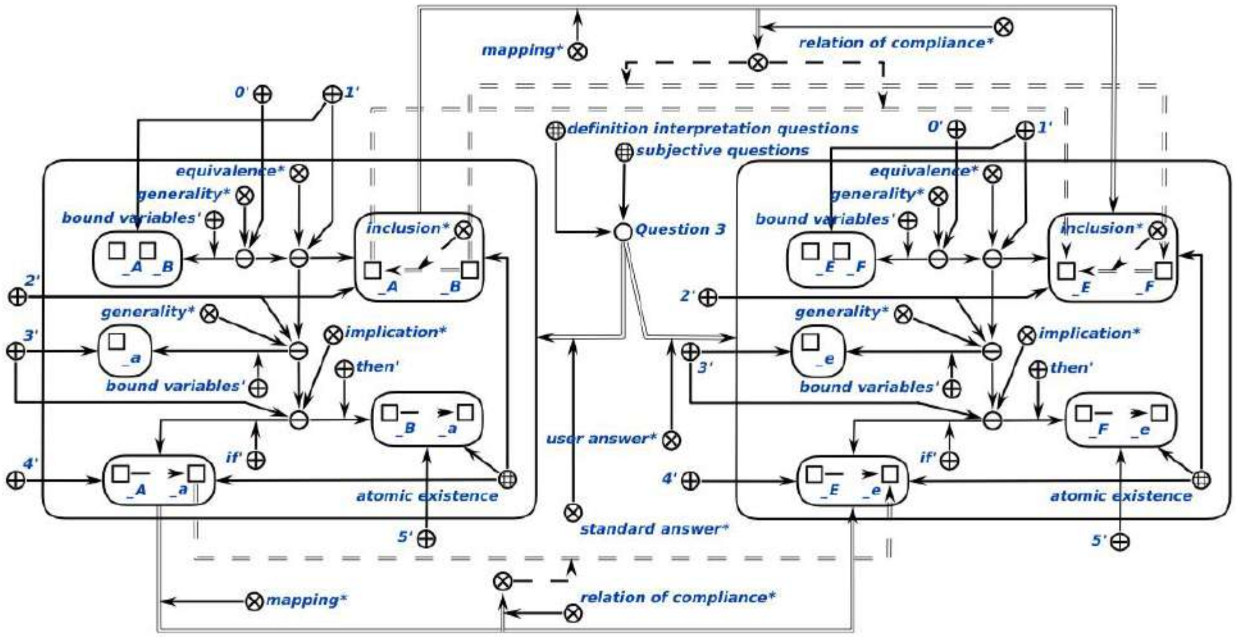


Figure 3. An example of establishing the mapping relationship of potential equivalent variable sc-node pairs between semantic graphs

user answer are numbered respectively, and the mapping relationship of potential equivalent variable sc-node pairs between answers is established;

5) iteratively traverse each substructure of the standard answer and user answer, classify them according to the type of substructure, and count the number of all substructures;
6) one type of substructure is randomly selected from the set of recorded standard answer substructure types;
7) according to the standard answer substructure type selected in step 6), a corresponding type of substructure is selected from the set of recorded user answer substructure types;
8) iteratively compare each substructure with the same substructure type between the standard answer and the user answer, and record the number of matched substructures and the matched substructures.
9) repeat step 6 — step 8 until all types of substructures have been traversed;
10) using formulas (1), (2), (3) calculate precision, recall and similarity, and generate semantic fragments for recording sc-agent running results;
11) removing all temporary sc-elements created while the sc-agent is running;
12) exit the program.

When the precision, recall and similarity between the answers to the subjective questions are obtained, the completeness and correctness of the user answers can be evaluated. According to the similarity, user answers are divided into the following situations:

- if the similarity is equal to 1, the user answer is completely correct ($F_{sc} = 1$);
- if the similarity is greater than 0 and less than 1 ($0 < F_{sc} < 1$), there may be two situations:
  - the user answer is partially correct and incomplete (the default mode of the current version);
  - the logical formulas used to formalize standard answers and user answers may satisfy logical equivalence (in the following article, we will introduce in detail the approach to judging the equivalence between answers based on predicate logic).
- if the similarity is equal to 0 ($F_{sc} = 0$), the user answer is completely wrong.

V. Conclusion and further work

Automatic answer verification is one of the most basic functions of ITS, which can quickly check the user mastery of new knowledge and improve the user learning efficiency. Therefore, this article introduces in detail an approach to developing a problem solver for automatic answer verification in the ITS developed using OSTIS Technology. The developed problem solver can not only verify the correctness and completeness of the answer to the subjective question, but also the correctness and completeness of the answer to the objective question. Because the problem solver is developed based on multi-agent technology, sc-agent for computing semantic similarity of factual knowledge, sc-agent for processing semantic similarity calculation results of factual knowledge, and sc-agent for computing semantic similarity of logical knowledge are developed in this article respectively. The developed problem solver completes the answer verification by combining different sc-agents according to the type of the question.

The developed problem solver for automatic answer verification in this article has the following advantages:

- verify the correctness and completeness of user answers based on semantics;
- by calling different sc-agents, the similarity between any two semantic fragments in the knowledge base can be calculated;
- because the problem solver is developed based on multi-agent technology, it is easy to add new functions;
- because the ostis-systems developed for different application fields have the same knowledge representation

approach and knowledge processing model, the problem solver developed in this article can be easily transplanted to other ostis-systems;

In future work, we hope to automatically generate some comments and suggestions by analyzing the verification results of user answers.

Acknowledgment

The work in this article was done with the support of research teams of the Department of Intelligent Information Technologies of Belarusian State University of Informatics and Radioelectronics. Authors would like to thank every researcher in the Department of Intelligent Information Technologies.

References

[1] V. V. Golenkov and N. A. Guljakina, "Proekt otkrytoj semanticheskoj tehnologii komponentnogo proektirovanija intellektual'nyh sistem. chast' 1: Principy sozdanija project of open semantic technology for component design of intelligent systems. part 1: Creation principles]," Ontologija proektirovanija [Ontology of design], no. 1, pp. 42–64, 2014.

[2] V. Golenkov, N. Guliakina, I. Davydenko, and A. Eremeev, "Methods and tools for ensuring compatibility of computer systems," in Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems], V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2019, pp. 25–52.

[3] Xu G. P., Zeng W. H., Huang C. L. Research on intelligent tutoring system. Application research of computers, 2009, Vol. 26(11), pp. 4020-4030.

[4] Li W., Grakova N., Qian L. Ontological Approach for Question Generation and Knowledge Control. Communications in Computer and Information Science, 2020, Vol. 1282, pp. 161-175.

[5] Qian L., Sadouski M., Li W. Ontological Approach for Chinese Language Interface Design. Communications in Computer and Information Science, 2020, Vol. 1282, pp. 146-160.

[6] (2021, JAN) Ims.ostis metasystem. [Online]. Available: https://ims.ostis.net

[7] Li X. J. Realization of automatic scoring algorithm for subjective questions based on artificial intelligence. Journal of Jiangnan University (Natural Science Edition), 2009, Vol. 08(03), pp. 292-295.

[8] Peter A., Basura F., Mark J. SPICE: Semantic Propositional Image Caption Evaluation. Computer Vision and Pattern Recognition (cs.CV), 2016.

[9] Zhang Z. P., Zhao H. L., Tian S. X. Ontology integration method based on RDFS. Computer Engineering and Applications, 2008, Vol. 44(15), pp. 131-141.

[10] Pan M. Q., Ding Z. J. A Simple Method for Solving Pyrenex Disjunction(Conjunction) Normal Forms. Computer Engineering and Science, 2013, Vol. 30(10), pp. 80-84.

[11] Wan H. R., Yang Y. H., Deng F. Review on Research Progress of Text Similarity Calculation. Journal of Beijing Information Science (Technology University), 2019, Vol. 34(01), pp. 68-74.

[12] Zhu J. Z., Qiao J. Z., Lin S. K. Entity Alignment Algorithm for Knowledge Graph of Representation Learning. Journal of Northeastern University (Natural Science), 2018, Vol. 11(39), pp. 1535-1539.

[13] Socher R., Chen D., Manning C. D., et al. Reasoning with neural tensor networks for knowledge base completion. Advances in Neural Information Processing Systems, 2013, pp. 926-934.

[14] Su J. L., Wang Y. Z., Jin X. L., et al. Knowledge Graph Entity Alignment with Semantic and Structural Information. Journal of Shanxi University(Nat. Sci. Ed.), 2018, Vol. 42(1), pp. 23-30.

[15] Zhuang Y., Li G. L., Feng J. H. A Survey Entity Alignment of Knowledge Base. Journal of Computer Research and Development, 2016, Vol. 53(1), pp. 165-192.

[16] Wang X. Y., Hu Z. W., Bai R. J., et al. Review on Concepts, Processes, Tools and Methods Ontology Integration. Library and Information Service, 2011, Vol. 55(16), pp. 119-125.

Разработка решателя задач для автоматической проверки ответов в интеллектуальных обучающих системах

Ли Вэньцзу, Цянь Лунвэй

В данной работе предложен подход к разработке решателя задач для автоматической проверки ответов в интеллектуальных обучающих системах, построенных с использованием Технологии OSTIS. Решатель задач разработан на основе многоагентного подхода к обработке информации. Разработанный решатель задач автоматически проверяет правильность и полноту ответа пользователя на семантическом уровне, используя соответствующие наборы sc-агентов в соответствии с типом вопросов (вопросы на выбор, вопросы на заполнение пробелов, вопросы на толкование определений и т. д.). В ситуации, когда может существовать несколько стандартных ответов (например, для вопросов на толкование определений), решатель задач может автоматически заранее отфильтровать фрагмент стандартного ответа, который наилучшим образом соответствует ответу пользователя, а затем продолжить проверку правильности и полноты между ними.

Keywords—решатель задач, проверка ответов, технология OSTIS, интеллектуальные обучающие системы, онтология, база знаний

С развитием современных информационных технологий искусственный интеллект как важная часть современных компьютерных приложений быстро применяется в сфере образования. Применение технологий искусственного интеллекта в сфере образования может не только предоставить новые методы и инструменты обучения, но и позволить учащимся тратить меньше времени на приобретение более полезных знаний, эффективно повышая точность поиска информации и эффективность отбора знаний. В то же время сочетание технологий искусственного интеллекта и образовательного процесса также является одним из важных средств обеспечения справедливости образования, чтобы каждый учащийся имел равные возможности для получения знаний. Особенно после вспышки COVID-19 интеллектуальные обучающие системы (ИОС) играют все более важную роль в дистанционном образовании. По мере того как разрабатывается все больше и больше ИОС, используемых в различных областях, большинство разработчиков считают, что ИОС необходимо удовлетворять, по крайней мере, следующим основным функциям:

- автоматическая проверка ответов, если обнаружена ошибка, необходимо проанализировать причину ошибки и принять корректирующие меры;
- в соответствии с уровнем обучающегося и ситуацией обучения автоматически корректируется содержание и прогресс обучения;
- автоматическая генерация различных вопросов и упражнений;
- обладая способностью генерировать и понимать естественный язык, а также осуществлять относительно свободный человеко-машинный разговор;
- наличие способности объяснять содержание обучения;
- автоматическое решение вопросов на основе понимания содержания обучения.

Являясь наиболее основной и важной функцией ИОС, автоматическая проверка ответов может быстро проверить усвоение пользователем новых знаний и значительно повысить эффективность обучения пользователя, позволяя пользователям получить больше знаний за ограниченное время.

Received 01.06.21

Application of an integration platform for ontological model-based problem solving using an unified semantic knowledge representation

Valerian Ivashenko
Department of Intellectual Information Technologies
Belarusion State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
ivashenko@bsuir.by

***Abstract*—This article describes a solution in the form of an intelligent integration platform based on the model of an unified semantic knowledge representation for the development of applied knowledge-driven systems. The model of an unified semantic knowledge representation using semantic networks, models and methods of measure and probability theories, methods of descrete optimization and applied mathematics, computer simulation and multi-agent approaches were used. The purpose is to develop computer tools with cognitive architecture relying on elements of artificial consciousness and being able to communicate and to be flexible and adaptive in complex educational applications. Virtual machines, subsystems of integration platform and tutoring applied multi-agent software system were developed and implemented as part of human-machine interaction system.**

***Keywords*—artificial intelligence systems integration, integration platform, multi-agent system, knowledge-driven system, knowledge processing model, unified semantic knowledge processing model**

I. Introduction

The basis for the success of a learning intellectual system is integration openness [42], [45]. When it is possible to integrate not only new knowledge related to different types but also various mechanisms for solving problems. Integration is one of the important understanding mechanisms for knowledge systems allowing the acquisition and improvement of problem-solving skills which is important for knowledge-driven systems [42], [45]. This article presents a solution in the form of an integration platform based on the unified semantic knowledge representation model. This presentation attempts to answer the following questions.

1 What goals can be achieved using such platforms?
2 What developments are already in this direction?
3 What are the architecture, mechanisms and rules for using the platform?
4 What are the positive and negative peculiarities of the platform?
5 What results have been achieved in the process of its application?
6 What are the perspectives for the development of the platform?

The general goals planned to be achieved are:

- creation of dynamically updating knowledge-based system able to accept new knowledge via machine learning and high-level or natural language communication;
- creation of scalable knowledge-driven systems maintaining big knowledge and large scale integrated ontology;
- creation of artificial consciousness systems which are self-descriptive and introspective;
- creation of multi-agent distributed applied intellectual systems.

II. Overview of models and approaches

The necessity of artificial intelligence integration historically caused by the existence of separate artificial intelligence solutions for specific problems such as reasoning knowledge, speech synthesis and recognition, computer vision problems. General approaches to the integration of information systems: integration through translation with control passing, integration through interpretation or through communication without control passing. In the case of control passing each process or agent of system should store own state. Therefore, the corresponding storage is available for one control flow and both are shared by all processes or agents on the system.

From the other side, consciousness as a more advanced kind of intelligence is determined by social experience of natural language communication. Thus, by natural way, such communicative models as actor models [1], [5], [13] or multi-agent systems are the basis [6], [45] not only for concurrent computer system but artificial intelligence systems. Therefore, concurrent models are integration models. These models are divided into two classes. The first class is models without shared common memory or storage. The second class is models with shared common memory or storage. Such models and means as actor models [5], Communicating Sequential Processes [10], Calculus of Communicating Systems [12], pi-calculus [15], Algebra of Communicating Processes [14], Language of Temporal Ordering Specifications [16]–[18], reactive multi-agent systems [13] and so on [19]–[23] can be referenced with the first class of models. The models of the first class using message passing communication provide implementation basis of the collective or swarm intelligence.

Models of the second class is oriented to maintain shared storage. Shared storage is seemed to be capable to maintain large knowledge bases and ontologies to solve complex problems and to provide sophisticated processes control. The bright examples of such approach are blackboard architecture models and blackboard systems. Blackboard system is shared information resource with the own communication protocols and
or consistency model. Means oriented to use with blackboard system are: CORBA, MOSID, OpenAir, OAA and others [32], [34]–[36].

MOSID (Messaging Open Service Interface Definition (OSID)) is an Open Knowledge Initiative specification which provides a means of sending, subscribing and receiving messages. OSIDs are programmatic interfaces which comprise a Service Oriented Architecture for designing and building reusable and interoperable software [35].

CORBA is a standard can be used for AI systems integration [32]. CORBA enables software components written in multiple computer languages and running on multiple computers to interoperate. CORBA is developed by the Object Management Group. Similar standard developed by Microsoft was DCOM (Dynamic Common Object Model).

OpenAIR Protocol is a routing and communication protocol based on a publish-subscribe architecture. It is mean and environment («AIR») that allows numerous A.I. researchers to share code more effectively. This is mean for distributed multi-module systems. OpenAIR is oriented to be foundation for markup languages and its semantics of hardware-software interfacing including computing vision (as at CVML (Computer Vision Markup Language)), gesture recognition and generation and so on. OpenAIR Protocol follows similar principles and architecture as the CORBA.

OAA (Open Agent Architecture) is a hybrid architecture that relies on a special inter-agent communication language (ICL) [36]. ICL is a logic based declarative language adopted to express high-level, complex tasks being close to natural language expressions.

There are implemented systems which use models of the second class. Psyclone AIOS is an implementation of a blackboard system that supports the OpenAIR message protocol [34]. It is a software platform, or an AI operating system (AIOS), developed by Communicative Machines Laboratories for use in creating large, multimodal artificial intelligence systems. Elvin is a content-based router with a central routing station, similar to the Psyclone AIOS [33].

Examples of applied integrated systems based on this approach are such robots and humanoids as MIRAGE, ASIMO, QRIO, Cog, AIBO and etc.

The important aspect of models is possibility of process introspection including methods of process mining for the purpose of inductive programming using action languages. Such languages as LTML (Learnable Task Modeling Language), PDDL (Planning Domain Definition Language) [7]–[9] and MAPL (Multi-Agent Planning Language) [6] can be considered as means of plan specification of artificial intelligence systems including concurrent and multi-agent systems.

The next approach can be viewed as pragmatic or problem specific approach. This approach concentrates on developing cognitive architectures. Examples of cognitive architectures are: 4D-RCS (Real-time control system) Reference Model Architecture [29], SOAR (State Operator And Result) cognitive architecture [30], architecture of Hierarchical temporal memory [31] and many others. The models investigated in the range of OSTIS projects can be also referred to this last approach [41].

Another approaches and models are determined by history of development of computing systems for artificial intelligence. These include architectures of LISP-machines (Connection machines), PROLOG-machines and machine learning processors and accelerators.

However, these architectures and models have restricted capabilities of knowledge integration via unified representation including limitations to deal with NON-factors of knowledge [42], [48], to introspect processes using semantic logging and to combine knowledge declarative semantics with operational semantics of synchronous and asynchronous knowledge processing using various knowledge representation languages.

III. Integration platform description

Proposed integration platform mainly concentrated on integration via translation and via communication. While integration via interpretation limited by languages of unified semantic knowledge representation model and languages supported by model used to implement this platform.

Specification is main part of self-descriptive and introspective systems such as knowledge-driving systems. There are several models and means to specify and implement such discrete systems as abstract machines or information processing models including concurrent systems. These are transition systems, actor models, Communicating Sequential Processes, Calculus of Communicating Systems, pi-calculus, Algebra of Communicating Processes, Language of Temporal Ordering Specifications, temporal logics and others [1]–[4], [11]–[15], [17], [18], [28], [42]. Proposed knowledge specification model [42] is the key mean to specify knowledge represented with the unified semantic knowledge representation model [42], [45].

Knowledge processing model [42] describes dynamic of knowledge accumulation and optimization processes. To define knowledge processing model seven components need to be specified: alphabet, language, syntactic relations, initial states, interpretations, operations and semantic metric [42]. Every finite structure can be specified with its formal ontology model. Languages are specified by formal ontology model relations mapping its texts into their representations in its other texts and by the relation between formal ontology models of allowed and forbidden syntactic structure and formal ontology models of representations of language texts in its other texts. Initial states are specified as sublanguages. Infinite number of initial states can also be specified by the knowledge specification models relation which specify (generative) initial states order with a finite number of primary initial states (information constructions or language texts). Language syntactic (incidence) relation is specified as a language via its texts representations. Language semantics specification consists of denotational semantics and operational semantics specifications [1]–[4], [13]. Denotational semantics is specified by the knowledge specification model relation between formal ontology models of language text fragments and formal ontology model of finite subsets of its denotations. Such relation specify whatever mapping which is projection of interpretation of language text fragments on finite sets of denotations. Operational semantics is specified by the knowledge specification model relation between formal ontology models of reifications of projections of language text fragments interpretations on finite sets of denotations in two situations. In case of reflexive semantics, denotation semantics specification can express syntactic restrictions for forbidden syntactic structures in the vicinity of language key elements. The allowed syntactic structures are all which are described in denotational semantic specifications, any other structures are forbidden. Thus, we get specifications of language key elements. Semantic metric can be specified by mapping language texts to metric space. Simple semantic metric is specified by mapping language texts to binary logic scale σ with exclusive disjunction as a metric operation.

$$\sigma \in \{\Lambda\} \times \{\bot, \top\}^{\Lambda} \times \{\{\bot, \top\}\} \tag{1}$$

$$\psi\left(\sigma_2(\alpha) \veebar \sigma_2(\beta)\right) \tag{2}$$

$$\psi(\chi) \underset{def}{=} \begin{cases} 0 \,|(\neg\chi) \\ 1 \,|\chi \end{cases} \tag{3}$$

The equivalence can be also considered as a semantic metric operation due the isomorphism existence between equivalence and exclusive disjunction.

$$\psi\left(\sigma_2(\alpha) \veebar \sigma_2(\beta)\right) = 1 - \psi\left(\sigma_2(\alpha) \sim \sigma_2(\beta)\right) \tag{4}$$

More sophisticated metrics may take into account syntactical and spatial-temporal structure of knowledge.

One knowledge processing system integrated by the another knowledge processing systems if some conditions fulfilled [27], [42], [46], [47]. The necessary conditions to integrate one knowledge processing system in the second is the existence of such text inclusion mapping π and bijection [27] i as

$$\begin{array}{l} \forall\rho\exists i(\rho)\left(\rho = i \circ \pi \circ i(\rho) \circ \pi^{-1} \circ i^{-1}\right) \\ \forall\rho\exists i(\rho)\left(\pi^{-1} \circ i^{-1} \circ \rho \subseteq i(\rho) \circ \pi^{-1} \circ i^{-1}\right) \\ \forall\rho\exists i(\rho)\left(i(\rho) = \pi^{-1} \circ i^{-1} \circ \rho \circ i \circ \pi\right) \\ \forall\rho\exists i(\rho)\left(i \circ \pi \circ i(\rho) \subseteq \rho \circ i \circ \pi\right) \end{array} \tag{5}$$

where π is a text mapping relation between text fragments and texts of the second knowledge processing model containing its; i is bijective integration mapping; ρ is an operation.

These can be shown with the next diagram.

$$\begin{array}{ccccc} \sigma & \overset{i}{\longleftrightarrow} & i(\sigma) & \overset{\pi}{\underset{\leftarrow}{\rightarrow}} & \tau \\ \rho\downarrow & & \overset{i}{\longleftrightarrow} & & i(\rho)\downarrow \\ \omega & \overset{i}{\longleftrightarrow} & i(\omega) & \overset{\pi}{\underset{\leftarrow}{\rightarrow}} & \gamma \end{array} \tag{6}$$

Architecture of implemented system is based on architecture of control levels for knowledge-driven systems [42]. These levels are: device control level, data control level, knowledge control level. The levels of control are disposed along the implementation hierarchy direction (vertical). While there are several sub-architectures which are placed along the communication direction (horizontal). These sub-architectures relate to abstract machines or information processing models which correspond to different variants of implementation subplatform. There are two virtual machines which implements core of the proposed integration platform: variety virtual machine and ontology virtual machine [37], [42], [45].

If an implementation is considered as a problem domain then the objects of the implementation are parts of the problem domain of this implementation. These objects, its kinds with the relations between them form a subject domain of the implementation. Any finite part of the subject domain of the implementation can be specified accordingly with the knowledge specification model. Various models are being specified depending on which objects are included in a corresponded part. Kinds of such models are situation structure model, system structure model, motion model, process model, device structure model, device process model, instruction set syntax model, instruction set semantic model, typed data structure representation model, typed data structure process model, typed knowledge structure syntax model, typed knowledge structure process model. From the point of view of architecture of a chosen implementation these models can be grouped in more complicated models. A distributed system can be represented with a set of communicating nodes. Each node can be realized as a computing machine. A machine realizes two or three levels of control. Machines encapsulate knowledge, data, and such devices as processors, memory and controllers. Each subsystem of a machine is specified with a model. Models are specifications of subsystems of implementation. Memory models, operation models, allocator memory models, synchronization (activation deactivation) models, access management models, reallocation memory models, number and strings processing models are more specialized models. Models of machines can share or include each of them [42], [45].

The platform was implemented with the virtual machines of two types: ontology virtual machine (knowledge processing machine) and variety virtual machine (user interface data processing machine) [37], [45]. The ontology virtual machine shares all three levels of control while variety virtual machine implements first two of them.

Subsystems of the ontology virtual machine specified with memory, allocator, allocation and reallocation, machine operation instruction subset, synchronization (including excitation and inhibition), access management, strings processing, unified semantic processing, communication and IO controller models [42], [45].

Subsystems of the variety virtual machine specified with memory, allocator, allocation and reallocation, machine operation instruction subset, strings processing, communication and IO controller models [37], [45].

All knowledge structure models match the unified semantic representation model [42].

All knowledge programming models (unified semantic processing) match knowledge processing model based on the unified semantic representation model. Operations of ontology virtual machine instructions are based on string processing model for knowledge-driven systems. They are operation both with such simple data types as numbers and also with strings. Strings are used as multiple nested stacks or meta-stacks for ontology virtual machine [42], [45].

Programs can be structured using sc -chains or sc -sequences. The lasts differ from the first by link membership connectives which not include a membership of contained element of the next link but include a membership of the next link to sequence set which contains all links and the membership of the first link. The structure of each instruction (command) or operator shares the METAPHORM [44] principles including the ordered pairing of input and output parameters and the encoding of operator types using either associative positioning (using key elements) or a structural morphological approach. The operator semantics of basic task types can be defined with systems of patterns (similar to semantic query language constructions [43]) for search and construction operators and sets of deleting elements for delete operators. Commands can be constructed during the processing of SEC (Semantic Execution Code) texts or their representations by sc -chains [42], [45].

Ontology virtual machines integrates operations with not only asynchronous but synchronous semantics. Synchronization is implemented by mechanism of inhibition and excitation phases. Thus, processes of the next excitation phase can not be started before finalizing all synchronous processes of previous excitation phase. Some processes can be triggered and suspended during inhibition phase. All ready results of each operation can be represented with an actual membership to a sc-set of ready results [42]. Thus, all events can be reduced to membership events. Processes generating these events are triggered by mechanism of inhibition and excitation phases.

Synchronous and asynchronous semantic specification is constructed using phenomenons description ontology for key elements with operation semantics and their semantic vicinities.

Knowledge access and control model also can be considered part of inhibition phase mechanism. Knowledge access and control model deals with agents, areas, modes and its types of access.

Multi-agent interaction model includes active structure reconfigurable (Semantic Code) memory maintaining synchronization mechanisms and models for semantic logging and multi-agent plan specifications languages [42]. The common interaction scheme for active structure reconfigurable memory is show on the figure below.

Multi-agent interaction is represented by events and relations between them in forms of phenomena, protocols and plans [42]. The spatial and temporal relations should be used to define spatial and temporal constraints for the plans. Unlike to MAPL constructions [6], these constrains do not use topology of the line of real numbers and the corresponding scale but the topology of events with the corresponding scale of rough sets over the lattice of their sets. Consistency of a multi-agent plan is determined by the several requirements. These are:

$$(N(m) \cap P(n)) \cup (P(m) \cap N(n)) = \emptyset \qquad (7)$$

where m is a membership event, while n is a non-membership event between same elements, N and P are lower and upper bounds membership functions of L-fuzzy rough sets (sc -set) [42]. In the case of defined clock measure μ and time duration δ consistency is

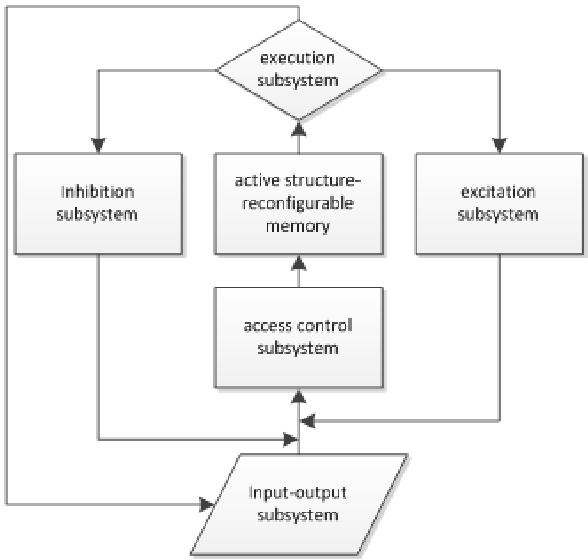


Figure 1. Scheme of subsystem interaction.

determined by the following requirements:

$$\begin{gathered}\delta(\emptyset)=0\\ \delta(S\cap T)\leqslant\delta(S)\\ \delta(S)\leqslant\delta(S\cup T)\\ \delta(S\cup T)\leqslant\delta(S)+\delta(T)\\ ((S\prec: T)\wedge(S\rhd\lhd T))\rightarrow(\delta(S\cup T)=\delta(S)+\delta(T))\end{gathered}\tag{8}$$

where $\prec:$ is temporal precedence relation, $\rhd\lhd$ is disjoint relation and S and T are phenomenons.

There are two general class of agents: internal agents operating only with active structure reconfigurable memory and external agents communicating via IO controllers or operating with variety virtual machine. More detailed specification of variety virtual machine is available at [37].

The general principles of knowledge processing [42] for the integration platform are:

- accounting of NON-factor;
- semantic logging;
- knowledge stream processing.

IV. Other approaches: competition and cooperation

Despite the universality of proposed integration platform and its knowledge representation capabilities, there are performance limitations as for its implementation as for a implementation of any other computer architecture, abstract machine or information processing model [42]. That is why there is a number of systems which can not be effectively implemented on the current implementation of the integration platform. These systems can be implemented separately and be able to concurrent with the platform in cooperative or competitive mode. The communication between integration platform and such external separative system is organized by the programming interfaces provided by the input-output subsystems (IO controllers) of the integration platform.

V. Application of integration platform

Ones of the most common classes of generalized problems are the searching, the choosing, the verifying, the constructing, the reconstructing and the destructing, while the searching, the constructing and the reconstructing are the most common classes of individual problems [42]. Problems of all classes can be composed using problems from the searching, the reconstruction and the destruction classes. There are agents for problems of each class [45]. Agents solving any cognition problem, i.e. search, choice or verification, are cognitive agents. Call the other agents performative agents. Depending on correspondence type between agent states and states of whole knowledge base these agents can solve external or internal problems. Therefore, there can be external and internal agents for searching, choosing and verifying. It is important to admit that an agent who solves the internal problem of search or another cognition problem may not be able to solve any cognition problem as an external problem. In the process of human-machine interaction tasks of different kinds arise. Consider the tasks arising in intelligent tutoring systems. Such systems are able to provide students with educational material in the reference system mode, from the other side there is a task to examine student knowledge. In the reference system mode there are three types of problems: wait user question (waiting is kind of the searching), retrieve information and output the result to the user. These problems can be solved correspondingly by agents of three types: external and internal search agents and external performative agent. All these agents solve task of navigation on tutorial materials. During navigation human-computer dialog contains interfaces with suggestions to user which can be interpreted as alternative questions. Each user reaction can be interpreted as answer on such questions. During a series user reactions on independent suggestions quantity of received information can be calculated as:

$$\left\lceil\left(\sum\nolimits_{i=1}^{q}\log_2(v(i))\right)-\sum\nolimits_{i=1}^{q}\log_2(t(i))\right\rceil\tag{9}$$

where $t(i)$ is number of confirmed variants among $v(i)$ different choice variants on suggestion i. When the suggestions are causally dependent, received information volume can be calculated as:

$$\left\lceil\log_2\left(\sum\nolimits_{i=1}^{q}\left(\frac{v(i)}{t(i)}-t(i)\right)\right)\right\rceil\tag{10}$$

When the suggestions are dependent and alternative, received information volume can be calculated as:

$$\left\lceil\log_2\left(\sum\nolimits_{i=1}^{q}\left(\frac{v(i)}{v(i)-t(i)}+t(i)-v(i)\right)\right)\right\rceil\tag{11}$$

Accordingly the unified semantic knowledge representation model all knowledge at the knowledge control level are represented as homogeneous semantic networks [42]. Therefore, tutorial material accessed by intelligent agents is represented by semantic network too. This material can contain others types of data such as natural texts, graphic images, audio files. Thus, used semantic network can be classified as a hypermedia semantic network [40].

The other problem solving by intelligent tutoring system is to examine student knowledge and compute its measures (scores) [45]. There are also three types of agents: external search and performative agents and internal knowledge measurement performative agent. The measure function can be expressed by following formulas:

$$m\left(\frac{\max\left(\{0\}\cup\{\rho\left(q\right)*\pi\left(q\right)-q*\sigma\left(q\right)\}\right)}{q*\left(\pi\left(q\right)-\sigma\left(q\right)\right)}\right) \tag{12}$$

where q is number of questions. Each of them has $v\left(i\right)$ risk chances to answer. Also,

$$\begin{aligned}\pi\left(q\right)&=\prod\nolimits_{i=1}^{q}v\left(i\right)\\ \sigma\left(q\right)&=\sum\nolimits_{j=1}^{q}\tfrac{\pi(q)}{v(j)}\\ \rho\left(q\right)&=\sum\nolimits_{j=1}^{q}r\left(j\right)\end{aligned} \tag{13}$$

$$\kappa\left(q\right)=\frac{\sigma\left(q\right)}{\pi\left(q\right)}, \tag{14}$$

$$m\left(x\right)=10*x. \tag{15}$$

To reduce the range of computed values $\pi\left(q\right)$ can be computed as least common multiple.

$$\pi\left(q\right)=\begin{cases}LCM\left(\{v\left(q\right)\}\cup\{\pi\left(q-1\right)\}\right)|q>1\\ v\left(1\right)|q=1\end{cases} \tag{16}$$

If a question i has $t\left(i\right)$ right and $f\left(i\right)$ wrong alternative homogeneous answers then

$$v\left(i\right)=\tfrac{t(i)+f(i)}{t(i)} \tag{17}$$

If right answer is one of the question answers which are short strings in alphabet having a symbols with the length restricted from s to h then

$$v\left(i\right)=\sum\nolimits_{k=s}^{h}a^{k} \tag{18}$$

If unique right answer has length $l(i)$ while question answers are medium strings in alphabet having a symbols with the length restricted from s to h then

$$v\left(i\right)=a^{l(i)}*\left(h-s+1\right) \tag{19}$$

If unique right answer has length $l(i)$ while question answers are strings in alphabet having a symbols [39] with the length restricted from s to h then

$$v\left(i\right)=\left(2*a\right)^{l(i)}*\sum\nolimits_{k=s}^{h}2^{-k}=\left(2*a\right)^{l(i)}*\left(2^{1-s}-2^{-h}\right) \tag{20}$$

If two series of questions (with q_i and q_k questions) have no dependent questions then joint series has q_j questions which satisfy following expressions

$$\begin{aligned}&q_j=q_i+q_k\\ &\pi\left(q_j\right)=\pi\left(q_i\right)*\pi\left(q_k\right)\\ &\sigma\left(q_j\right)=\pi\left(q_i\right)*\sigma\left(q_k\right)+\pi\left(q_k\right)*\sigma\left(q_i\right)\\ &\rho_j\left(q_j\right)=\rho_i\left(q_i\right)+\rho_k\left(q_k\right)\\ &\kappa\left(q_j\right)=\kappa\left(q_i\right)+\kappa\left(q_k\right)\end{aligned} \tag{21}$$

$$\begin{aligned}&m^{-1}\left(e_j\right)*\left(1-\kappa\left(q_j\right)\right)*q_j\leqslant\\ &m^{-1}\left(e_i\right)*\left(1-\kappa\left(q_i\right)\right)*q_i+\\ &m^{-1}\left(e_k\right)*\left(1-\kappa\left(q_k\right)\right)*q_k\end{aligned} \tag{22}$$

$$\begin{aligned}&\left(m^{-1}\left(e_i\right)*\left(1-\kappa\left(q_i\right)\right)-\kappa\left(q_i\right)\right)*q_i+\\ &\left(m^{-1}\left(e_k\right)*\left(1-\kappa\left(q_k\right)\right)-\kappa\left(q_k\right)\right)*q_k\leqslant\\ &\quad m^{-1}\left(e_j*\left(1-\kappa\left(q_j\right)\right)*q_j\right)\end{aligned} \tag{23}$$

If two series of questions (q_i and q_k) have identical questions then joint series has q_j questions which satisfy following expressions

$$\begin{aligned}&q_j=q_i=q_k\\ &\pi\left(q_j\right)=\pi\left(q_i\right)=\pi\left(q_k\right)\\ &\sigma\left(q_j\right)=\sigma\left(q_k\right)=\sigma\left(q_i\right)\\ &\rho_j\left(q_j\right)=\min\left(\{\rho_i\left(q_i\right)\}\cup\{\rho_k\left(q_k\right)\}\right)\\ &\kappa\left(q_j\right)=\kappa\left(q_i\right)=\kappa\left(q_k\right)\end{aligned} \tag{24}$$

Score e_j of the joint series is expressed

$$e_j=\min\left(\{e_i\}\cup\{e_k\}\right) \tag{25}$$

If two series of questions (q_i and q_k) have q_c identical questions then joint series has q_j questions which satisfy following

$$q_c\leqslant\min\left(\{q_i\}\cup\{q_k\}\right) \tag{26}$$

$$\begin{aligned}&2*k_c-q_c\geqslant\\ &\max\left(\{2*q_i*\kappa\left(q_i\right)-q_i\}\cup\{2*q_k*\kappa\left(q_k\right)-q_k\}\right)\end{aligned} \tag{27}$$

$$k_c\leqslant\min\left(\{q_i*\kappa\left(q_i\right)\}\cup\{q_k*\kappa\left(q_k\right)\}\right) \tag{28}$$

$$k_{ic}=q_i*\kappa\left(q_i\right)-k_c \tag{29}$$

$$k_{kc}=q_k*\kappa\left(q_k\right)-k_c \tag{30}$$

Score of the joint series is

$$m\left(\frac{\max\left(\{0\}\cup\{t_c-k_c-k_{ic}-k_{kc}\}\right)}{q_i+q_k-q_c-k_c-k_{ic}-k_{kc}}\right) \tag{31}$$

The task of navigation on tutorial materials is a part of navigation problem on hypermedia semantic networks [40], [42], [43]. Cooperative agents used solve it relies on the search query language. Kinds of agents depend on complexity of queries. The basic navigation interface operates with elementary queries [43].

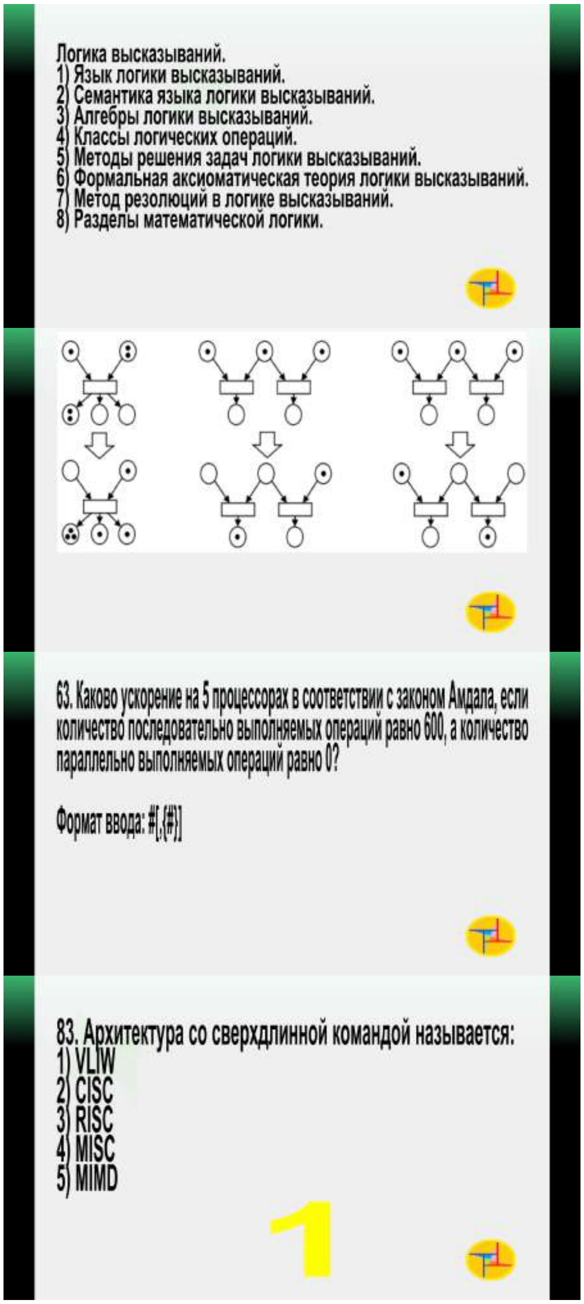


Figure 2. System UI screenshots.

VI. Conclusion

The machines of integration platform were implemented using WebSockets with TCP communication protocol, browser javascript virtual machines and C\C++ compiler for Windows platform. The application was executed in mixed global and local area computer network environment. The server running ontology virtual machines has configuration including AMD Ryzen Threadripper 2950X processor with 8MiB of RAM and Windows 10 operating system. While configuration of clients has Windows 10 Intel Core i3 6100\i5 2310\2500 4\8MiB platform with Chrome, Firefox or Microsoft Edge browsers executing variety virtual machine UI implementation [38]. The described application was used during one semester period in purposes of help information and students knowledge testing system for two disciplines.

The future and perspective plans dealing with the integration platform includes: optimization and development of implementation of models of integration platform architecture to increase its performance and security qualities; implementation of new application of integration platform.

References

[1] William Clinger. Foundations of Actor Semantics. Mathematics Doctoral Dissertation. MIT, June 1981.

[2] Irene Greif. Semantics of Communicating Parallel Processes. EECS Doctoral Dissertation. MIT, August 1975.

[3] Gul Agha, Ian Mason, Scott Smith, Carolyn Talcott. A Foundation for Actor Computation. Journal of Functional Programming, January 1993.

[4] Carl Hewitt. What is Commitment? Physical, Organizational, and Social. April 2006.

[5] Carl Hewitt, Peter Bishop, Richard Steiger. A Universal Modular Actor Formalism for Artificial Intelligence. IJCAI, 1973.

[6] M. Brenner. A Multiagent Planning Language. Proceedings of the Workshop on PDDL. 13th International Conference on Automated Planning and Scheduling (ICAPS-200,3), Trento, Italy, 2003.

[7] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins. PDDL—The Planning Domain Definition Language. Technical Report CVC TR98003/DCS TR1165. New Haven, CT, Yale Center for Computational Vision and Control, 1998.

[8] D. L. Kovacs. A Multi-Agent Extension of PDDL3.1. Proceedings of the 3rd Workshop on the International Planning Competition (IPC). 22nd International Conference on Automated Planning and Scheduling (ICAPS–2012), Atibaia, São Paulo, Brazil, 2012. pp. 19–27.

[9] Manuela Veloso. PDDL by Example. Carnegie Mellon University, 2015.

[10] C.A.R. Hoare. Communicating Sequential Processes. Communications of the ACM, August 1978.

[11] S.D. Brookes, C.A.R. Hoare and W. Roscoe. A theory of communicating sequential processes, JACM, 1984.

[12] Robin Milner, A Calculus of Communicating Systems. Springer Verlag, 1980.

[13] Gul Agha. Actors: A Model of Concurrent Computation in Distributed Systems. Doctoral Dissertation. MIT Press, 1986.

[14] J.C.M. Baeten, T. Basten, M.A. Reniers. Algebra of Communicating Processes. Cambridge University Press, 2005.

[15] Robin Milner, Communicating and Mobile Systems: The Pi Calculus, Cambridge University Press, 1999.

[16] P.H.J. van Eijk et al., The Formal Description Technique LOTOS. North-Holland, 1989.

[17] ISO/IEC international standard 8807:1989. Information Processing Systems – Open Systems Interconnection – LOTOS: A Formal Description Technique based on the Temporal Ordering of Observational Behaviour. Geneva, September 1989.

[18] ISO/IEC international standard 15437:2001. Information technology – Enhancements to LOTOS (E -LOTOS). Geneva, September 2001.

[19] L. Cardelli, A.D. Gordon. Mobile Ambients. Proceedings of the First international Conference on Foundations of Software Science and Computation Structure (March 28 - April 4, 1998). M. Nivat, Ed. Lecture Notes in Computer Science, Springer-Verlag, 1998, vol. 1378, pp. 140–155.

[20] Shahram Rahimi. ACVisualizer: A Visualization Tool for Api-Calculus. October 2015.

[21] Jane Hillston. A Compositional Approach to Performance Modelling. Cambridge University Press, 1996.
[22] Cedric Fournet, Georges Gonthier. The Join Calculus: A Language for Distributed Mobile Programming. 2000.
[23] Robin Milner. Communication and Concurrency. USA, Prentice-Hall, 1989.
[24] Wil van der Aalst. Process Mining: Data Science in Action. 2016.
[25] W. M. P. van der Aalst, A. J. M. M. Weijters, and L. Maruster. Workflow Mining: Discovering process models from event logs, IEEE Transactions on Knowledge and Data Engineering, 2004, vol. 16.
[26] A. K. de Medeiros, W. M. P. van der Aalst, A. J. M. M. Weijters. Workflow Mining: Current Status and Future Directions. Lecture Notes in Computer Science, Springer-Verlag, 2003, vol. 2888.
[27] Davide Sangiorgi. Introduction to Bisimulation and Coinduction. Cambridge University Press, 2011.
[28] Robert M. Keller. Formal Verification of Parallel Programs. Communications of the ACM, July 1976, vol. 19, no 7, pp. 371—384.
[29] James S. Albus. 4D/RCS A Reference Model Architecture for Intelligent Unmanned Ground Vehicles. Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL, 1-5 April 2002.
[30] John E. Laird. The Soar Cognitive Architecture. MIT Press, 2012.
[31] Yuwei Cui, Subutai Ahmad, Jeff Hawkins. "The HTM Spatial Pooler—A Neocortical Algorithm for Online Sparse Distributed Coding". Frontiers in Computational Neuroscience, 2017, Vol. 111, no 11.
[32] R. Orfali, D. Harkey. Java i CORBA v prilozheniyakh klient-server [Client/Server Programming with Java and CORBA] (Transl: Client/Server Programming with Java and CORBA, New York, 1998). Moscow, LORI, 2000, 712 p.
[33] B. Segall, D. Arnold, J. Boot, M. Henderson, T. Phelps. Content Based Routing with Elvin4. Proceedings AUUG2K, Canberra, Australia, June 2000.
[34] Psyclone – Communicative Machines Available at: https://cmlabs.com/psyclone (accessed 2021, May 5)
[35] Open Service Interface Definitions Available at: http://osid.org/ (accessed 2021, May 5)
[36] The Open Agent Architecture™. Available at: http://www.ai.sri.com/∼oaa/oaaslides/ (accessed 2021, May 5)
[37] version / openjsVVM / wiki / Home — Bitbucket (Variety Desert project). Available at: https://bitbucket.org/version/openjsvvm/wiki/Home (accessed 2021, May 5)
[38] Virtual Variety Machine. Available at: http://scnedu.sf.net/ variety/_/index.html?variety=alterface.json&gradient=interscreen .json (accessed 2021, May 5)
[39] R. Solomonoff. Preliminary Report on a General Theory of Inductive Inference. Report V-131 (revision of the Feb. 4, 1960 report). Zator Co., Cambridge, Ma, November 1960.
[40] V.V. Golenkov, N.A. Gulyakina. Primenenie tekhnologii iskusstvennogo intellekta v obuchenii [Application of artificial intelligence technology in education] IV chteniya, posvyashchennye 70-letiyu so dnya rozhdeniya professora V.A. Karpova, 2010, pp. 14–16.
[41] V.V. Golenkov. Otkrytyi proekt, napravlennyi na sozdanie tekhnologii komponentnogo proektirovaniya intellektual'nykh sistem [An open project aimed at creating a technology for the component design of intelligent systems] Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2013, pp. 55—78.
[42] V.P. Ivashenko. Modeli resheniya zadach v intellektual'nykh sistemakh. V 2 ch. Ch. 1 : Formal'nye modeli obrabotki informatsii i parallel'nye modeli resheniya zadach : ucheb.-metod. posobie [Models for solving problems in intelligent systems. In 2 parts, Part 1: Formal models of information processing and parallel models for solving problems: a tutorial] MInsk, BGUIR, 2020, 79 p.
[43] V.P. Ivashenko. Yazyk opisaniya sintaksicheskikh pravil dlya odnorodnykh semanticheskikh setei [Syntax rules description language for homogeneous semantic networks]. Distantsionnoe obuchenie – obrazovatel'naya sreda XXI veka, 2007. pp. 185–188.
[44] V.P. Ivashenko. Predstavlenie neironnykh setei i sistem produktsii v odnorodnykh semanticheskikh setyakh [Representation of neural networks and production systems in homogeneous semantic networks]. Izvestiya belorusskoi inzhenernoi akademii, 2003, vol. 1 no 1. pp. 184–188.
[45] V.P. Ivashenko. Spravochno-proveryayushchaya sistema na osnove unifitsirovannogo semanticheskogo predstavleniya znanii [Reference and testing system based on the unified semantic representation of knowledge] Informatsionnye tekhnologii i sistemy 2020 (ITS 2020) [Information Technologies and Systems 2020 (ITS 2020)], 2020, pp. 80—81.
[46] L.A. Kalinichenko. Metody i sredstva integratsii neodnorodnykh baz dannykh [Methods and tools for integration of heterogeneous databases]. Moscow, Nauka, Fizmatlit, 1983, 424 p.
[47] L.A. Kalinichenko, S.A. Stupnikov, V.N. Zakharov, Extending information integration technologies for problem solving over heterogeneous information resources, Inform. Primen., vol. 1, no 6, 2012, pp. 70—77.
[48] A.S. Narinyani. NE-faktory: netochnost' i nedoopredelennost' – razlichie i vzaimosvyaz' [Non-factors: inaccuracy and underdetermination – difference and interrelation]. Izv RAN (RAS). Ser. Teoriya i sistemy upravleniya 5 (2000). pp. 44—56.

Применение интеграционной платформы для решения задач, основанном на онтологических моделях, использующих унифицированное семантическое представление знаний

Ивашенко В.П.

В статье рассматривается решение в виде интеллектуальной интеграционной платформы, основанной на модели унифицированного семантического представления знаний, для разработки многоагентных систем, управляемых знаниями. В работе применяются: модель унифицированного семантического представления знаний на основе семантических сетей, модели и методы теории меры и теории вероятностей, методы дискретной оптимизации и прикладной математики, компьютерное моделирование и многоагентный подход. Работа направлена на разработку компьютерных средств с когнитивной архитектурой, использующих элементы искусственного сознания, способствующие гибкому взаимодействовию и адаптации этих средств в сложных образовательных приложениях. Были разработаны и реализованы виртуальные машины, другие подсистемы интеграционной платформы, а также - справочно-проверяющая прикладная многоагентная система, функционирующие в рамках системы человеко-машинного взаимодействия.

Received 14.05.2021

Implementation principles of the training subsystem for end-users and developers of intelligent systems

Natalia Grakova
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: grakova@bsuir.by

Maria Koroleva, Georgiy Bobrov
Bauman Moscow State Technical University
Moscow, Russian Federation
Email: maria.svyatkina@gmail.com, bobrov.georgij@yandex.ru

***Abstract*—In the article, an approach to solving the problem of training end-users and developers of intelligent systems is proposed, which involves the supplement of each intelligent system with a module that is an intelligent training subsystem, the purpose of which is to train the end-user and the developer of the main system in the principles of working with it, its operation and development. As a foundation for the implementation of this approach, it is proposed to use an Open Semantic Technology for Intelligent Systems (OSTIS Technology).**

***Keywords*—OSTIS, intelligent system, intelligent learning system, ontological approach**

I. Introduction

Currently, artificial intelligence technologies are being rapidly developed and are being used in various spheres of human activity. Unfortunately, the question is increasingly about systems that contain elements of artificial intelligence, rather than about intelligent systems, to which much higher requirements are imposed. The most important of these requirements is the *learnability* of the system, that is, its ability to acquire new knowledge and skills, and, in particular, *unlimited learnability*, that is, such a degree of learnability, when no constraints are imposed on the typology of this knowledge and skills. In other words, a system with *unlimited learnability* can, if necessary, acquire any knowledge and the ability to solve any problem over time. Let us clarify that this does not mean that one particular system will be able to solve any problem; it means that the system can acquire the ability to solve the required problem, while there are no fundamental restrictions on the class of such problems [1].

At the same time, intelligent computer systems are complex technical systems, the development and even usage of which often require high professional qualities. In particular, the following problems are relevant:

- the lack of efficiency of using modern intelligent systems, the complexity of their implementation and maintenance, which are largely determined by the high threshold of entry of end-users into intelligent systems;
- the user often does not use a significant part of the functions of even traditional computer systems simply for the reason that they do not know about their availability and do not have a simple mechanism to find out about them. For intelligent systems, this problem is even more pressing;
- there are high costs for training developers of intelligent systems, their adaptation to the features of the organization of a particular intelligent system.

These difficulties are connected not only with the inherent complexity of intelligent computer systems compared to traditional computer systems but also with the low level of documentation for such systems, the inconvenience of using such documentation, the complexity of localization of tools and the scope of solving a particular problem both for the end-user and for the developer.

II. Proposed approach

Within the framework of this article, an approach to solving these problems is proposed, involving the supplement of each intelligent system with a module that is an intelligent training subsystem, the purpose of which is to train the end-user and the developer of the main system in the principles of working with it, the principles of its operation and development.

In other words, the main idea of the proposed approach can be exemplified as follows: regardless of what problems an intelligent system is being developed for, it must have some of the functions of a training system, even if the system is not initially a training one. So, the end-user should be able to study both the principles of working with an intelligent system and gain new knowledge about the subject domain, for which an intelligent system is being created. In turn, the developer of intelligent systems should be able to study the principles of the internal organization of the system, the principles of its operation, the functions of specific components of the system, have the opportunity to localize the part of the system, on which they must gain insight for making changes to the functionality of the system.

To implement this idea, an intelligent system must contain not only knowledge about the subject domain, for which it is designed, but also

- knowledge about itself, its architecture, components, functions, operating principles, etc.;
- knowledge about the user, their experience, skills, preferences, interests;
- knowledge about the problems that the system solves independently at the moment and the problems that are planned to be solved in the future;
- knowledge about current problems in the development of the system and its maintenance.

The representation and processing of all of the above require a common formal foundation for the representation of knowledge of various types as well as a common foundation for various types of means of processing this knowledge.

As such a foundation, it is proposed to use an Open semantic technology for intelligent systems (*OSTIS Technology*) [2], which allows integrating any type of knowledge and any problem-solving model. The systems being developed on the basis of this technology are called *ostis-systems*.

The usage of the OSTIS Technology gives the following advantages for solving the specified problem:

- The technology is based on the *SC-code* – a universal and unified language for encoding information in the dynamic graph memory of computer systems. The SC-code allows representing <u>any</u> information in a <u>unified</u> (similar) form, which will make the proposed approach universal and suitable for any class of intelligent systems;
- The OSTIS Technology and, in particular, the SC-code, can be easily integrated with any modern technology, which will allow applying the proposed approach to a large number of already developed intelligent systems;
- The SC-code allows storing and describing in the ostis-system knowledge base any external (heterogeneous) information in relation to the SC-code in the form of internal files of ostis-systems [3]. Thus, the knowledge base of the training subsystem can explicitly contain fragments of existing documentation for the system, represented in any form;
- Within the framework of the OSTIS Technology, models of the ostis-system knowledge bases [4], ostis-system problem solvers [5] and ostis-system user interfaces [6] have already been developed, asserting their full description in the system knowledge base. Thus, for ostis-systems, the proposed approach is implemented much easier and provides additional advantages, which are discussed in more detail in this article [7];
- One of the main principles of the OSTIS Technology is to ensure the flexibility (modifiability) of systems developed on its basis. Thus, the usage of the OSTIS Technology will provide an opportunity for the evolution of the intelligent learning subsystem itself.

Further, let us consider in more detail the possibilities of using the OSTIS Technology for the development of intelligent learning systems.

III. Intelligent learning systems

Intelligent learning systems (ILS) are an important class of intelligent systems. Such systems, in comparison with traditional e-learning systems (for example, electronic textbooks), have a number of significant advantages [8]. At the same time, the issue of intellectualization of the learning process remains relevant [9]–[11], including with the usage of semantic technologies [12]. At the same time, as can be seen from the above papers, the relevance of this issue is realized by specialists in the field of education and not only by specialists in the field of intelligent system development.

In the case of the implementation of ILS based on the OSTIS Technology, additional features appear, which include the following:

- The semantic structure of the studied educational material and the studied subject domain is explicitly presented to the user. At the same time, visualization of any level of the specified semantic structure is enabled;
- The user gains access to sufficiently complete information about the studied subject domain; all its aspects are reflected thanks to the explicit placement of all subject conformities and interrelations of concepts in the knowledge base;
- In addition to the ability to read texts and illustrative materials of the textbook, it is possible to navigate through the semantic space of the subject domain;
- The user is allowed to ask the system any questions and set problems on the studied subject domain. This is achieved by including a problem solver in the ILS, which can solve problems according to their formulations, including those input by the user. At the same time, the specified problem solver can find a way to solve the problem even if the corresponding solution method (for example, an algorithm) is unknown to it;
- The user is given the opportunity to train (acquire practical skills) in solving a variety of problems in the studied subject domain under the control of the system. At the same time, the system
  – performs a semantic analysis of the correctness of solving problems both according to freely generated answers (results) and the solution protocols;

- localizes the errors made by the user in solving problems, determines their cause and gives appropriate recommendations to the user.

- The ILS has an intelligent user interface with computer (virtual) models of various objects of the studied subject domain, which allows the system to "understand" the meaning (analyze the semantics) of user actions on the transformation of these objects. All this significantly increases the level of the interactive virtual laboratory environment of the electronic textbook;
- When communicating with the system, the user is given free hand in choosing any of the many synonymous terms (identifiers) registered in the system knowledge base. At the same time, these terms may belong to different natural languages;
- There is a principal possibility of implementing a natural-language interface with the user (thanks to the wide possibilities of semantic analysis of user messages and the possibilities of synthesis of messages addressed to users at the semantic level);
- It is quite easy to reorient the ILS to serve users with a different natural language (since the major part of the ILS knowledge base, which directly describes the semantics of the corresponding subject domain, is independent of the external language including the natural one);
- The user can choose the sequence of studying the educational material (the route-based navigation through the educational material), but relevant recommendations are given;
- The user can choose the problems being solved (in the book of problems and laboratory-based works), but the appropriate recommendations are given. These recommendations are aimed at minimizing the number of solved problems that ensure the acquisition of the required practical skills;
- The system does not have a special mode for control (verification, testing) of knowledge. Such control is carried out seamlessly for the user by monitoring and analyzing user actions when solving various problems in the studied subject domain. To do this, the ILS knowledge base contains information about what types of problems and laboratory-based works should be completed by the user for satisfactory, good and excellent assimilation of the educational material, respectively;
- It is quite easy to integrate several independent ILS in related disciplines into a single textbook, which, in particular, provides an opportunity to ask questions and set problems at the intersection of these disciplines;
- The ILS user works under the supervision and control of an intelligent helper, which helps the user master the capabilities of the system quickly and effectively. In fact, this is nothing more than an ILS user's guide designed as a semantic electronic textbook;
- When designing the ILS knowledge base, there is a unique opportunity to check the semantic correctness of the information resource being formed:
  - correctness of definitions and statements;
  - correctness of the usage of various concepts;
  - correctness of algorithms;
  - correctness of proofs of theorems;
  - etc.

Some of these opportunities (in the limiting case, all of them) can be implemented within the framework of the subsystem for training users of an intelligent system. Next, let us consider in more detail the architecture of the proposed subsystem.

IV. Architecture of the subsystem for training users of intelligent systems

Figure 1 shows the architecture of the proposed *subsystem for training users of intelligent systems*. To implement the interaction of the *subsystem for training users of intelligent systems*, implemented on the basis of the OSTIS Technology, with the main intelligent system, it is planned to develop an interface component, which is also part of the subsystem. It is important to note that for different intelligent systems, such components will be largely overlapping, which is due to the features of the OSTIS Technology itself, which, in turn, will reduce the cost of integrating the training subsystem and the main intelligent system.

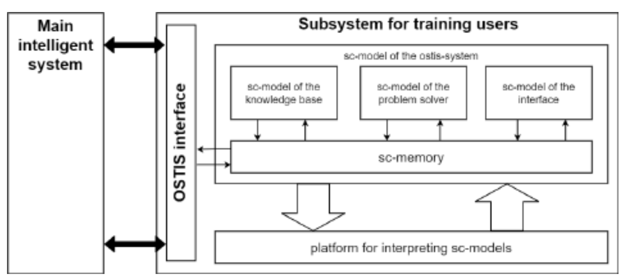


Figure 1. The architecture of the subsystem for training users of intelligent systems

If the intelligent system under consideration is an ostis-system, its integration with the *subsystem for training users of intelligent systems* is carried out more deeply, and the architecture of the resulting integrated system can be represented as follows (fig. 2). As can be seen from the figure, the components of the *subsystem for training users of intelligent systems* solely complement the already existing in the main ostis-system components, which allows minimizing the cost of integrating the *subsystem for training users of intelligent systems* and the main ostis-system.

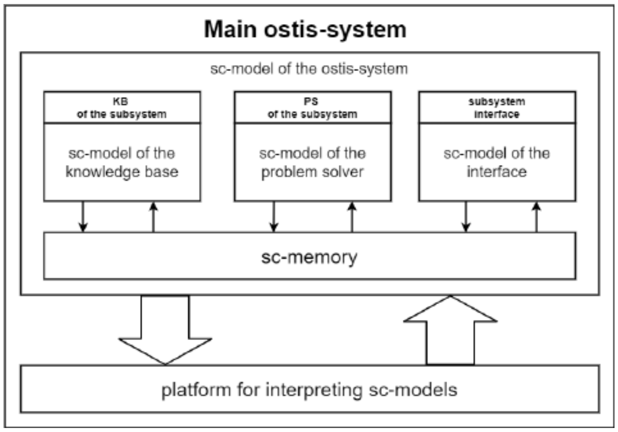


Figure 2. The architecture of the subsystem for training users of intelligent systems as part of another ostis-system

V. Examples of the representation of various types of knowledge within the framework of the subsystem for training users of intelligent systems

Let us consider a number of examples that illustrate the possibilities of the approach proposed within the framework of the OSTIS Technology to the development of knowledge bases when describing various types of knowledge in the subsystem for training users of intelligent systems. For the illustrations, the SCg- and SCn-codes [3] will be used, which are the languages for the external representation of SC-code constructions.

Figure 3 in the SCg-code shows various information about a particular user of the intelligent system. This example shows how using the knowledge base structuring tools developed within the framework of the OSTIS Technology [4], it is possible to describe various types of information about the same entity in the knowledge base, in particular, the current employment and professional skills of the user. Any other information about the user can be described in the same way.

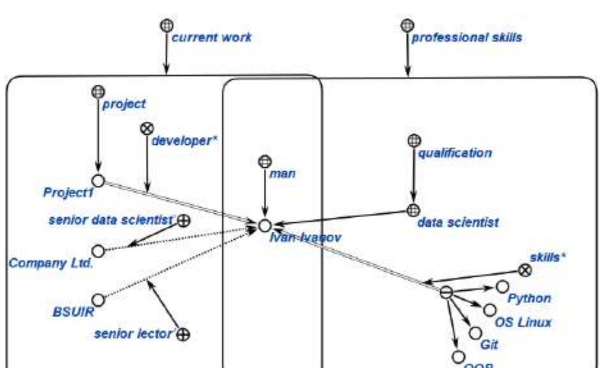


Figure 3. The description of the user of the intelligent system

The ostis-system knowledge base can be structured according to various features [4]. Within the framework of this article, the structuring of the knowledge base from the point of view of its development process is of the greatest interest. Let us consider a variant of such structuring in the SCn-code:

semantic model of the knowledge base
⇒ *abstract basic decomposition**:
{• *history and current processes of computer system operation*
⇒ *abstract basic decomposition**:
{• *history of computer system operation*
• *current processes of computer system operation*
}
• *computer system documentation*
• *context of the subject part of the knowledge base within the Global knowledge base*
• *subject part of the knowledge base*
• *history, current processes and development plan of the computer system*
⇒ *abstract basic decomposition**:
{• *current processes of computer system development*
• *history of the computer system development*
• *structure and organization of the computer system project*
• *computer system development plan*
}
}

Figure 4 in the SCg-code shows an example of describing information about the performers of a certain project who take on various roles in it. From the point of view of the knowledge base structure, this information is part of the *structure and organization of the computer system project* section.

Figure 5 in the SCg-code shows an example of a description of project tasks and their performers, taking into account the qualifications of each performer. From the point of view of the knowledge base structure, this information is part of the *current processes of computer system development* section.

According to the approach proposed within the framework of the OSTIS Technology to the development of problem solvers, the basis of the solver is a hierarchical system of agents over semantic memory (sc-agents) [5]. The structure of the solver can also be described in the ostis-system knowledge base. Next, in the SCn-code, the structure of the problem solver of the labeling quality control system for the formulating enterprise is presented:

Problem solver of the labeling quality control system
⇒ *decomposition of an abstract sc-agent**:

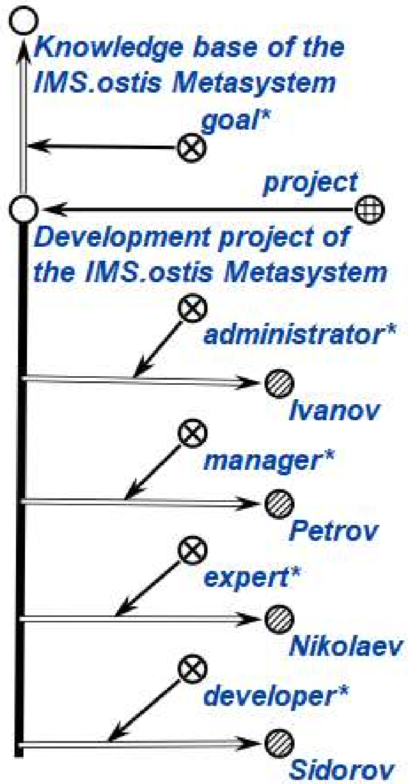


Figure 4. The description of the performers of the project for the development of some intelligent system

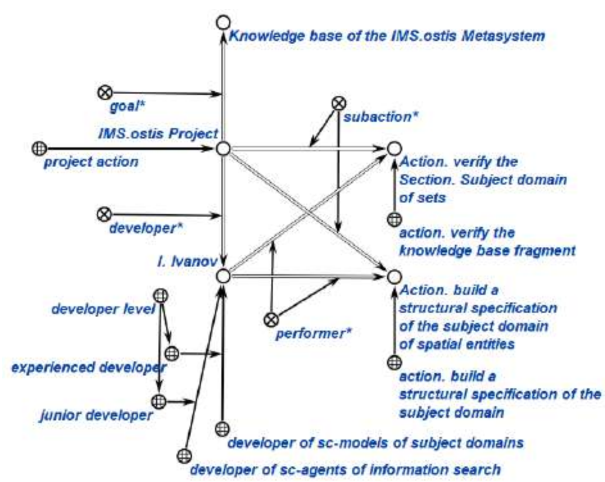


Figure 5. The description of the project tasks

{• *Atomic abstract sc-agent of labeling recognition based on a neural network*
• *Non-atomic abstract sc-agent of decision-making*
⇒ *decomposition of an abstract sc-agent**:
{• *Atomic abstract sc-agent that implements the concept of a software package*
• *Non-atomic abstract sc-agent of certain inference*
• *Non-atomic abstract sc-agent of reliable inference*
}
• *Non-atomic abstract sc-agent of content-addressable retrieval*
• *Non-atomic abstract sc-agent of interpretation of control programs for a robotic installation*
⇒ *decomposition of an abstract sc-agent**:
{• *Atomic abstract sc-agent of interpretation of the movement action*
• *Atomic abstract sc-agent of interpretation of the acquisition action*
}
}

Non-atomic abstract sc-agent of certain inference
⇒ *decomposition of an abstract sc-agent**:
{• *Atomic abstract sc-agent that implements a certain inference strategy*
• *Non-atomic abstract sc-agent of logical rules interpretation*
}

Non-atomic abstract sc-agent of reliable inference
⇒ *decomposition of an abstract sc-agent**:
{• *Atomic abstract sc-agent that implements a reliable inference strategy*
• *Non-atomic abstract sc-agent of logical rules interpretation*
}

Non-atomic abstract sc-agent of logical rules interpretation
⇒ *decomposition of an abstract sc-agent**:
{• *Atomic abstract sc-agent of applying implicative rules*
• *Atomic abstract sc-agent of applying equivalence rules*
}

VI. Conclusion

As part of the further development of the proposed idea, models and tools for improving the efficiency of training users of intelligent systems will be developed in the form of an appropriate set of ontologies that describe knowledge about the intelligent system itself, its users and tasks for its development, models of the subsystem for training users of intelligent systems, a corresponding set of software agents, which implements the functionality of the subsystem considered in the article.

The implementation of the proposed idea will allow:

- reducing the threshold of entry of end-users into intelligent systems as well as significantly improving the efficiency of using such systems;
- reducing the time for training developers of intelligent systems, their adaptation to the features of a particular intelligent system, development and refinement of the intelligent system. All of the above will also solve the problem of employee turnover at enterprises engaged in the development of complex intelligent systems as well as reduce the cost of developing and maintaining intelligent systems and, as a result, make them more available to the end-user.

ACKNOWLEDGMENT

The authors would like to thank the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics for the help and valuable comments.

The work was carried out with the partial financial support of the BRFFR and RFFR, project numbers F21PM-139 and 20-57-04002.

REFERENCES

[1] V. Golenkov, N. Guliakina, N. Grakova, I. Davydenko, V. Nikulenka, A. Eremeev, and V. Tarassov, "From training intelligent systems to training their development tools," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 81–98, 2018.

[2] V. Golenkov, N. Gulyakina, I. Davydenko, and D. Shunkevich, "Semanticheskie tekhnologii proektirovaniya intellektual'nyh sistem i semanticheskie associativnye komp'yutery [Semantic technologies of intelligent systems design and semantic associative computers]," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 42–50, 2019.

[3] (2021, Jun) IMS.ostis Metasystem. [Online]. Available: https://ims.ostis.net

[4] I. Davydenko, "Semantic models, method and tools of knowledge bases coordinated development based on reusable components," in *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2018, pp. 99–118.

[5] D. Shunkevich, "Agentno-orientirovannye reshateli zadach intellektual'nyh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 119–132.

[6] A. Boriskin, D. Koronchik, I. Zhukau, M. Sadouski, and A. Khusainov, "Ontology-based design of intelligent systems user interface," in *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2017, pp. 95–106.

[7] V. Golenkov, N. Guliakina, I. Davydenko, and A. Eremeev, "Methods and tools for ensuring compatibility of computer systems," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2019, pp. 25–52.

[8] V. Golenkov, V. Tarasov, and O. Eliseeva, *Intellektual'nye obuchayushchie sistemy i virtual'nye uchebnye organizatsii [Intelligent Learning Systems and Virtual Learning Organizations]*, V. Golenkov and V. Tarasov, Eds. Minsk. BSUIR, 2001.

[9] G. Y. Yalamov and T. S. Shikhnabieva, "Adaptivnye obrazovatel'nye informatsionnye sistemy: podkhody k intellektualizatsii [adaptive educational information systems: approaches to intellectualization]," in *Chelovek i obrazovanie [Human and education]*, no. 4 (57), 2018, pp. 84–90, (in Russ).

[10] Y. A. Vagramenko and G. Y. Yalamov, "Analiz napravlenii intellektualizatsii sovremennykh informatsionnykh sistem uchebnogo naznacheniya[analysis of the directions of intellectualization of modern information systems of educational appointment]," in *Upravlenie obrazovaniem: teoriya i praktika [Education management: theory and practice]*, no. 4(24), 2016, pp. 44–57, (in Russ).

[11] V. M. Trembatch, "Elektronnye obuchayushchie sistemy s ispol'zovaniem intellektual'nykh tekhnologii[e-learning systems using intellectual techniques]," in *Otkrytoe obrazovanie [Open education]*, no. 4, 2013, pp. 52–62, (in Russ).

[12] T. S. Shikhnabieva, "Adaptivnye semanticheskie modeli avtomatizirovannogo kontrolya znanii [adaptive semantic model of automatic knowledge control]," in *Pedagogicheskoe obrazovanie v Rossii [Pedagogical education in Russia]*, no. 7, 2016, pp. 14–20, (in Russ).

[13] V. Golenkov, V. Tarasov, and V. V. Emelyanov, "Virtual'nye kafedry i intellektual'nye obuchayushchie sistemy [virtual department and intelligent tutoring systems]," *Novosti iskusstvennogo intellekta [News of Artificial Intelligence]*, no. 4, pp. 3–13, 2001, (in Russ).

[14] V. Tarasov and V. Golenkov, "Sinergeticheskie uchebnye organizatsii v sfere vysshego obrazovaniya [synergistic organizations in the field of higher education]," in *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2017, pp. 221–228, (in Russ).

Принципы реализации подсистемы обучения конечных пользователей и разработчиков интеллектуальных систем

Гракова Н.В., Королева М.Н., Бобров Г.С.

В работе предложен подход к решению проблемы обучения конечных пользователей и разработчиков интеллектуальных систем, предполагающий дополнение каждой интеллектуальной системы модулем, представляющим собой интеллектуальную обучающую подсистему, целью которой является обучение конечного пользователя и разработчика основной системы принципам работы с ней, принципам ее функционирования и развития. В качестве фундамента для реализации данного подхода предлагается использовать открытую семантическую технологию проектирования интеллектуальных систем (Технологию OSTIS).

Received 03.06.2021

Semantic analysis of the video stream based on neuro-symbolic artificial intelligence

Kroshchanka Aliaksandr, Mikhno Egor
Brest State Technical University
Brest, Belarus
kroschenko@gmail.com,
dzinushi.kun@gmail.com

Kovalev Mikhail, Zahariev Vadim,
Zagorskij Alexandr
Belarusian State University
of Informatics and Radioelectronics
Minsk, Belarus
michail.kovalev7@gmail.com, zahariev@bsuir.by,
alexandr.zagorskiy.work@gmail.com

***Abstract*—In the article, the model developed by the authors is considered, which is used for the semantic analysis of the video stream. The model is based on a neuro-symbolic approach. The features and advantages of the model are described. Based on the proposed model, a hybrid system for semantic analysis of the emotional state of the user is implemented. The configuration of the hardware platform necessary for the operation of the developed system is given.**

***Keywords*—neuro-symbolic AI, computer vision, artificial neural network, knowledge base, inference**

I. Introduction

In the last decade, there has been a steady tendency to widely use methods of machine learning and computer vision in various areas of human activities, primarily due to the development of the theory of artificial neural networks and hardware capabilities.

The development of applied methods in the field of computer vision leads to new original practical solutions.

The number of processes, which are being automated using new approaches in computer vision and which often could not be automated with acceptable quality earlier, is growing rapidly. In industries, it has become possible to reduce human participation in the process of product development and quality control [1]; in medicine, computer vision is used to analyze medical images; in the transport industry, it helps to carry out visual control of maintaining traffic regulations and operate autonomous vehicles.

Developments that are able not only to support basic functionality (even intellectual one) but also to conduct complex semantic analysis that produces new knowledge that can be used to improve the quality of the system as a whole are of incontestable value.

This combination involves the joint usage of ideas and methods from the fields of artificial neural and semantic models (and in the limit – connectionist and symbolic approaches in AI).

The advantage of artificial neural networks (ANN) is that they can work with unstructured data. The main disadvantage of ANN is the lack of human-understandable feedback, which could be called a reasoning chain, i.e., it can be said that ANN work as a "black box" [2].

Symbolic AI is based on symbolic (logical) reasoning. Such AI allows solving problems that can be formalized and plays an important role in human knowledge [3], [4]. However, it is not designed to work with unstructured data. Thus, a proper combination of these approaches will allow transforming unstructured data into knowledge.

The building of hybrid neuro-symbolic systems has already been widely covered in the literature [5]–[8], but currently all these attempts are mostly theoretical.

In this article, a neuro-symbolic model is proposed, which integrates various ANN models, engaged in solving problems of detection and recognition, with a knowledge base built on the basis of an ontological approach [9]. Based on the proposed model, the authors developed a real system for semantic analysis of the emotional state of the user.

The article is structured as follows:

- in section II, the problem definition to develop a model of a computer vision system for the semantic analysis of a video stream based on a neuro-symbolic approach is presented;
- section III is dedicated to the architecture of the developed model and the integration of its main components;
- section IV is concerned with the description of aspects of the implementation of a hybrid system for the semantic analysis of the emotional state of the user based on the proposed model;
- finally, in section V, the description of the hardware platform, on which the system described in section IV is launched, is given.

II. Problem definition

The authors were set the problem of developing:

1) a model of a computer vision system designed for semantic analysis of a video stream;

2) a hybrid system for semantic analysis of the emotional state based on this model.

When defining the problem, it was assumed that the nature of the data analyzed by the model can change, so the model should have a modular structure that allows for the simple replacement of modules in case of changes in the analyzed data.

The model should analyze the most semantically "rich" high-level data that can be obtained as a result of the detection of any objects and their recognition.

The following requirements are imposed on the model:

- support for various embedded modules for detecting and recognizing objects on video;
- simplicity in adding new models;
- semantic analysis of the results of recognition;
- availability of using the results of semantic analysis;
- the possibility of explaining the results of this analysis to a human.

To meet the specified requirements, a model of neuro-symbolic AI is proposed, within which the interaction of ANN and KB is organized to solve the problems of image recognition and semantic analysis, respectively.

The following requirements are imposed on the project of a hybrid system for semantic analysis of the emotional state of the user:

- The system should be able to identify the face of a person from the list of persons known to it;
- The system should be able to determine the fact of the appearance of an unknown person and add it to the list of persons known to it;
- The system should evaluate the emotional state of the person in the frame and respond accordingly;
- The system should accumulate statistical information about persons, who appeared in front of the camera, and their emotional state.

All the specified problems were solved within the developed model and the system based on it.

III. Overall structure of the proposed model

The main components of the proposed system model are:

- an interface, which in the simplest case can be represented by a camera, the video stream from which is transmitted to the computer vision module;
- a computer vision module, in which the input video stream is split into separate frames, which are recognized by available neural network models, and the obtained recognition results are transmitted to the knowledge integrator;
- a knowledge integrator that forms the necessary constructions from the results of recognition and places the generated knowledge in the knowledge base;
- a knowledge base that stores knowledge about the subject domains of the problems of recognition solved by the system, indications of neural network models that solve these problems as well as the results of recognition;
- a problem solver that performs semantic analysis of the results of recognition in the knowledge base.

Figure 1 shows the scheme of interaction of the above components. This scheme displays the upper layer of abstraction of the system components, which in practice can be implemented more comprehensively. The interaction between the above components can be implemented in various ways, for example, based on a network protocol, direct access within a monolithic architecture, shared access to some data storage, etc.

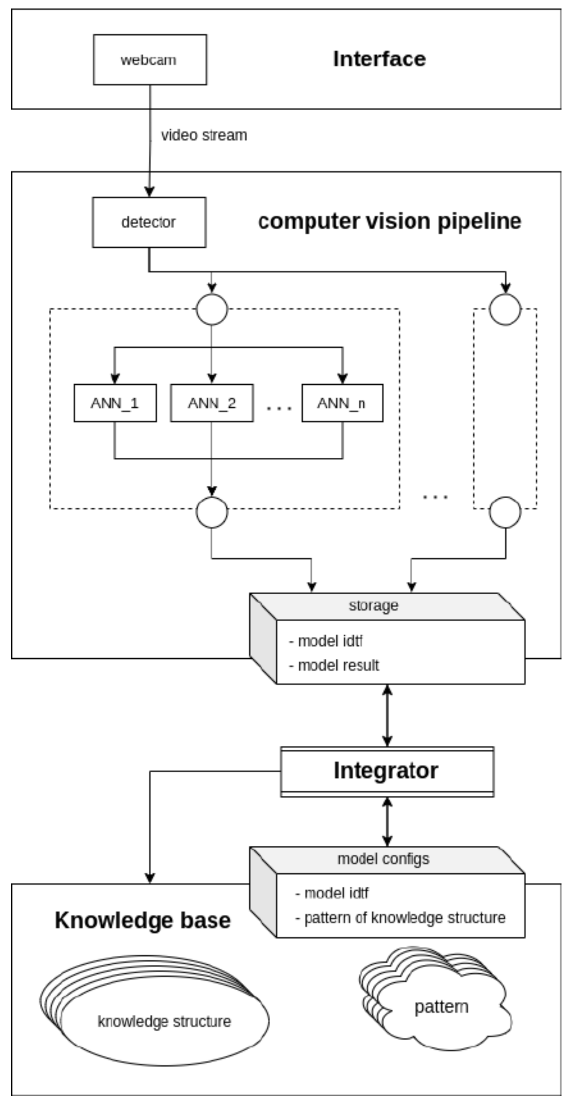


Figure 1. The scheme of interaction of the main components of the system

Since the developed model is a hybrid, the mechanisms of integration of the computer vision module and the knowledge base are of the greatest interest. Next, let us consider these mechanisms in more detail.

A. Integration of the computer vision module with the knowledge base

The integration of the computer vision module with the knowledge base as well as the direct implementation of the knowledge base and its processing are based on the usage of the OSTIS Technology [10].The model of the neuro-symbolic AI system considered in this article is an ostis-system.

Among the advantages of ostis-systems, it is possible to distinguish:

- the ability to perform semantic integration of knowledge in own memory;
- the ability to integrate different types of knowledge;
- the ability to integrate various problem-solving models.

The internal language used by ostis-systems is called an SC-code. Any knowledge can be represented in the form of an SC-code construction. A set of knowledge forms the ostis-system knowledge base. It is quite convenient to operate with knowledge in such a form: the technology supports the search and generation of necessary constructions both according to special templates and elementwise. You can read more about knowledge representation using the SC-code here [11].

1) Knowledge integrator: To carry out a semantic analysis of the result of the functioning of the neural network model, it is necessary to place these results in the knowledge base, i.e., to transform information into knowledge. The problems of converting information into knowledge are:

- the design of knowledge structures (defining the key nodes of the selected subject domain, their coordination with the formed knowledge base);
- automation of the formation and placement of structures in the KB.

Within the framework of the computer vision module, these problems are solved by the knowledge integrator, which places the results of the functioning of neural network models in the KB.

Neural network models solve the following problems:

- detection of objects;
- identification of detected objects of a certain class;
- recognition of the class of all detected objects.

As a result of solving these problems, various knowledge is placed in the knowledge base:

- knowledge about the presence/absence of objects in the “field of vision”;
- knowledge about the correspondence of the detected objects of a certain class to some entities available in the knowledge base (if there are no such entities, new ones are created);
- knowledge about the classes of detected objects and the periods, during which the objects corresponded to these classes.

Due to this unification of the problems that neural network models solve within the computer vision module, it is possible to achieve the independence of the operation of the integrator from particular neural network models. It would be adequate to formalize the specification of such models in the knowledge base, which includes:

- the type of problem being solved;
- input and output data (depending on the type of the problem);
- the work frequency (if necessary);
- the state (on/off);
- the identifier, by which the specification of the model in the knowledge base can be correlated with the model in the computer vision module.

The knowledge integrator receives the specifications of all neural network models presented in the computer vision module and performs the integration of knowledge in accordance with these specifications.

Figure 2 shows an example of formalization of the specification of a neural network model for recognizing emotions in the knowledge base.

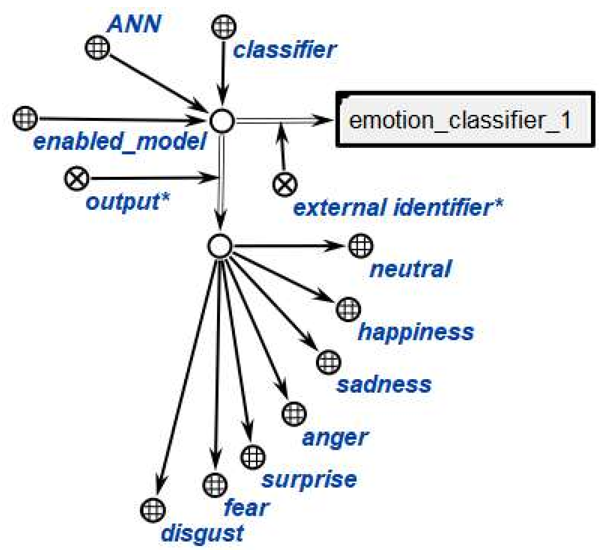


Figure 2. An example of formalization of the specification of a neural network model for recognizing emotions in the knowledge base

This neural network model solves the problem of classifying emotions and outputs for each recognized class a degree of confidence that the recognized object belongs to this class. Further, in accordance with the period stated in the model specification, statistics of the operation of such a neural network model are accumulated.

Depending on the configuration of the neural network model, which indicates the need to place operation statistics, confidence percentage for the answers, active time, etc. into the knowledge base, the integrator chooses a template, according to which it will generate knowledge. Such templates are also presented in the knowledge base.

Figure 3 shows an example of a template for generating the result of recognizing an object class over a certain period.

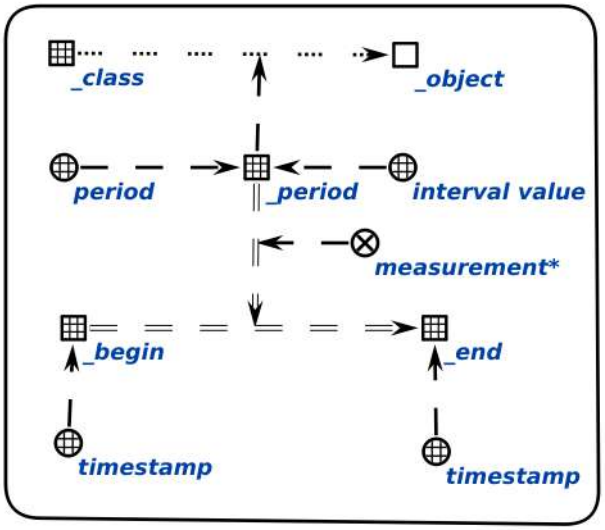


Figure 3. A template for generating the result of recognizing an object class over a certain period

An example of the knowledge structure that is formed based on the results obtained by the neural network model of emotion recognition of the computer vision module is shown in figure 4.

The approach, when the integrator works with the specifications of neural network models in the knowledge base, allows:

- avoiding overhead costs when integrating new neural network models, since it is enough only to describe the specification of the new model in the knowledge base;
- managing the computer vision module from the KB (for example, it is sufficient to add a neural network model to a set of enabled models so that the computer vision module stops running this neural network model for a video stream).

IV. Implementation of a hybrid system for analyzing the emotional state

Let us consider in more detail the architecture of the computer vision module and the principle of operation of the semantic analyzer.

A. Structure of the computer vision module

Adhering to the described system requirements, we implemented the computer vision module. Its structure is shown in fig. 5. The scheme is greatly simplified from the point of view of interaction with the knowledge base and reflects only the sequence and connectivity of computer vision modules. Let us consider these modules in more detail.

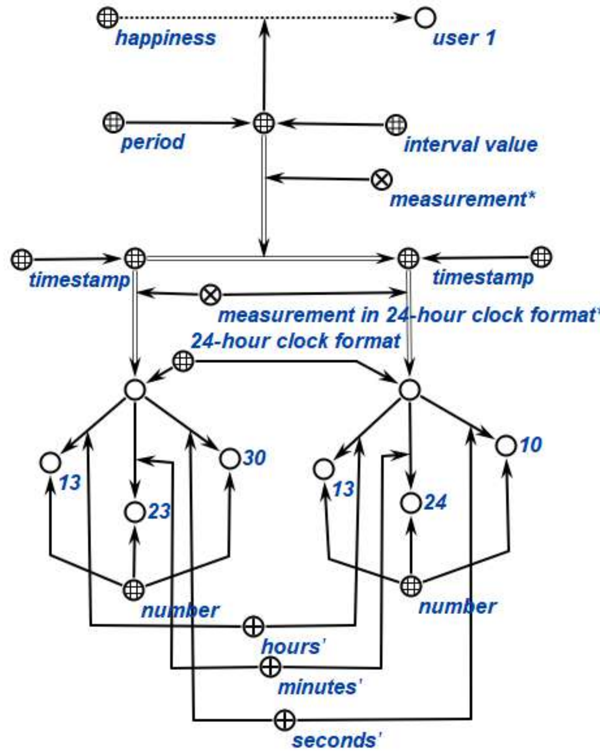


Figure 4. An example of a representation of the user and their emotions for a certain period in the knowledge base

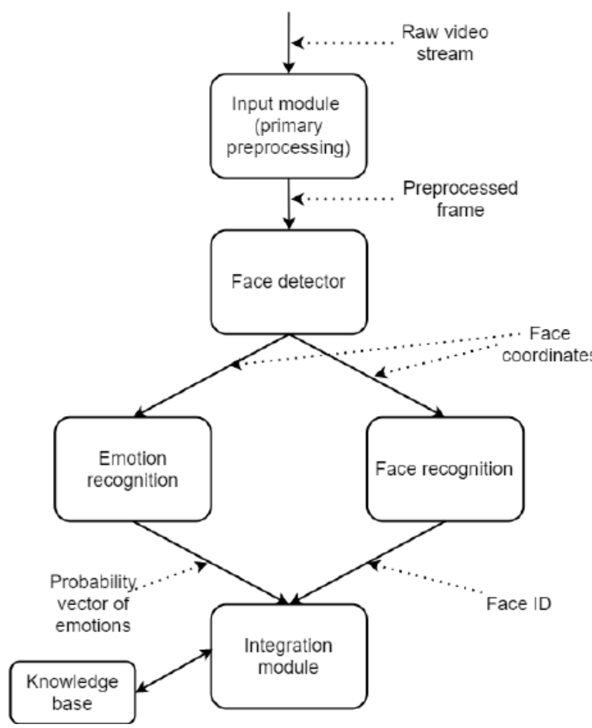


Figure 5. The scheme of the computer vision module pipeline

The **face detection module** solves the problem of detecting faces in the frame. It receives a frame from the video stream as input and returns the coordinates of the found faces.

The **identification module** is required to recognize the face of the user being identified. Receiving the coordinates of the detected persons as input, this module returns the user IDs.

The **emotion recognition module** is independent of the identification logic. Like the identification module, it receives the coordinates of the detected faces as input and returns a probability vector of emotions.

After the performance of particular branches of the pipeline, the results of the work of the modules are transmitted to the input of an integrator, which performs an immersion in the knowledge base.

Next, we will consider in more detail the main functions of the computer vision module and the results obtained in the preparation of appropriate neural network models.

B. Functions of the computer vision module

The functions of the computer vision module include:

- identification of a user known to the system;
- identification of unknown users after additional training;
- recognition of user emotions.

C. User identification

User identification is carried out by the identification module. The processes that occur in the module are most fully represented in the flowchart in fig. 6.

To implement this function, the FaceNet [12] neural network model is used. For this model, classical models of deep convolutional neural networks with a triplet loss function are used. In our case, the ResNet [13] convolutional network was used. The overall scheme of the model is shown in fig. 7.

This model consists of successive layers of a deep convolutional neural network and L_2 normalization using a triplet loss function at the training stage. At the output of the model, a 128-dimensional feature vector is formed, which can be used for the native comparison of faces.

The models from the dlib [14] library were used as the basic implementation.

Let us consider the user identification algorithm.

1) The user's face is detected by the detector in the frame (the MTCNN [15] model is used as a detector);
2) For the detected face, the feature vector using ANN is calculated;
3) After calculating the feature vector, it is compared with other feature vectors stored in the database. This comparison is performed with the preset threshold;

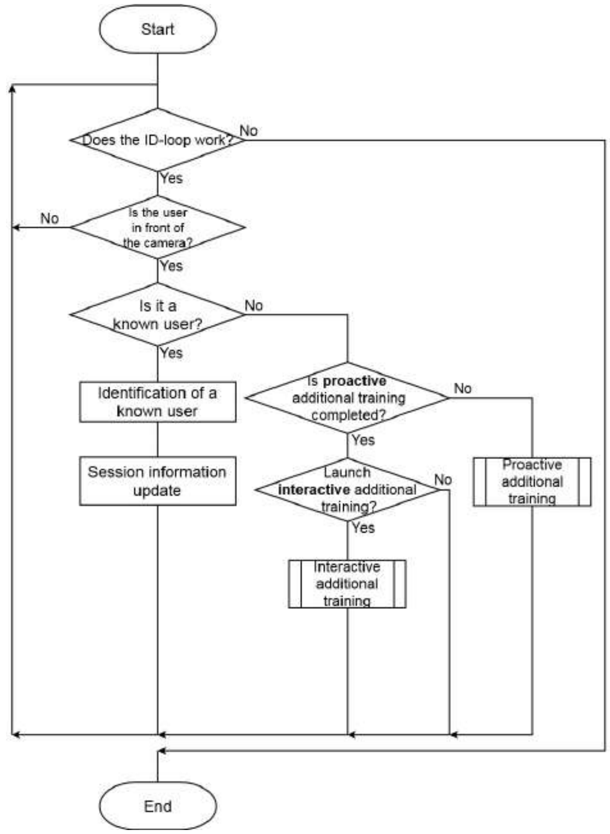


Figure 6. The overall flowchart of identification

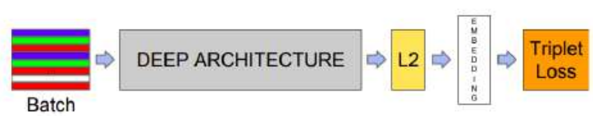


Figure 7. The structure of FaceNet (original image from [12])

4) Based on the results obtained in item 3, an ID is assigned to the detected face.

Thus, a necessary condition for the functioning of the algorithm is the presence of pre-calculated feature vectors for known users. This approach allows identifying the user with acceptable speed and accuracy.

The advantage of the proposed approach is that the implementation of recognition does not require a large training dataset of data, since the used FaceNet model is pre-trained on a large data array (a dataset of more than 3 million images) and can be used unchanged to identify people who were not included in this dataset.

For training, a dataset of photos of 7 different people – 7 photos of each person – was used. Thus, the dataset size was 49 photos. For testing, an independent test dataset of 14 photos and a set of video fragments were used to assess the quality of user recognition.

As a result of the evaluation of the proposed algorithm, the efficiency of face recognition was 95.84%. At the same time, the percentage of correctly recognized faces in the test dataset (due to its small size) was 100%.

D. Additional training for new unknown users

In addition to identifying known users by the feature vectors that are present in the database, the system allows performing additional training for recognizing unknown users in real-time. *Additional training* here means calculating a set of feature vectors for new users and saving them for further usage in the identification process. This process is carried out within so-called proactive and interactive additional training.

Proactive additional training (fig. 8) is conducted without direct user participation while calculating feature vectors based on frames received from the video stream.

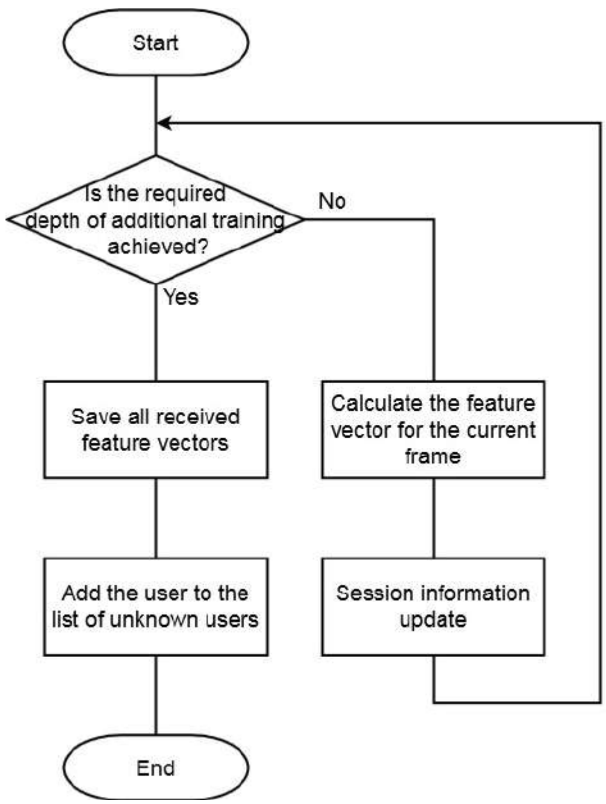


Figure 8. The scheme of proactive additional training

Interactive additional training for new users (fig. 9) conducted separately and with the direct participation of the user allows improving the results of proactive additional training and getting more representative feature vectors.

E. Recognition of user emotions

The user emotions are recognized in seven basic classes: neutral facial expression, happiness, sadness, anger, fear, surprise and disgust.

The problem of classifying emotions by the image of the user's face is studied in many papers (for example,

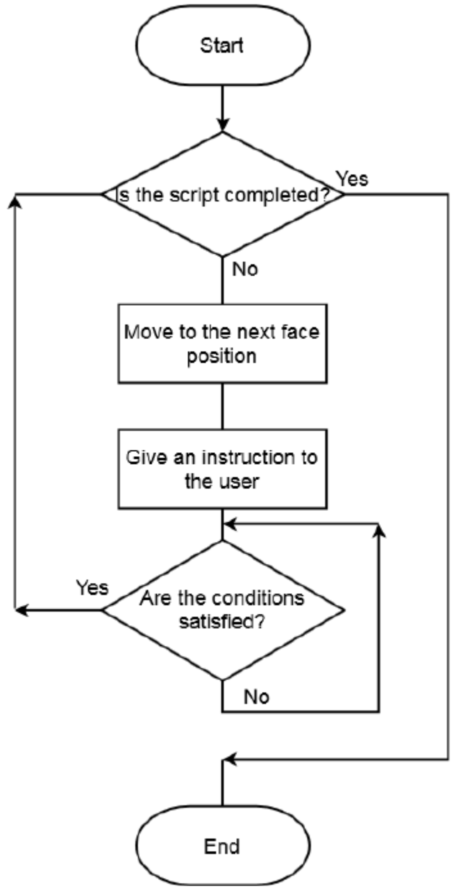


Figure 9. The scheme of interactive additional training

[16], [17], [18], etc.). With the active development of deep neural network training technologies, special emphasis by solving this problem began to be placed on the CNN-architectures (Convolutional Neural Networks) of various configurations. So, state-of-the-art results of emotion recognition were obtained for the eXnet [19] architecture. This network is used as a basic model in the proposed system. Its structure is shown in fig. 10.

In addition to choosing a model for recognizing emotions, the selection of the training dataset used is of great importance. So, in our work, a combined version of the training dataset from the well-known CK+ [20] dataset and datasets collected manually by the authors (the composition of the datasets is described in more detail below) is used. This approach allowed diversifying the training sample, making it less synthetic.

When forming the final dataset, images of faces with expressions of emotions from three basic sources were used:

1) The CK+ dataset. The training dataset consists of 4,615 images, the control dataset consists of 554 images structured according to 7 basic recognizable classes of emotions. This dataset consists of video fragments in the format of 640x490 or

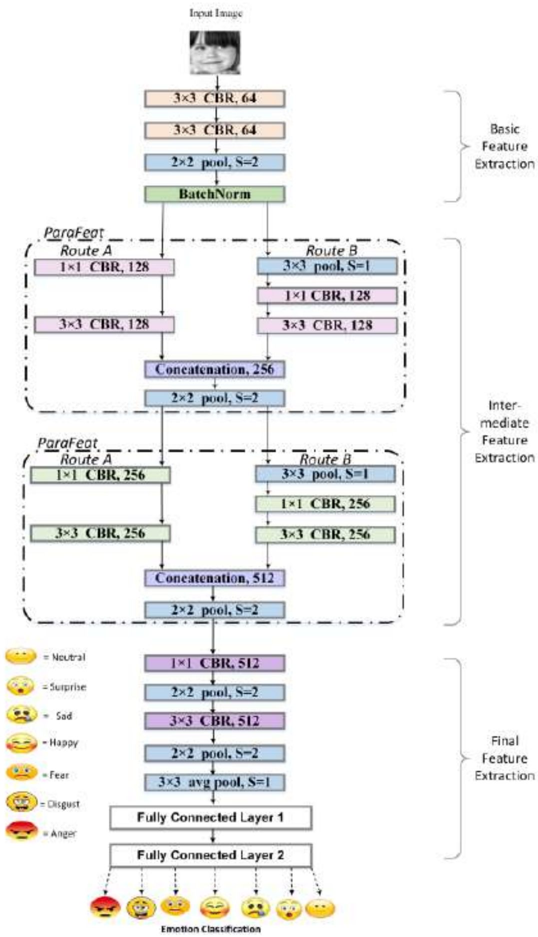


Figure 10. The structure of eXnet (original image from [19])

640x480 pixels. It should be noted, however, that the classes for this dataset differ from the above-mentioned ones (for example, instead of a neutral emotion, there is an emotion of despisal). This was one of the reasons why there was a need to form a combined dataset;

2) The "Students and colleagues" dataset. This dataset was collected from data obtained by self-determined image collection and processing. It is a set of images of the faces of 38 people with the expression of basic emotions. The training part included 20,947 images, the control part – 2,101;
3) Internet Emotions. This dataset was formed from images taken from the Internet and includes one-man photos of 295 people, which include training (268 images) and control (27 images) datasets.

Thus, the total size of the training dataset was 25,830 images and the control one – 2,682.

As a result of additional training of the eXnet model on the dataset described above, the results presented in table I were obtained.

avg valid – the percentage of successfully recognized emotions taken as 1 in the test dataset.

avg softmax – the percentage of successfully recognized emotions, taking into account the obtained probability from the range (0..1] in the test dataset.

ck+, **student_colleague** and **internet** denote the overall accuracy that results from the corresponding test datasets. **average_valid** and **average_valid_softmax** are the metrics used to evaluate accuracy.

Table I
RESULTS OF TRAINING THE EXNET MODEL

	avg. valid	*avg. valid softmax*
CK+	0.877	0.859
Students and Colleagues	0.765	0.742
Internet	0.37	0.305

F. Semantic analysis

The described integration mechanism allows enriching the knowledge base with the results of recognition of various models used by the computer vision module (identification model and emotion recognition model). The processing of this knowledge will be no different from the processing of any other knowledge in the ostis-system, regardless of whether they got there from the computer vision module, any sensors, visual or natural language interface or in some other way. In this case, computer vision is another receptor of the system.

Knowledge processing in the KB, i.e., semantic analysis, is performed by the problem solver. The problem solver is a set of agents that react to events in the knowledge base (for example, a problem definition), solve its problem (generating, transforming knowledge, accessing external systems) and put the result of the work in the same KB.

For example, one of the methods of knowledge processing can be the usage of logical inference [21], which generates new knowledge based on a set of rules. Logical rules, in the simplest case, can be represented by "if-then" bindings, where the "if" part describes the knowledge that must be in the knowledge base to make it possible for us to generate the knowledge described in the "then" part. The origin of such rules can be different: from adding them manually by knowledge base engineers to automatically generating them.

In the considered implementation of the hybrid system, the logical rules [21] are used to generate some standard system responses to the interlocutor's messages. These rules use such knowledge as the identification of the interlocutor and their current emotion. Figure 11 shows a fragment of such a rule in a simplified form for clarity (in a real system, such rules have a more complex specification).

The meaning of the rule is as follows: if we received a greeting message from a user, whose emotion is recognized by the system as "sadness" and whose name the

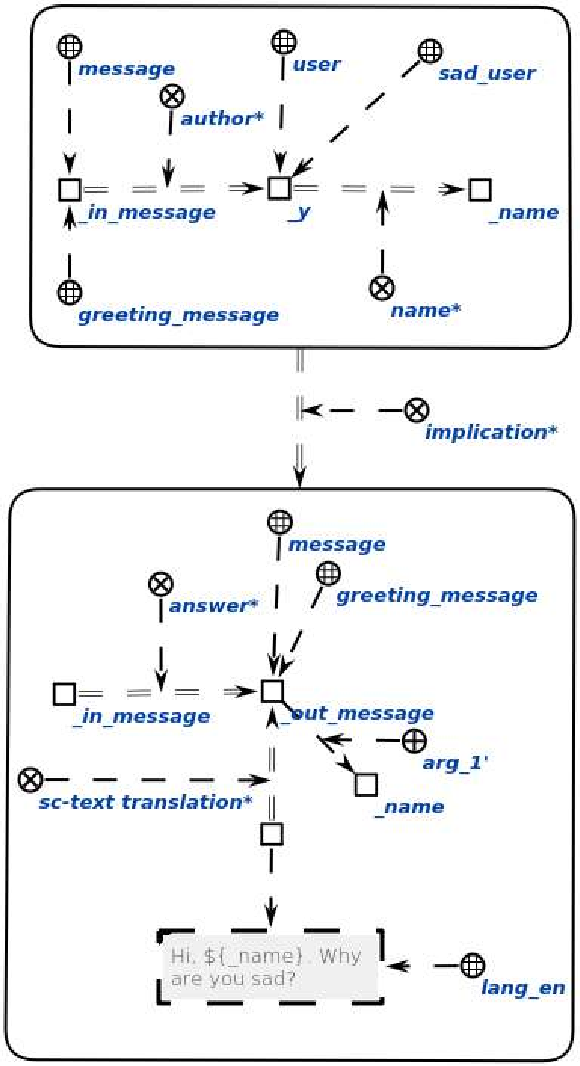


Figure 11. An example of a logical rule that uses the result of recognizing a user's emotion

system knows, then we need to respond to this message with a greeting with a reference by name and ask the reason for sadness.

V. Hardware architecture

The implementation of the represented models within the framework of a hybrid system for analyzing the emotional state requires appropriate support, both from the software and hardware. Therefore, the issue of creating a hardware architecture of the system that allows effectively implementing the functional responsibilities imposed on the system is one of the significant stages for achieving the overall goal.

The developed hardware architecture of the system should take into account the requirements and features of the implementation of both the semantic and neural network components of the system.

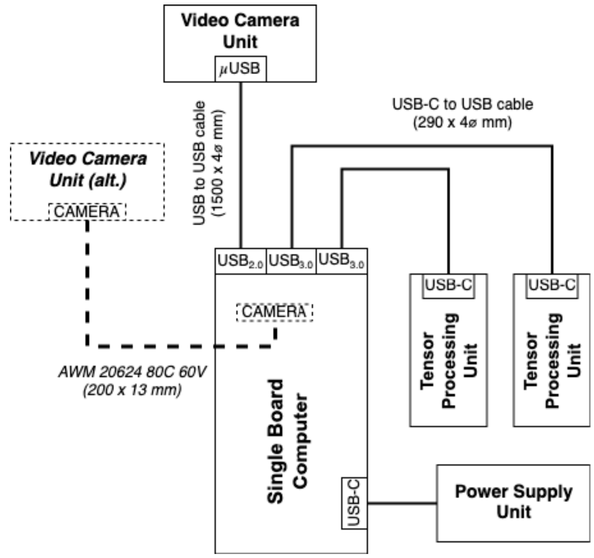


Figure 12. The system hardware architecture

The hardware requirements, from the point of view of the semantic part of the system, include the need to use computing tools with a processor based on the 'x86' architecture. This is due to the fact that initially the OSTIS platform as the basis of the software part of the system was developed for general-purpose CISC processors. Using this type of processor allows eliminating compatibility problems, simplifying debugging and testing the system. Therefore, it is necessary to have a computing device in the system based on this hardware architecture.

On the other hand, to solve computer vision problems, modern neural network architectures require support for tensor operations from the hardware platform, which allows effectively organizing the process of running trained neural network models on the target device. However, general-purpose processors are not suitable for performing such operations with maximum performance. Therefore, to increase the speed of the system, it is necessary to include a special coprocessor device in the hardware architecture, which allows increasing the speed of the neural network part of the system.

Taking into account the listed requirements, we have developed a hardware architecture of the system, the block diagram of which is shown in figure 12:

The main elements that build up the hardware of the system are:

- A single-board computer (SBC) that serves as a central device, on which the OSTIS virtual machine is run and, accordingly, the interpretation of intelligent agents and a list of peripheral devices that perform the functions of input and output of video and audio information as well as auxiliary devices that perform the functions of supporting neural network computing is carried out;

- To input video information, a camera (Video Camera Unit – VCU) is used designed to solve computer vision problems: detecting and tracking the user's face in the frame, identifying the user, recognizing the emotions of the subject in the frame based on frames from the video stream. According to the general scheme of the system model 1, the camera transmits the video stream to the computer vision module (computer vision pipeline), and only then the information is sent to the internal part of the system for further processing. The peculiarity of the proposed solution, which gives it additional flexibility, is that the computer vision module can be deployed both on the single-board computer itself and on a separate machine (for example, a PC or a specialized device designed to solve problems of this type). Therefore, the camera can be connected both to a single-board computer directly and to an external device that will transmit information through the abovementioned module to the central device where the OSTIS system core is run. The scheme of the hardware architecture considers the option when the camera is connected directly;
- To speed up the performance of operations on vectors and matrices when calculating neural networks on hardware with limited resources, which include single-board computers, it is proposed to use tensor coprocessors (Tensor Processing Unit – TPU) for neural network calculations.

Let us focus on each of the hardware components of the system and consider them in more detail.

A. Single-board computer

A single-board computer (SBC) is a computer set up on a single printed circuit board, on which a microprocessor, RAM, I/O systems and other modules necessary for the operation of a computer are installed [22], [23].

As the basis of the hardware platform, it was proposed to use a single-board computer, since such a form factor, on the one hand, can provide the necessary and sufficient runtime environment for the OSTIS Technology in terms of performance and functionality and, on the other hand, preserve the minimum weight-and-dimensional and cost characteristics of computing tools, including for solving problems connected with the Internet of things and the usage of OSTIS within the framework of the "Edge Computing and AI" [24], [25] concept.

We have reviewed and compared the models of single-board computers on the local and international markets that are available for delivery to the territory of the Republic of Belarus. The model lines of computers from such manufacturers as Interl NUX, LattePanda, Rock Pi, UDOO, ODYSSEY [26] were considered. We have set the maximum cost of a single-board computer, so that it does not significantly exceed the cost of the option for the ARM architecture. The set cost was no more than \$100, which significantly narrowed the search area.

Figure 13. A single-board computer "Rock PI X Model B"

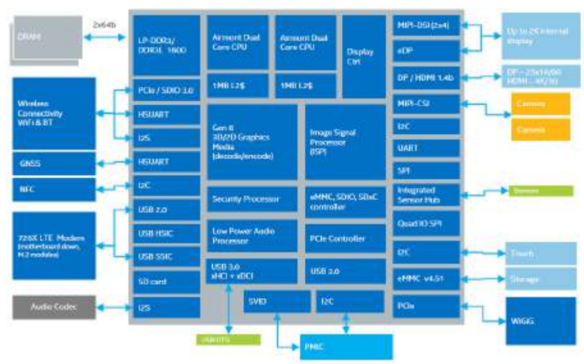


Figure 14. A functional diagram of the Intel Atom X5 processor [28], [29]

Among the currently available models of single-board computers, the Rock Pi computer has become the most preferred option, namely the Rock PI X Model B model [27], the appearance of which is shown in fig. 13.

The functional diagram of the CPU [28], [29] is shown in figure 14.

A distinctive feature of this single-board computer is the presence of ROM based on eMMC, which allows ensuring the functioning of the system and high-speed access to data on a solid-state drive without using an external SD drive. The maximum available capacity is 128 GB. The proposed architecture uses a version with 32 GB of memory, which is sufficient to contain the OS as well as the necessary software modules and OSTIS intelligent agents.

B. Video camera

It is an element of the system, through which a video stream is received and transmitted to a single-board computer to solve subproblems connected with computer vision and recognition of visual images. It acts as the

Figure 15. The Sony PS3 Eye camera

Figure 16. The Coral USB Accelerator TPU device

main channel of input information for the neural network modules of the system.

Within the current version of the hardware architecture, it is possible to connect various video cameras, depending on the type of video interface and the configuration of the device package, which will be directly determined by the type of problem being solved and the requirements for viewing angles, focal point, depth of field of the shown space, dimensions and scale of recognized objects in the image. The main difference between them is the design of the camera box itself as well as the type of interface that is used to connect them.

The camera is a separate device in its box, which can be set on a tripod and shoot at the level of the user's face. As such a removable camera, it is proposed to use the Sony PS3 Eye device (fig. 15) [30], [31].

It should be noted that both cameras are used in a mode of obtaining images of relatively low resolution, i.e., 640 x 480 60 fps, 320 x 240 120 fps. This is done for two purposes: the first is to increase the performance of the video subsystem as well as to prevent the image obtaining process, which is resource-consuming for single-board computer processors, from becoming a "bottleneck" in the common processing pipeline; the second is that the image will be transformed into images with a lower resolution one way or another before processing by neural network models. Such a transformation is called "oversampling", or "resampling", that is usually performed in all machine vision systems to ensure a balance between performance and quality of work, since processing large-resolution images requires significantly large computing resources as well as time for training and performing neural network models. In the case of our proposed system, image resampling is carried out for user identification models in the format of 160x160 pixels as well as in the format of 48x48 pixels – for emotion recognition models. For this reason, for the development of the hardware architecture, we chose the Sony PS3 Eye camera, which is significantly inferior in characteristics to modern cameras but, on the other hand, has a minimum cost on the market relative to the quality of the optical system and the CCD-matrix installed in it, which allow solving the entire range of denoted machine vision problems.

C. Neural network computing accelerator

An important element of the hardware architecture is a tensor coprocessor for neural network computing to speed up the work of the neural network modules of the application [32].

We have considered the model lines of tensor coprocessors of the main manufacturers of specialized purpose processors from Intel, NVidia and Google [33], [34]. The most suitable option in terms of performance characteristics, cost, amount of documentation and open source projects is the "Coral USB Accelerator" processor, which was chosen as this component of the system [35].

This processor was developed by the Google corporation and is intended for usage with the TensorFlow machine learning library. This device, in comparison with GPUs, is designed to perform a large number of calculations with reduced accuracy (in integer arithmetic) with higher performance per watt. The device is implemented as a matrix multiplier controlled by the instructions of the central processor over the USB 3.0 bus.

The Coral USB Accelerator coprocessor can perform 4 trillion operations (teraflops) per second (TOPS), using 0.5 watts for each TOPS (2 TOPS per watt). For example, it can perform tensor calculations for one of the most popular neural network architectures in technical vision problems, such as "MobileNet v2", with a performance close to 400 frames per second with a low energy consumption of about 1.5 W [36].

Conclusion

The proposed model of neuro-symbolic AI is an example of combining different directions of AI. This model allows using various neural network models to solve computer vision problems and conduct a semantic analysis of the results of the work of such models in the knowledge base. It is also possible to add new neural network models and control their operation mode through the knowledge base.

The proposed model is used to implement a hybrid system of semantic analysis of the emotional state of the user, which operates with knowledge formed in the process of interaction with neural network models.

In the article, a variant of a hardware platform for the operation of the developed system based on single-board computers of the "Raspberry Pi 4B" and "Rocks Pi X" models as well as "Raspberry Pi Camera v1.3" and "Sony PS3 Eye" video cameras and the "Google Coral USB Accelerator" tensor processor is also proposed.

The proposed hardware and software architecture provides the necessary level of performance and mobility for the semantic and neural network parts of the system.

The described model creates the basis for further research in the field of developing:

- universal integration with the knowledge base of any neural network models (not only solve computer vision problems);
- an approach to deeper integration of neural network models with the knowledge base, when through the knowledge base it becomes possible to control not only the operating mode of neural network models but also their topology, architecture, combination with other models, etc.;
- an approach to automatic decision-making on the usage of a particular neural network model for solving system problems;
- an approach to usage of the knowledge base to improve the training of artificial neural networks;
- new hardware architectures that can support such systems.

Acknowledgment

The authors would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help and valuable comments. In particular, Vladimir Adamovich Golovko, Vladimir Vasilyevich Golenkov, Daniil Vyacheslavovich Shunkevich. We also thank Intelligent Semantic Systems LLC for providing the equipment for conducting research.

References

[1] V. Golovko, A. Kroshchanka, M. Kovalev, V. Taberko, and D. Ivaniuk Neuro-Symbolic Artificial Intelligence: Application for Control the Quality of Product Labeling, Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2020, pp. 81–101.

[2] Castelvecchi D. (2016). Can we open the black box of AI?. Nature, 538(7623), pp. 20–23. https://doi.org/10.1038/538020a

[3] J. Sowa Semantic networks, in encyclopedia of artificial intelligence, Expert Systems with Applications (1987)

[4] F. Lehmann Semantic networks, Computers Math. Applic 23(2–5), 1–50 (1992)

[5] T. Besold, A. d'Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K. Kuehnberger, L. Lamb, D. Lowd, P. Lima, L. de Penning, G. Pinkas, H. Poon, G. Zaverucha Neural-symbolic learning and reasoning: A survey and interpretation, (Nov 2017), https://arxiv.org/pdf/1711.03902.pdf, (accessed 2021, Jun)

[6] Garcez, A., Besold, T. R., Raedt, L., Foldiak, P., Hitzler, P., Icard, T., Kuhnberger, K-U., Lamb, L. C., Miikkulainen, R. and Silver, D. L. Neural-Symbolic Learning and Reasoning: Contributions and Challenges, AAAI Spring Symposium Series, 23-03-2015

[7] A. d'Avila Garcez, L. Lamb, D. Gabbay Neural-symbolic cognitive reasoning, In: Cognitive Technologies, Springer (2009)

[8] B. Hammer, P. Hitzler Perspectives of neural-symbolic integration, In: Studies in Computational Intelligence (77), Springer (2007)

[9] V. Golovko, V. Golenkov, V. Ivashenko, V. Taberko, S. Ivaniuk, A. Kroshchanka, M. Kovalev Integration of artificial neural networks and knowledge bases, Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2018, pp. 133–145

[10] V. Golenkov, N. Guliakina, I. Davydenko, and A. Eremeev, Methodsand tools for ensuring compatibility of computer systems, Otkrytyesemanticheskie tekhnologii proektirovaniya intellektual'nykh system[Open semantic technologies for intelligent systems], 2019, pp. 25–52.

[11] Intelligent meta system(IMS). Available at: http://ims.ostis.net (accessed 2021, Jun)

[12] F. Schroff, D. Kalenichenko, and J. Philbin FaceNet: A Unified Embedding for Face Recognition and Clustering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815–823.

[13] K. He, X. Zhang, S. Ren, and J. Sun Deep Residual Learning for Image Recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[14] D. E. King Dlib-ml: A Machine Learning Toolkit, Journal of Machine Learning Research, vol. 10, 2009, pp. 1755–1758.

[15] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks, in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[16] C. F. Benitez-Quiroz, R. Srinivasan and A. M. Martinez EmotioNet: An Accurate, Real-Time Algorithm for the Automatic Annotation of a Million Facial Expressions in the Wild, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5562–5570.

[17] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition, in arXiv, arXiv:1905.04075v2 [cs.CV], https://arxiv.org/abs/1905.04075, 2019.

[18] S. Shojaeilangari, W. Yau, K. Nandakumar, J. Li, and E. K. Teoh Robust Representation and Recognition of Facial Emotions Using Extreme Sparse Learning, in IEEE Transactions on Image Processing, vol. 24, no. 7, pp. 2140–2152, July 2015.

[19] M. N. Riaz, Y. Shen, M. Sohail, and M. Guo eXnet: An Efficient Approach for Emotion Recognition in the Wild, in Sensors, vol. 20(4), 2020.

[20] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops, 2010, pp. 94–101.

[21] V. Vagin, A. Zagoryanskaya, M. Fomina Dostovernii i pravdopodobnii vivod v intellektualnih sistemah, Moscow: FIZMATLIT, (in Russian) p. 704 p. (2008)

[22] Single-Board Computer. Available at: https://en.wikipedia.org/wiki/Single-board_computer (accessed 2021, May)

[23] U. Isikdag Internet of Things: Single-board computers, Enhanced Building Information Models, Springer Cham, 2015. pp. 43–53.

[24] P. Li Arhitektura interneta veshhej, DMK Press (in Russian), 2018, 454 p.
[25] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, & X. Chen Edge AI: Convergence of Edge Computing and Artificial Intelligence, Springer Nature, 2020, 218 p.
[26] P. Galkin, L.Golovkina, I. Klyuchnyk Analysis of single-board computers for IoT and IIoT solutions in embedded control systems, 2018 International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), IEEE, 2018, pp. 297–302.
[27] ROCK Pi X Hardware Information. Available at: https://wiki.radxa.com/RockpiX/hardware (accessed 2021, May)
[28] Intel Atom Z8000 Processor Series Vol. 1. Available at: https://dl.radxa.com/rockpix/docs/hw/atom-z8000-datasheet-vol-1.pdf (accessed 2021, May)
[29] Intel Atom Z8000 Processor Series Vol. 2. Available at: https://dl.radxa.com/rockpix/docs/hw/atom-z8000-datasheet-vol-2.pdf (accessed 2021, May)
[30] Play Station Eye. Available at: https://en.wikipedia.org/wiki/PlayStation_Eye (accessed 2021, May)
[31] PS3 Eye Camera Technical Specification. Available at: https://icecat.biz/en/p/sony/9473459/webcams-eye-camera-+ps3-1269549.html (accessed 2021, May)
[32] AI accelerator. Available at: https://en.wikipedia.org/wiki/AI_accelerator (accessed 2021, May)
[33] Y. Chen, Y. Xie, L. Song, F. Chen, T. Tang A survey of accelerator architectures for deep neural networks, Engineering, vol. 6 no. 3, 2020, pp. 264–274.
[34] W. Li, M. Liewig et al. A Survey of AI Accelerators for Edge Environment, Trends and Innovations in Information Systems and Technologies. WorldCIST 2020. Advances in Intelligent Systems and Computing, vol 1160. Springer, Cham, 2020, pp 35–44.
[35] Coral USB Accelerator Datasheet. Available at: https://coral.ai/docs/accelerator/datasheet (accessed 2021, May)
[36] Edge TPU Performance Benchmark. Available at: https://coral.ai/docs/edgetpu/benchmarks (accessed 2021, May)

Семантический анализ видео-потока на основании нейро-символического искусственного интеллекта

Крощенко А.А., Михно Е.В.,
Ковалев М.В., Захарьев В.А., Загорский А.Г.

Статья посвящена модели компьютерного зрения, базирующейся на нейро-символическом подходе. Приведена архитектура предлагаемой модели с подробным описанием составляющих ее компонентов. Описаны основные применения преимущества подобной модели на примере диалоговых систем. Во второй части работы приводится пример разработки системы-компонента диалоговой системы для оценки эмоционального состояния пользователя, базирующейся на предложенной нейро-символической модели. Показано, что класс подобных систем сочетает в себе преимущества коннекционистского и символического подхода в искусственном интеллекте. Приводится обзор аппаратной платформы, позволяющей осуществлять запуск и поддержку работы системы в компактном форм-факторе одноплатного компьютера.

Received 09.07.2021

The role of semantic and ontological networks in the digital twin management in manufacturing

Alona Shyshak, Nataliia Lutska, Lidiia Vlasenko, Oleksandr Pupena
the Department of Automation and
Computer Technologies of Control Systems
National University of Food Technologies
Kyiv, Ukraine
al_sh_94@ukr.net, lutskanm2017@gmail.com, vlasenko.lidia1@gmail.com, pupena_san@ukr.net

***Abstract*—The paper is devoted to present an overview of the current state of the digital twin development in manufacturing domain in order to determine the place of semantic and ontological networks in digital twin structure. In particular, the current international standards, technical reports and recommendations in the field of Industry 4.0 and Industrial Internet of Things regarding digital twins are considered and analyzed.**

The article is divided into 3 parts: the first part deals with the concept and main functions of digital twin from different points of view, including standards, the second part concentrates on modern systematized, standardized and applied functions of semantic and ontological networks in industry, in the third part the authors present their vision regarding the role of semantic and ontological networks in the digital twin development and implementation in industry.

Keywords – digital twin, Industry 4.0, Industrial Internet of Things, semantic, ontology.

I. Introduction

Today, the technologies of the fourth industrial revolution are becoming increasingly important in various industries. The technologies behind the Industry 4.0 and Industrial Internet of Things (IIoT) concepts offer new opportunities for products and services. Driven by Industry 4.0 and IIoT approaches and the development of Big Data analytics computing power, fast algorithms, and amount of available data allow to model and optimize the physical process with real-time control. The digital representation of the physical twin, known as a digital twin (DT), is one of the most important aspect of the fourth industrial revolution. DT accompanies the relative asset throughout its lifecycle from conception to disposal and even after disposal. For process control systems, which are considered as an asset as well, it is also advisable to develop a digital twin in combination with the controlled object. It is expected that at the operation stage the digital twin of such technological complexes will provide accurate prediction of their future behavior and will help to effectively maintain the quality of technological processes by easy visualization and integration of cognitive capabilities into the real system. However, there is currently no single methodological approach for the development and implementation of DT in manufacturing. Despite the presence of a large number of existing international standards, designers and developers of automated process control systems offer a variety of DT solutions that differ in purpose, functionality, architecture, etc. Alternatively, representatives of specialized software such as SCADA, CAD, MES/MOM offer variant solutions for the implementation of DT at its appropriate level of automation.

II. Digital twin: characteristics and standards

The digital twin concept was first introduced in 2002 at the University of Michigan Executive Course on Product Lifecycle Management by Michael Grieves. At that time, digital representations of actual physical products were relatively new and immature. In addition, the information being collected about the physical product as it was being produced was limited, manually collected, and mostly paper-based. According to the model proposed by Dr. Michael Grieves the Digital Twin concept model (Figure 1) consists of three parts [1]:

- physical products in Real Space;
- virtual products in Virtual Space;
- the connections of data and information that ties the virtual and real products together.

The White Paper, published by the Industrial Internet Consortium (IIC) and Plattform Industrie 4.0 [2], describes the applicability of the digital twin concept in various domains of the Industrial Internet of Things, including smart manufacturing, automotive, supply chains and logistics, building automation, smart cities and critical infrastructures. Today, Digital Twin Consortium is doing the promotion of the digital twins use in various fields, the development of digital twin technologies, the

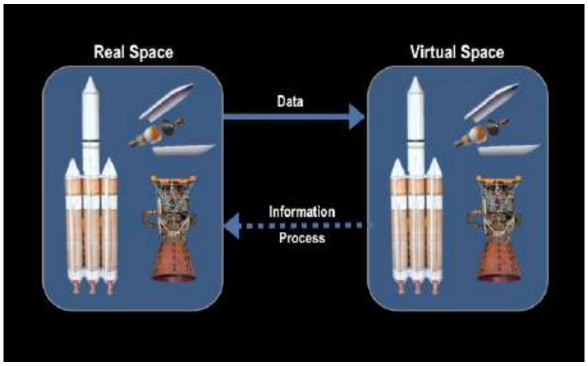


Figure 1. Digital Twin concept model by Dr. Michael Grieves (source – [1])

development of technical guidelines, and requirements for new standards in order to maximize the benefits of digital twins. Unlike IIC and Digital Twin Consortium a German platform Plattform Industrie 4.0 focuses on implementing the concept only in the manufacturing sector focusing on machine industry.

The Consortium released an official definition of a digital twin: “A digital twin is a virtual representation of real-world entities and processes, synchronized at a specified frequency and fidelity” [3]. In contrast to the concept model of Dr. Michael Grieves, the definition of digital twin focuses on real-world entities, because the things for which virtual representation should be provided can be immaterial things like organizations, supply-chains, work-orders. Official documents from Digital Twin Consortium have not been published yet. However, according to the organization’s work [4], a digital twin is implemented in a digital twin system. It consists of a virtual representation, services, service interfaces and various applications according to the purposes of digital twin use case. A virtual representation includes stored structured information that represents states and attributes of entities and processes, computational algorithms and supporting data that represent entities and processes from a dynamic perspective.

As part of the Industrie 4.0 initiative, the Reference architecture model industry 4.0 (RAMI4.0) was developed to build smart manufacturing. The model provides a representation of all enterprise assets, including personnel and software, in the network form of not hierarchical I4.0 components linked with each other by I4.0-compliant communication. I4.0 components are globally and uniquely identifiable participants capable of communication, and consist of the asset administration shell and the asset (as in Figure 2) with a digital connection within an I4.0 system. An I4.0 component can be a production system, an individual machine or unit, or a module within a machine [5]. Assets shall have a logical representation in the "information world" (virtual world), which is called the asset administration shell (AAS). In fact, AAS is an implementation of a digital twin for manufacturing applications. It has been defined and designed to ensure interoperability between companies throughout the value stream. ASS includes asset information collected throughout the lifecycle in a structured manner and real-time data. ASS consists of properties, which are grouped in a submodel according to the relevant domain. Such properties are standardized data elements. It is recommended to use repositories such as IEC CDD (common data dictionary) or ECLASS to define such properties [6]. The worldwide unique identifier associates a property type with a definition, which is a set of well-defined attributes.

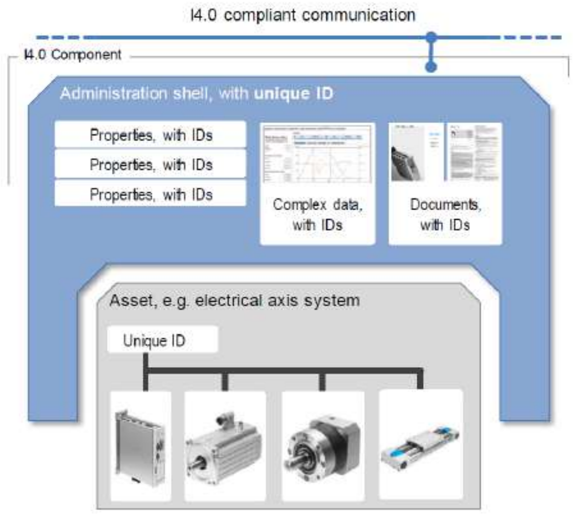


Figure 2. Asset administration shell (source – [6])

There are various activities regarding standardization of digital twins. IEC 62832-1 – “Digital Factory” is a well-established standard, which defines a digital factory framework with the representation of digital factory assets. A lot of standards are on the development stage, such as IEEE P2806.1 – “Standard for Connectivity Requirements of Digital Representation for Physical Objects in Factory Environments”, IEEE P2806 – “System Architecture of Digital Representation for Physical Objects in Factory Environments”, ISO/DIS 23247-1 “Automation systems and integration – Digital Twin framework for manufacturing – Part 1: Overview and general principles”, IEC 63278-1 ED1 Asset administration shell for industrial applications – Part 1: Administration shell structure.

III. Semantic and ontology: place and purpose

Due to the active digitalization of the industrial sector, in particular the widespread use of the Industrial Internet of Things and digital twins, some problems have been revealed:

- absence of a single unified industrial thesaurus;

- variety of large amounts of data that need to be processed;
- absence of a single digital twin model that describes all aspects of the system behavior and its parts from different points of view and is intended for different purposes.

As described in many sources, in particular in [7], there is no single thesaurus that will unite the various branches of the industrial sector. In particular, relevant identified standardization activities of ontological semantics used in IIoT are: eCl@ss (ISO 13584-42 and IEC CDD classes and properties), “Semanz4.0”, “AutomationML” (IEC 62714), “WSDL” (Web Services Description Language by W3C), IEC SC3D with IEC 61360, IEC 62656, IEC CDD (the semantic and ontological repository based on IEC 61360 and IEC 62656). On the one hand, the problem with the unity of the digital twin model lies in the need for availability of many different, unrelated and non-unified models. On the other hand, the interconnection of digital twins in a single system [8] requires their interaction, which requires the unification of such interaction at the conceptual level.

An effective way to overcome these problems is to use an ontology. Ontology is classified according to various criteria, including completeness, level of generalization, application domain, purpose or descriptive language. The need for an ontology is indicated in the follow resource [7], however, final view of ontology is not sensible. The only thing written in this document are two ways to use the integrated semantics: to provide an information model (related to a specific ontology) for querying or reasoning purposes, and to provide a system dynamics model supporting checking inconsistencies during interoperation. Both possible interpretations shall be related to semantics web descriptions.

Data models and ontology play different functional roles in the digital twin system. Persistence data models are proposed to be used for stored structured information. Service interfaces embody a logical data model that describes the data structures and types used by the API or protocol. Conceptual data models compatible with the general ontology can be used in the digital twin system that integrates information from several structured data repositories, each of which has persistence and logical data models. A variety of data modeling languages can be used, in particular Digital Twins Definition Language, OWL.

Standardization is involved in the formation of the upper-level ontology (for example, BFO – ISO/IEC 21838-2 [9]), ontology description languages (for example, RDF [10] and OWL [11]), as well as models of preservation and representation of a particular application domain (for example, ISO 15926 [12]). In addition, there are various applications of ontological networks, for example, OSTIS developed in accordance with ISA-88 [13] and highly specialized domains and cores [14].

The RAMI 4.0 architecture defines a number of W3C standards that reflect the semantic representation of the model data in the network, there are OWL, RDF, RDFS [15], SPARQL [16], RIF/SRWL [17, 18]. RDF, OWL and SPARQL underlie the semantic WEB. The first three technologies are used to represent metadata, the fourth one is used to query the ontological database, and the fifth one is used to form axioms and rule interchange format. Such semantics have a number of advantages, such as reasoning over data and working with heterogeneous data sources, the single structure that allows data to be shared and reused across application boundaries.

In fact, these recommendations are not enough to build a comprehensive distributed network for IIoT. Therefore, new recommendations from W3C covering IoT appeared since 2017. They include: OWL-Time [19], SSN ontology [20] and WoT [21]. OWL-Time describes the subclasses and properties of the two generalized classes of temporal position and temporal duration of resources on WEB pages. SSN ontology describes an ontology of sensors and actuators based on a self-contained core ontology (Sensor, Observation, Sample, and Actuator – SOSA) and defines several conceptual modules that cover the key concepts of sensor, actuation and sampling: observation/actuation/sampling, deployment, system, system property, feature, condition, procedure, result. WoT recommends a general concept for the existence of IoT in the network. In particular, thing description, interaction model, protocol binding, WoT interface, etc. are described. Thus, a set of templates for Web of Things is formed.

IV. Results and suggestions

Based on the analysis, the following conclusions can be drawn:

- there are problems that need to be solved by means of unified semantic and ontological approaches during the development of DT;
- there are DT developments based on ontological models in scientific articles, but there is no single approach;
- in the context of DT ontology can be used as:
  - a knowledge base that stores information throughout the asset lifecycle – structured storage for DT;
  - a knowledge base for the application domain modeling, that deals with a regular historical database and extracts the necessary knowledge – a framework for structuring different types of DT models;
  - a database for presenting metainformation about DT to provide the adapted interaction with other components of the system – WoT;

– a mean of DT information support, that works with open ontological databases in the network and serves to provide supporting information – semantic WEB, OSTIS, etc.

However, today there are very few implementations of ontological approaches in DT. In authors‘ opinion the most promising DT model should have: ease of understanding and implementation, openness, a set of ready-made components for deployment (frameworks) and significant added value.

V. Conclusion

The paper discusses some standards and documents regarding the DT development from leading industry organizations and committees such as ISO/IEC, W3C. It has been found that the use of semantic and ontological networks to integrate and formalize different types of DT parts is offered by standards and applied solutions. However, there is currently no single systematized solution.

The authors come to the conclusion that a number of recommendations from W3C is the most promising direction of applying semantic and ontological networks during the DT development. W3C recommendations have significant advantages and can be used today because they provide interoperability, thus it will help to connect parts of different distributed components based on unified interfaces or protocols.

References

[1] M. Grieves Digital twin: manufacturing excellence through virtual factory replication, White paper, 2015.

[2] B. Boss, S. Malakuti, S.W. Lin, T. Usländer, E. Clauer, M. Hoffmeister, L. Stojanovic Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0, An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper, 2020.

[3] Digital Twin Consortium Defines Digital Twin - Digital Reflections Digital Twin Consortium. Available at: https://blog.digitaltwinconsortium.org/2020/12/digital-twin-consortium-defines-digital-twin.html (accessed 2021, May).

[4] DigitalTwinSystems. Available at: https://www.digitaltwinconsortium.org/glossary/index.htm#digital-twin-system (accessed 2021, May).

[5] DIN SPEC 91345:2016-04. Reference Architecture Model Industrie 4.0 (RAMI4.0).

[6] SPECIFICATION Details of the Asset Administration Shell. Part 1 – The exchange of information between partners in the value chain of Industrie 4.0.

[7] ISO/IEC TR 30166:2020. Internet of things (IoT) – Industrial IoT.

[8] Digital Twins for Industrial Applications, An Industrial Internet Consortium White Paper, 2018.

[9] International Organization for Standardization. ISO/IEC 21838-2 Information technology – top-level ontologies – Part 2: Basic Formal Ontology, 2019.

[10] World Wide Web Consortium (W3C). Resource Description Framework. Model and Syntax Specification, 1999. Available at: https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/ (accessed 2021, May).

[11] World Wide Web Consortium (W3C). OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax, 2nd Edition, 2012. Available at: https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/ (accessed 2021, May).

[12] International Organization for Standardization. ISO 15926-13:2018 Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities Part 1 – 13, 2012-2021.

[13] V. Taberko, D. Ivaniuk, D. Shunkevich, O. Pupena Ontological Approach for Standards Development Within Industry 4.0. International Conference on Open Semantic Technologies for Intelligent Systems, OSTIS 2020: Open Semantic Technologies for Intelligent System, pp 64-80.

[14] N. Lutskaya, L.Vlasenko, N.Zaiets, V. Shtepa Ontological Aspects of Developing Robust Control Systems for Technological Objects. Intelligent Computing and Optimization. ICO 2020. Advances in Intelligent Systems and Computing, vol 1324. Springer, https://doi.org/10.1007/978-3-030-68154-8_107.

[15] World Wide Web Consortium (W3C). Resource Description Framework (RDF) Schema Specification, 1999. Available at: https://www.w3.org/TR/WD-rdf-schema/ (accessed 2021, May).

[16] World Wide Web Consortium (W3C). SPARQL 1.1 Overview, 2013. Available at: https://www.w3.org/TR/2013/REC-sparql11-overview-20130321/ (accessed 2021, May).

[17] World Wide Web Consortium (W3C). RIF Overview (Second Edition), 2013. Available at: https://www.w3.org/TR/rif-overview/ (accessed 2021, May).

[18] World Wide Web Consortium (W3C). SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004. Available at: https://www.w3.org/Submission/SWRL/ (accessed 2021, May).

[19] World Wide Web Consortium (W3C). Time Ontology in OWL, 2020. Available at: https://www.w3.org/TR/owl-time/ (accessed 2021, May).

[20] World Wide Web Consortium (W3C). Semantic Sensor Network Ontology, 2017. Available at: https://www.w3.org/TR/vocab-ssn/#Sampling-overview (accessed 2021, May).

[21] World Wide Web Consortium (W3C). Web of Things (WoT) Thing Description, 2020. Available at: https://www.w3.org/TR/wot-thing-description/ (accessed 2021, May).

Роль семантических и онтологических сетей при организации и реализации цифровых двойников промышленных предприятий

Шишак А.В., Луцкая Н.Н.,
Власенко Л.А., Пупена А.Н.

В работе представлен обзор современного состояния области разработки цифровых двойников промышленных предприятий с целью определения семантических и онтологических сетей в их структуре. В частности рассмотрены и проанализированы существующие на сегодня международные стандарты, технические отчеты и рекомендации в области Industry 4.0 и Industrial Internet of Things по направлению, которое касается цифровых двойников.

Статья структурирована следующим образом: в первой части приводится концепция и основные функции цифровых двойников с разных точек зрения, включая стандарты, во второй части рассмотрены современные систематизированые, стандартизированные и прикладные функции семантических и онтологических сетей в промышленности, в третьей части - авторы приводят свое видение и роль онтологических сетей при проектировании и реализации цифровых двойников в промышленности.

Received 31.05.2021

Principles of building a system for automating the activities of a process engineer based on an ontological approach within the framework of the Industry 4.0 concept

Valery Taberko, Dzmitry Ivaniuk
JSC "Savushkin Product"
Brest, Republic of Belarus
tab@pda.savushkin.by,
id@pda.savushkin.by

Nikita Zotov, Maksim Orlov
Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
nikita.zotov.belarus@gmail.com,
orlovmassimo@gmail.com

Oleksandr Pupena, Nataliia Lutska
National University of Food Technologies
Kyiv, Ukraine
pupena_san@ukr.net,
lutskanm2017@gmail.com

***Abstract*—In this article, an approach to the continuous development of automation of the processes of creating, developing and applying standards based on the OSTIS Technology is proposed. Examples of these processes due to the involvement of end-users of the system using the tools and mechanisms of the OSTIS Technology are considered. Examples of further formalization of standards within the framework of the proposed approach are given.**

***Keywords*—automation of manufacturing processes, information service, ontological production model, Industry 4.0, ontology, knowledge base, OSTIS Technology**

I. Introduction

The implementation of the Industry 4.0 concept at production facilities is accompanied by the development of a single ontological production model, which is the core of the complex information service of the enterprise. At the first stage of developing such an enterprise model, it is necessary to nest data on the lower level of production, namely on the manufacturing process and equipment. As the source of this data, P&ID-schemes of production can serve. Thus, the formalization of the ISA 5.1 [1] standard is necessary to work with P&ID-schemes, which are widely used in control systems together with the ISA 88 [2] standard and allow describing the lower level of production in full. At the same time, it is also necessary to consider the approach of formalization of the subject domain based on the ISO 15926 [3], [4] standard, which describes the integration of data on the life cycle of processing enterprises into a single ontological storage. New users will be added: an automation engineer and a master, who implement the new capability of the intelligent search together with the developed model. For the current user – the operator of the manufacturing process – the implementation of the mechanism for obtaining intelligent information that covers both particular and common issues of the manufacturing process, equipment, components and automated control systems becomes relevant. In this article, attention is paid to the continuous development of a system of complex information services by employees of a formulating enterprise on the example of the JSC "Savushkin product" using an Open semantic technology for intelligent systems. This article uses and develops the results represented in [5], [6].

II. Briefly about ISA-5.1

This standard describes the rules for drawing up functional schemes for the automation of manufacturing processes. Such schemes allow the graphical representation of the production technology and equipment as well as define the rules for identifying equipment and measuring and automation tools for design and service purposes. Figure 1 shows an example of a functional scheme.

The functional scheme shows: the coagulator itself (the unit), the lines (the machine) and the valves (the control device). Different colors indicate the purpose of the lines (red – washing, blue – mixture, green – whey, black – product). This fragment allows getting an insight into which devices are used and how they are connected.

III. Ontologies in production

The ISA-88 article described how to use the knowledge base on the basis of the OSTIS [7] ontologies to train the operator with complex concepts, search for objects according to ISA-88 and their interrelations. The need for knowledge bases for production is not restricted to the above. Among the most complex problems that can be solved using knowledge bases on the basis of ontologies, there are:

- decision support in unforeseen situations as well as start-ups and ends;

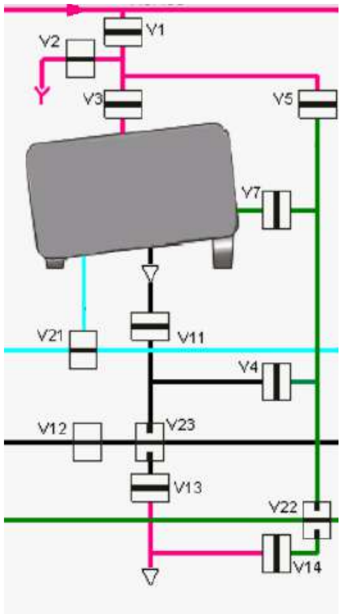


Figure 1. A functional scheme of the coagulator

- the determination of equipment failures and their causes;
- integration between systems with different data representation and functions of engineering problems.

Batch production is characterized by a high complexity of control systems – they should take into account the possibility of flexible development of various products on the same equipment. At the same time, it should be possible to develop the technology without making changes to the control program. This is why the ISA-88 standard was developed, which includes the best automation practices for this type of production. However, in addition to the fact that the creation of such systems requires a good understanding of the standard, the success of implementation depends on taking into account the capabilities of the equipment and the requirements of the technology. According to the standard, the automation of manufacturing operations and the running of equipment for their implementation are two interrelated but separate processes. This relation and engineering processes are shown in the ISA-88 standard in figure 2.

The choice of the role of the equipment in ISA-88 depends on what function it will perform in the production process. Thus, the technological procedural elements should be known. The equipment is designed and at the same time used to perform manufacturing operations. The knowledge base should be created to assist the technologist when generating a recipe (using PFC), that is, to provide the necessary choice of equipment, answering such questions:

- "What equipment can perform such a list of classes of procedural elements?";

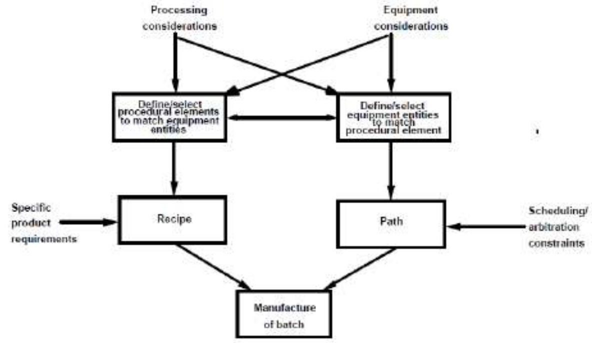


Figure 2. Relations and engineering processes of automation of manufacturing operations and the running of equipment

- "What methods can be available for developing a product?";
- "Which method is the most optimal from the point of view of the selected criterion?".

Similarly, the definition of procedural elements as the building blocks of recipes depends on the decomposition and capabilities of the equipment. For example, if the procedural element "preparation of a mixture of A, B, C" is created, then only the equipment that can automatically prepare a mixture from at least three components is required to use it. If a procedural element "batch up a component" is created, then the equipment that does not have automatic batching up functionality can perform manual operations or use the equipment sequentially. On the other hand, dividing into very small procedural elements can lead to lengthy and impractical recipes. A limiting case is the usage of a procedural element as a reference to basic functions, such as "unseat the valve", which does not correspond to the standard at all.

There are even more complex problems of creating an equipment hierarchy. Here are some examples. Let us consider the conditional P&ID-scheme of developing a food product. Though it is simplified, it is quite difficult to perform the decomposition of equipment. The criteria for creating the ISA-88 equipment hierarchy are considered below. There is one significant problem. It is logical to refer the valves that are located on the positions of the inlet to the tanks and the offset from the tanks to the tank as control modules, forming a unit. But what to do with the valves that are on the charging lines (V8, V7, V16, V23)? It cannot be said that V8 belongs to the unit with Tank 1, since it is controlled when loading any tank in the line. Then it might be logical to refer it to the process cell. But will it be convenient from the point of view of creating a recipe? In fact, there must be some kind of procedural control that coordinates the operation of these valves. Then it is necessary to create an equipment module that will combine the valves that participate in the same path (fig. 3).

What equipment should be referred to such paths? The

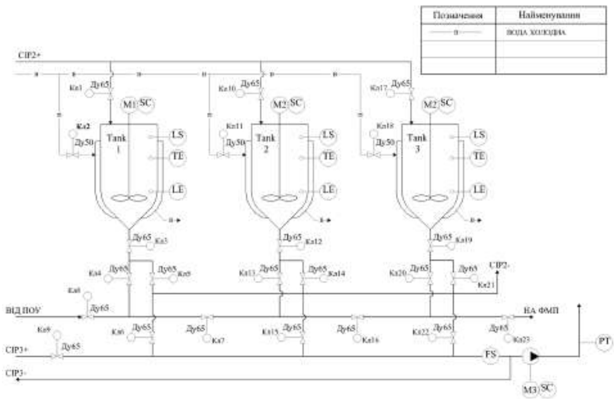


Figure 3. An example of an equipment module

answer to this question lies in the technological scheme itself. Even on such a simple scheme, it is easy to get confused about which of the control modules should be referred to which equipment and how to combine them into an equipment module or a unit. The analysis of many sources shows that the same problems lead to completely different solutions that can either fully comply with the standard or have certain differences from it. An engineer that develops a control system must take into account all technological connectivities and equipment specifications. This part is not part of the ISA-88 standard, it concerns more the field of equipment design.

Even the presence of a database can simplify and speed up the search process. But the best option would be the availability of a knowledge base about all production equipment. The developer of the control system could get the necessary information by asking the system various questions:

- to find the equipment that is involved in the charging/output/washing line of the tank;
- to find equipment at the crossing of the paths;
- to find the list of equipment for the specified set of conditions (for example, it is connected with Tank 1 and requires manual action).

One of the labor-intensive problems is the implementation of a Master Recipe for a certain process cell by the General or Site Recipe. The system implemented on the basis of the OSTIS Technology also significantly simplifies such a problem for the technologist.

Thus, in addition to the problem of creating a knowledge base for a system with ISA-88, there is a problem of creating a knowledge base of all production equipment. In the last 30 years, the issue of obtaining knowledge from industrial information, which the manufacture is aware of, has been actively addressed. One of the main and promising directions is the usage of ontological production models, which is confirmed by the development of international standards in this area.

The development and standardization of ontological systems was carried out by international organizations for standardization, such as ISO, IEEE, OMG, W3C and others. Some ontological structures have been developed, which, though have not been approved by international standards, have become standards de facto. They can be divided into several groups:

- ontological systems, models, languages and their parts for general and industrial purposes:
  - the ontological model of the hierarchical structure of production in the processing industry [3];
  - the technical dictionary [8];
  - a series of standards for the development of a top-level ontology [9], which is currently under development;
  - formal semantic models of global production networks [10];
  - a semantic approach for the computer-interpretable exchange of information related to production processes [11];
- general and specialized top-level ontologies:
  - MOF (Meta Object Facility) is a meta-object environment for model engineering [12];
  - BFO (Basic Formal Ontology) is a basic formal ontology, common for biomedicine [13];
  - ZEO (Zachman Enterprise Ontology) is the ontology of an enterprise for the description of its architecture;
  - DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) is a descriptive general-purpose ontology for linguistic and cognitive engineering, quite popular in the field of ontological engineering;
  - GFO (General Formal Ontology) [14];
  - SUMO (Suggested Upper Marged Ontology) is the proposed unified top-level ontology, a source document for a workgroup of IEEE employees from the fields of engineering, philosophy and computer science [15], [16];
- semantic Web: it includes all standards and rules for semantic processing of documents on the Internet, such as the Resource Description Framework (RDF) as well as its RDFS and RDF extensions; the Web Ontology Language OWL; the SPARQL query language; the Rule Interchange Format and a number of formats for saving RDF N-Triples, Turtle, RDF / XML, N-Quads, Notation 3 triples.

The basic (fundamental) ontology or a top-level ontology is a general ontology that is applied to various subject domains. It defines basic concepts, such as objects, relations, events, processes, etc. The most famous fundamental ontologies are listed above. BFO and DOLCE are the most commonly used in the development of engineering ontologies. These two ontologies are formal and provide a logical theory for representing common assumptions. When forming an ontology of a subject domain based on one of the specified top-level ontologies, it can be more easily integrated with other subject ontologies. The problem is that there are quite a lot of top-level ontologies and giving preference to one of them becomes a certain search problem that requires a lot of time and effort. In addition, some of them do not have open access and are also badly compatible with the semantic Web.

The ISO 15926 ontology [17] is considered separately. This standard is not only a top-level ontology but also a thesaurus of the processing industry, including the structure of retention and access to the ontological base. Standardization is implemented by using well-defined templates for technical and operational information, that include classes and relations of the invariant and temporal parts of the ontology. The advantages of this ontological model are the typification and identification of data located on the Internet; information is stored in RDF-triplets, access to triplet storages occurs using the SPARQL query language, etc. When creating this model, the developers tried to cover all aspects of requests that may arise in manufacturing. As a result, the model has hundreds of nested classes and attributes at the lower levels of production (description of technological equipment), most of which may not be used in practice. The temporal part increases the complexity of the model several times.

Thus, the ontology according to the ISO 15926 standard is most suitable for the specified problem. However, it should be noted that, taking into account the need for a common equipment and ISA-88 knowledge base, it was decided to implement the equipment knowledge base using OSTIS. In addition, there are restrictions in the ISO 15926 standard that are not present in the OSTIS Technology.

IV. About ISA-88 and the Criteria for the Decomposition of Technology and Equipment

As already noted above, when creating the equipment hierarchy, an engineer faces a number of problems related to the need to take into account many factors. The larger the technological scheme in terms of the amount of equipment and the more connectivities it has, the more difficult it is to allocate logically related equipment in it. The difficulty also lies in the fact that the standards do not and cannot have all the criteria for allocation. Therefore, this problem should be considered both from the point of view of the limitations and functional requirements of the ISA-88 standard and from the point of view of experience from best practices. Both can be put into the knowledge base [18].

To begin with, let us highlight clear restrictions, using which it is quite easy to determine whether the equipment belongs to one of the hierarchy levels. According to ISA-88, these levels are:

1) the level of the process cells;
2) the level of units;
3) the level of equipment modules;
4) the level of control modules.

According to the standard, “a process cell is a logical grouping of equipment that includes the equipment required for production of one or more batches”. Exactly from the point of view of batch production, the process cell is distinguished. If the entire batch of semi-products is not developed in the framework of the process cell, the equipment that is needed for this should be included. Within one process cell, there may be several connected elements of equipment capable of producing several batches in parallel. If they cannot be separated, they remain within the same process cell. In addition, the process cell must contain at least one unit.

The allocation of units is a little less obvious. There are several clear criteria:

1) one unit cannot contain several batches;
2) each technological action occurs immediately (simultaneously) with all the material within one unit;
3) the technological operation begins and ends within the same unit.

Even less obvious conditions for choosing and combining are the statements:

- the unit can include all the equipment and control modules involved in technological actions;
- the unit can work with part of the batch.

All equipment, except for the control module, can implement procedural control. That is, from the point of view of technology, it contains some procedural elements that perform a technological operation, separating itself from the method of its implementation. There are operational directives, for example, “heat to the required temperature”, as opposed to the directives “open valve 1” or “if TE101 > 23, close the valve”. The last control directive refers to equipment, not technology, and is called “basic control” in the standard. This is the main criterion that determines the principle of allocating the control module – this equipment does not contain procedural control. In addition, this part of the hierarchy enables real interaction with concrete equipment, while the other levels are more role groups. Therefore, the level of control modules cannot be omitted in the hierarchy.

The concept of procedural control per se is also not clear enough. It is difficult to formalize it as well as to define in an ontology. However, according to the standard, there are certain features inherent in it, in contrast to the basic control, such as visibility at the recipe level, a characteristic state engine, abstraction from equipment, etc.

As for the control module, there is one indirect but very useful property as a selection criterion – this type of equipment is shown in the P&ID-schemes as instrumentation. According to the standard, the control module can include other control modules, creating combined control modules.

The most uncertain criteria relate to the equipment module. Firstly, the presence of them is not necessary. Secondly, this group of equipment consists of an equipment module or other control modules, which may contain procedural control but at the same time does not meet the criteria for either a unit or even more so for a process cell. The presence of the word “may” is confusing, since the equipment module without procedural control has the same meaning as the control module. If we accept this as a strong restriction, then it is required to introduce criteria that determine how the equipment module differs from the unit. It is possible to use the criteria for belonging to the unit, and thus the most important selection criteria are as follows:

- it performs procedural control;
- it does not work with the entire batch or part of the batch at the same time.

Thus, if the technological action should take place in the flow, for example, heating/cooling in heat exchangers or batching up in the flow but within the process cell, then it should be related to the equipment module. If it is necessary to batch up the component into various units, then the batching up system is an equipment module, since it cannot belong to any unit.

All objects of the equipment hierarchy, except for the lower level, are always a group of control modules that are combined to perform a specific role. The design engineer of the control system should understand how a group of equipment can perform these roles jointly. To do this, the knowledge base that supports this should contain all the necessary knowledge about the lower level of equipment. As mentioned above, this can be done by transferring the knowledge from the P&ID-schemes to it, which are always present in the project documentation for batch production.

V. Analysis of existing approaches to the formalization of standards in the field of formulating

As already described above, there are solutions for formalizing the ISO 15926 standard based on OWL [19]. However, they have a number of disadvantages inherent in OWL-based systems [20]:

1) The need to describe metadata, in either case, leads to duplication of information. Each document should be created in two copies: marked up one for reading by humans and one for a computer;
2) An important issue is the openness and validity of the metadata used – such systems are more fragile to threats from the outside;
3) The multiformat representation of fragments of knowledge complicates the process of their processing;
4) The lack of tools for viewing and using the information provided by media resources.

The usage of the OSTIS Technology allows getting a solution without these disadvantages and with the following advantages:

1) The variety of types of knowledge stored in the system knowledge base;
2) The variety of types of questions that the system can answer;
3) The presence of a built-in intelligent help system for end-users, which provides a substantial improvement in the efficiency of the system operation;
4) The possibility of using the terminology of various natural languages;
5) Availability of comprehensive facilities for visualization of knowledge, including different styles of visualization of fragments of semantic space and convenient means of navigation through this semantic space;
6) The ability to easily extend the knowledge and skills of the system by the hands of developers;
7) System integrability with other related systems including ones built on the basis of the OSTIS Technology [21];
8) Availability of means of self-diagnosis, self-analysis and self-improvement [22].

The OSTIS Technology (an Open semantic technology for the component design of compatible computer systems controlled by knowledge) is based on a unified version of encoding and representation of information based on semantic networks with a basic set-theoretic interpretation called an SC-code and with various formats of representation of information based on it (SCg, SCs, SCn) [23]. The systems that are the target of formulating enterprises are developed on the OSTIS Technology platform.

VI. Ontological model of the ISA-5.1 standard on OSTIS

A. Content of the KB

The knowledge base according to the ISA-5.1 standard [24] describes the system of notations and symbols of tools, processes and functions, that is, it describes the lower level of control of manufacturing processes and includes a specification of the notation conventions of a toolkit.

The ISA-5.1 standard solves the problem of unification of notations and descriptions of the toolkit of manufacturing processes of various types of production including batch one. The system of notations allows describing the process and its components of production of any industry. Characters and notations are used as auxiliary means for conceptualization as well as brief and concrete means of connection between instances of various classes of the toolkit.

The knowledge base on the ground of the OSTIS Technology is based on a hierarchy of subject domains and their corresponding ontologies, which allows, on the one hand, localizing the area of solving certain problems and, on the other hand, describing the interrelations between different concepts and ensuring the inheritance of their features. Within the framework of the considered knowledge base, the hierarchy of subject domains was formed in such a way that the concepts studied in a particular domain correspond to entities that have some common function (purpose). At the top level of the hierarchy, the following set of subject domains that correspond to the ISA-5.1 standard is highlighted (fig. 4).

Figure 4. A hierarchy of subject domains of the ISA-5.1 standard

The subject domain of hardware and software, which is the key sc-element of the corresponding section of the knowledge base, which in turn is decomposed into particular subsections, describes general concepts and features that are characteristic of instruments, devices and other systems. From the point of view of the subject domain, these features are nonmaximal classes of objects of research or the relations under investigation. The degree of detail of the description of the concept depends on the problems, for the solution of which it is planned to use this information.

Each subject domain has a corresponding structural specification, which includes a list of concepts studied within this domain. Figure 5 shows the structural specification of the root subject domain – the Subject domain of hardware and software.

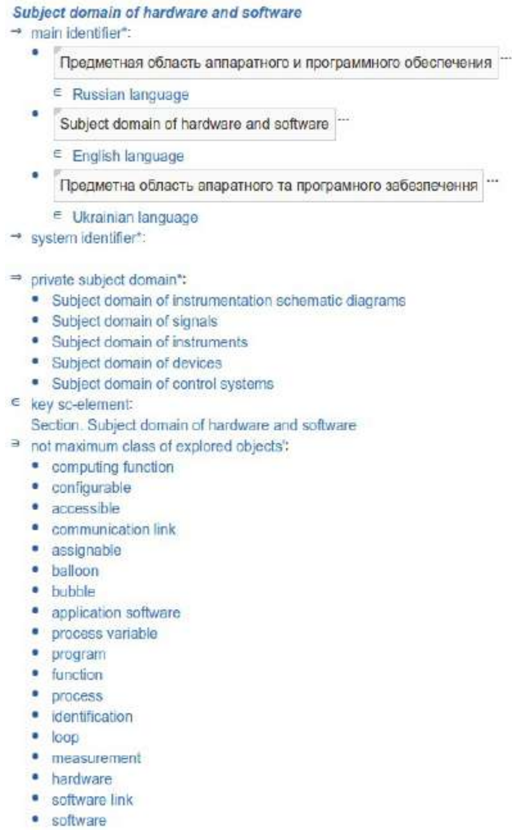


Figure 5. The specification of the subject domain of hardware and software

B. Hierarchy of subject domains

Let us consider in more detail the particular subject domains. Each type of hardware or software contained in the standard is described at the level of a certain subject domain (fig. 6).

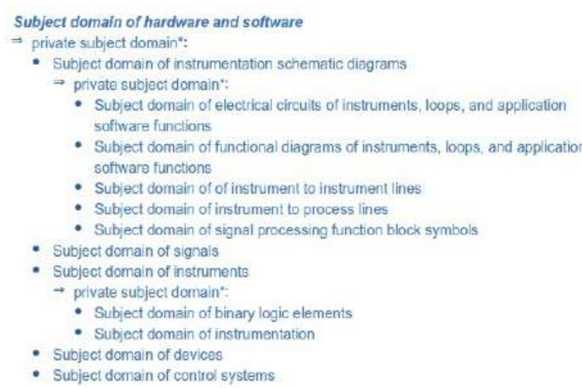


Figure 6. A hierarchy of subject domains of the ISA-5.1 knowledge base

The hierarchy at this stage is not comprehensive, it is assumed that the knowledge base will be further developed. To achieve this goal, it is necessary to increase the number of connectivities between the concepts, thereby eliminating the incompleteness and improperness of the knowledge described by the standard. Let us consider the structural specification of some of the above subject domains. They describe not only the roles of the concepts that build up them but also the relations with other subject domains.

C. Description of a particular concept, its relations with others.

The subject domain allows obtaining only that knowledge that can and could be common to the concepts contained in it. Thus, the more information about the object there is, the more clear it is to the user. Let us consider the principles of describing a specific concept and its relations with others using the example of the following system of concepts: a discrete tool (SD of instruments) $\longrightarrow$ a device (SD of devices) $\longrightarrow$ hardware (SD of hardware and software) $\longrightarrow$ a controller (SD of devices).

The specification of the concept “device” is shown in figure 7.

Figure 7. The absolute “device”

A device is the maximum class of objects of research in the Subject domain of devices. It is worth considering that the knowledge base includes the internationalization of systems of concepts necessary for the end-user of this system. In this case, an employee of a manufacturing enterprise or an engineering company may be the end-user. It is possible to map back not only the concepts of the same subject domain but also the interrelations of the subject domains themselves.

Let us consider the concept “discrete instrument” from the subject domain of instruments, which is also a subclass of the device class. It has the main identifier in three languages – Russian, English and Ukrainian – and a single system one. The “instrument” class includes entities of the “discrete instrument” class. The definition of this concept is given in a hypertext format with links to the used concepts described in the knowledge base. Different understanding of this term is incorrect, and it is inadvisable to divide it into synonyms or homonyms (fig. 8).

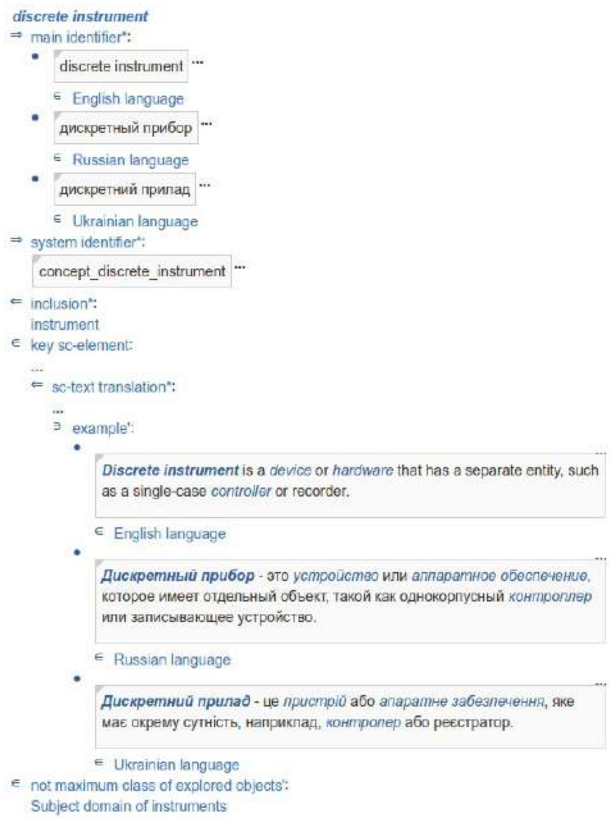


Figure 8. The absolute “discrete instrument”

Similarly, it is possible to investigate other connected concepts, for example, “software” (fig. 9), “controller” (fig. 10), etc.

Figure 9. The absolute “software”

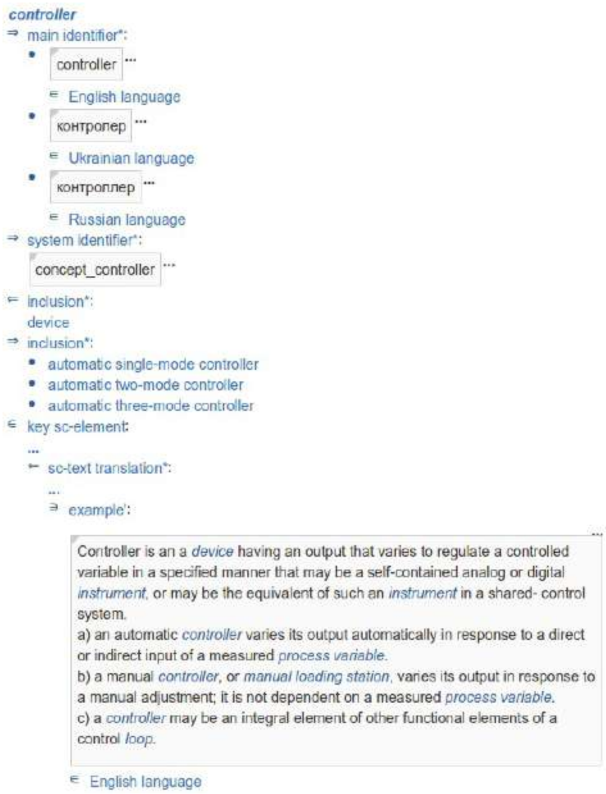


Figure 10. The absolute "controller"

An outstanding feature of the knowledge base according to the ISA-5.1 standard is the usage of logical formulas that allow describing logical conformities that characterize the features of the entities being described. Within the framework of the knowledge base according to the ISA-5.1 standard, the most interesting from this point of view are the features of binary logical elements, which are central to the subject domain of instruments and are the basis of hardware and software of formulating. To write logical statements about binary logical elements, the SCL – a sublanguage of the SC-code – was used. As an example, let us consider a qualified logical element OR, equal to "n" (fig. 11).

The KB fragment describes this concept and includes an identifier both in text format and in the form of an illustration accepted by the ISA-5.1 standard, a definition, inclusion of a binary logical element in a more general concept as well as a logical formula that describes the principle of operation of this device.

A logical formula is a structure that contains sc-variables. An atomic formula is a logical formula that does not contain logical connectives. By default, the existential quantifier is superposed on sc-variables within the framework of the atomic logical formula. Thus, the formula below means that there is a _gate entity that is a qualified logical element OR, equal to "n", which has a set of inputs of power "n", and if at least one input has

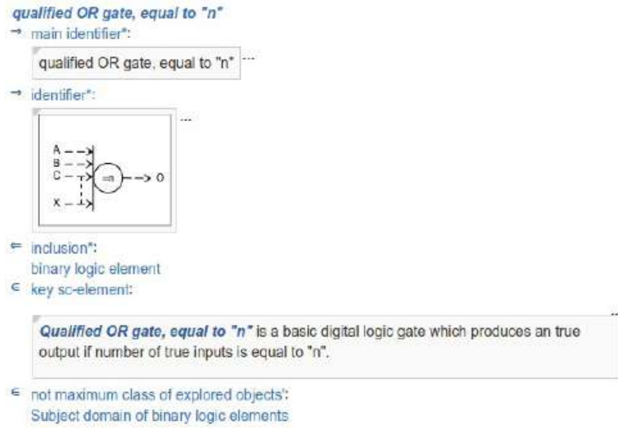


Figure 11. The semantic neighborhood of the "qualified logical element OR, equal to n" concept

the logical value "true", then the output of the formula is also "true" (fig. 12).

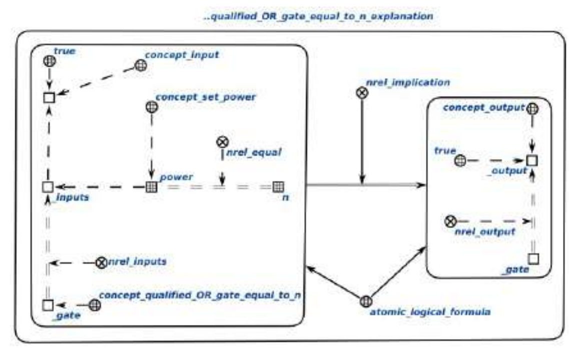


Figure 12. The logical formula of the "qualified logical element OR, equal to n" concept

VII. Examples of the operation of the system of formulating with the display of information in natural languages

The easiest way to convey information, including knowledge, to the user is a welcomed and understandable interface of the used system. For this purpose, a component was introduced into the interface of the system of formulating, which allows displaying structures written in the SC-code representation forms into natural languages familiar to the user.

As examples of the usage of the component, answers to questions in any language can serve. Figures 13 and 14 show a variant of the decomposition of the section of the SD of formulating enterprises in the SCn-editor and in natural language, respectively.

Any of the standard questions to the ostis-system can be a question in a natural language. Figures 15 and 16 show the answer to the question about the set, to which the specified concept belongs, and about the roles that it performs in this set.

Figure 13. The decomposition of the section of the SD of formulating enterprises in SCn

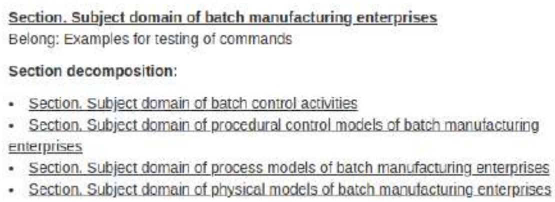


Figure 14. The decomposition of the section of the SD of formulating enterprises in the Russian language

Using this component, it is possible to represent the semantic neighborhoods of absolute and relative concepts in the knowledge base of the system. Knowledge about the concepts "unit" and "equipment phase" can have the representation forms shown in figures 17, 18 and 19, 20, respectively.

The main problem of developing this component is the need to expand the dictionary of key concepts used to make connections between fragments of neighborhoods of other concepts. The possibility of internationalization of systems of concepts causes the problem of storing and representing the used means of detecting and making such connections.

VIII. Conclusion

In this article, the principles of building a system for automating the activities of a process engineer based on an ontological approach within the framework of the Industry 4.0 concept are highlighted. The developed system includes a number of international industrial standards that are used to build a subject domain, and therefore the system can easily be combined with other ontological subject domains of the enterprise (MOM, ERP, etc.). The complex of tools and methods for developing ontology bases on the ground of the OSTIS Technology is a powerful tool for designing systems of formulating enterprises. The technology used, with many of its principles and the resulting advantages over other technologies, allows developing and multiplying the potential of existing formulating systems. At present, the

Figure 15. The answer to the question "What sets is control module an element of and what roles does it take on there?" in SCn

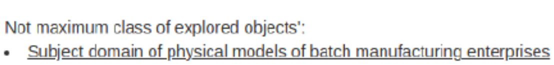


Figure 16. The answer to the question "What sets is control module an element of and what roles does it take on there?" in the Russian language

Figure 17. The absolute "unit" in SCn

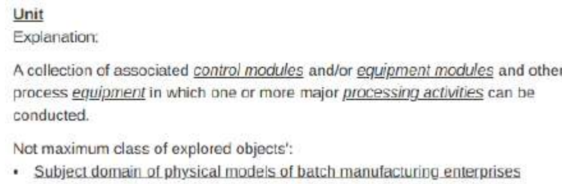


Figure 18. The absolute "unit" in the Russian language

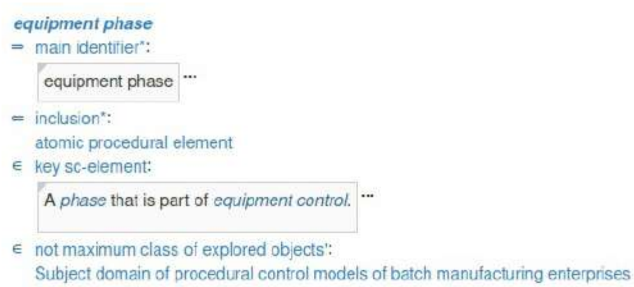


Figure 19. The absolute "equipment phase" in SCn

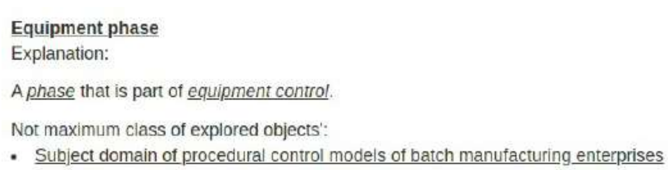


Figure 20. The absolute "equipment phase" in the English language

complex of information management systems is not just a knowledge base with a subsystem for processing user, including engineering, issues – it also has the right to be considered as a major help system of a process engineer. The general purpose of the following problems of system design is to achieve the maximum level of integration of the accumulated knowledge.

ACKNOWLEDGMENT

The authors would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics (Minsk) and the Brest State Technical University (Brest) as well as the Department of Automation and Computer Technologies of Control Systems of the National University of Food Technologies (Kyiv) for their help and valuable comments.

REFERENCES

[1] (2021) ISA5.1 standard. [Online]. Available: https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa5-1/
[2] (2021) S88 standard. [Online]. Available: https://www.isa.org/isa88/
[3] (2019) ISO 15926 Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities. [Online]. Available: https://www.iso.org/standard/68200.html
[4] (2008) GOST P/ISO 15926-1-2008 integration of life cycle data for processing enterprises, including oil and gas production enterprises. [Online]. Available: https://docs.cntd.ru/document/1200076803
[5] Taberko, V. V., Ivanyuk, D. S., Kasyanik, V. V., Golovko, V. A., Rusetsky, K. V., Shunkevich, D. V., Grakova, N. V., "Principy postroeniya sistemy kompleksnogo informacionnogo obsluzhivaniya sotrudnikov predpriyatiya recepturnogo proizvodstva," *Open semantic technologies for intelligent systems*, pp. 215–224, 2019.
[6] Taberko, V. V., Ivanyuk, D. S., Shunkevich, D. V., Pupena, A. N., "Principy povysheniya effektivnosti processov razrabotki i ispol'zovaniya standartov v ramkah koncepcii industry 4.0," *Open semantic technologies for intelligent systems*, pp. 167–174, 2020.
[7] Golenkov, V. V., Gulyakina, N. A., "Semanticheskaya tekhnologiya komponentnogo proektirovaniya sistem, upravlyaemyh znaniyami," *Open semantic technologies for intelligent systems*, pp. 42–50, 2015.
[8] (2016) GOST/ISO 22745-1-2016 – open technical dictionaries and their application to the main data. [Online]. Available: https://docs.cntd.ru/document/1200143361
[9] (2021) ISO/IEC 21838-1:2021 industrial automation systems and integration – open technical dictionaries and their application to master data. [Online]. Available: https://www.iso.org/standard/71954.html
[10] (2018) ISO 20534: 2018 Industrial automation systems and integration - Formal semantic models for the configuration of global production networks. [Online]. Available: https://www.iso.org/standard/68274.html
[11] (2004) ISO 18629 PSL: Industrial automation systems and integration (ISO TC 184 SC4) - Process Specification Language. [Online]. Available: https://www.iso.org/obp/ui/fr/#iso:std:iso:18629:-1:ed-1:en
[12] (2016) Meta Object Facility (MOF). [Online]. Available: https://www.omg.org/spec/MOF/
[13] Arp, R., Smith, B., and Spear, A.D., "Building ontologies with basic formal ontology," p. 220, august 2015. [Online]. Available: http://basic-formal-ontology.org/
[14] (2013) Zachman Enterprise Ontology (ZEO), Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), General Formal Ontology (GFO). [Online]. Available: https://www.zachman.com/resources/zblog/item/the-zachman-framework-requires-zero-documentation
[15] Niles, I., Pease, A., "Towards a standard upper ontology systems," pp. 2–9, 2001.
[16] Niles, I. and Pease, A., "Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology," pp. 2–6, 2003.
[17] A. Jordan, M. Selway, W. Mayer, G. Grossmann, and M. Stumptner, "An ontological core for conformance checking in the engineering life-cycle," pp. 358–371, 2014. [Online]. Available: https://15926.org/topics/
[18] Knowledge base for the isa-88 standard. [Online]. Available: https://github.com/savushkin-r-d/s88-ostis
[19] (2021) ISO 15926 templates and the Semantic Web . [Online]. Available: https://www.omg.org/spec/MOF/
[20] Semantic web. [Online]. Available: https://semantic-web.com/
[21] Shunkevich, D. V., "Modeli i sredstva komponentnogo proektirovaniya mashin obrabotki znanij na osnove semanticheskih setej," *Open semantic technologies for intelligent systems*, pp. 272–274, 2017.
[22] Ims metasystem. [Online]. Available: http://ims.ostis.net
[23] Golenkov, V. V., Gulyakina, N. A., Davydenko, I. T., Shunkevich, D. V., "Semanticheskaya model' predstavleniya i obrabotki baz znanij," *Data analytics and management in data-intensive fields: a collection of scientific papers of the XIX International Conference. DAMDID*, 2017. [Online]. Available: https://libeldoc.bsuir.by/handle/123456789/295039
[24] Knowledge base for the isa-5.1 standard. [Online]. Available: https://github.com/savushkin-r-d/isa-5.1-ostis

Принципы построения системы автоматизации деятельности инженера-технолога на основе онтологического подхода в рамках концепции Industry 4.0

Таберко В.В., Иванюк Д.С., Зотов Н.В.,
Орлов М.К., Пупена А.Н., Луцкая Н.Н.

В данной работе внимание уделено непрерывному развитию системы комплексного информационного обслуживания сотрудниками предприятия рецептурного производства на примере ОАО «Савушкин продукт» с использованием открытой семантической технологии проектирования интеллектуальных систем.

На примере стандартов ISA-88 и ISA-5.1 рассмотрена структура базы знаний и пользовательского интерфейса системы поддержки процессов рецептурного производства. Приведены методы для построения единой онтологической модели комплексного информационного обслуживания предприятия. В работе освещены принципы построения системы автоматизации деятельности инженера-технолога на основе онтологического подхода в рамках концепции Industry 4.0.

Received 29.08.2021

Neural network component of the product marking recognition system on the production line

Kroshchanka Aliaksandr
Brest State Technical University
Brest, Belarus
kroschenko@gmail.com

Ivaniuk Dzmitry
JSC “Savushkin product”
Brest, Belarus
id@pda.savushkin.by

***Abstract*—The paper considers the implementation of an intelligent computer vision component based on a neural network approach to solve the problem of recognizing various product markings manufactured by JSC Savushkin Product. A feature of the system is the use of modular architecture, which makes it easy to add new models. The proposed system is a component of a more general neurosymbolic system.**

***Keywords*—Computer vision, ANN, deep neural networks, IIoT, object detection**

I. Introduction

The competitiveness of any production is based on the release of various and high-quality products. Quality is the defining criterion when a client chooses a particular product, and diversity allows to cover various groups of potential buyers. Systems that automate the quality control process while maintaining the variety of products in a large enterprise are of particular value. At the same time, the quality control process is carried out not only for the product itself, but also for the packaging and marking that the buyer sees. Since the packaging forms the initial impression of the product, its quality is one of the reasons that gives the buyer a reason to buy. Marking as an element of the packaging guarantees the buyer the safety of the product for the specified period, subject to the storage conditions.

Recently, the methods of artificial intelligence in general and machine learning in particular have been widely used in industrial systems installed in enterprises to control the production process. Intelligent subsystems simplify many of the routine operations carried out to maintain the quality of the finished product. For example, control over the correctness of marking was previously carried out exclusively by a human operator. Now, with the evolution of the theory of computer vision, which has received a significant breakthrough thanks to ideas from the constantly evolving field of learning deep neural networks, systems are being created that allow this operation to be performed faster and more regularly than a human could do.

In addition to tracking markings of only one human-readable type (for example, alphanumeric), the modular implementation of such systems makes it possible to universalize the recognition process and easily add prepared models that recognize new types of markings (for example, a Data Matrix code). This special type of matrix barcode allows you to encode special identification information, as well as weight, expiration date, serial number, consignment and date of manufacture of the product [1].

In addition to universality in the processing of products of various types, automation of control in production is of particular importance. With the evolution of the concept of Industry 4.0 and IIoT (Industrial internet of things), more and more intelligent control solutions are being created. The implementation of such solutions allows full control over production, including control over the condition of production equipment and safety measures. Moreover, these important functions are carried out without human intervention. In the future, this can lead to the creation of completely autonomous production facilities, in which even the operations of planning the purchase of raw materials and the shipment of finished products will be carried out automatically.

This work is devoted to the development of a neural network component that is part of a more general neurosymbolic system described in [2].

II. Problem formulation

The task set before us was to develop a system for verifying the correctness of marking on products manufactured by JSC Savushkin Product. An important aspect is that the system must work in real time, based on data from a camera installed above the production line. This camera generates a video stream at 76 frames per second.

In addition to the alphanumeric code (Fig. 1), since recently, products can be produced with alternative marking including a Data Matrix code (Fig. 2) [3]. This type of marking is a convenient and concise representation of the specific and basic data about the product.

Figure 1. Product with alphanumeric marking

Figure 2. Product with Data Matrix code

Based on the general formulation of the problem, the following subtasks can be distinguished, which must be solved by the system:

1) detection and recognition of the type of marking;
2) marking recognition;
3) identifying possible marking problems.

In general, the following requirements can be imposed on the system that will solve the assigned tasks:

- **High speed of work**. The production line is moving very quickly, so the detection of defects should be carried out with minimal delays;
- **Autonomy**. The system should minimize operator involvement in the quality control process;
- **Universality**. The system should be configured to recognize the marking of any product;
- **Adaptability**. The system must work under any conditions that occur in production (for example, insufficient lighting, personnel errors, etc.).

Let's list the main problems with marking [2]:

1) **lack of ink:** in case of receipt of products on the production line without marking, the system must conclude that there is no ink;
2) **camera shift:** if no data on the recognition results are received from the neural network modules, but the system knows that movement along the production line has begun, then it must conclude that the camera has shifted;
3) **bad marking:** marking was found and recognized, but did not match the template representation. In this case, the system must conclude that the marking is incorrect;
4) **unreadable marking:** if the marking is blurry and cannot be recognized, it is necessary to stop the production line and report the error to the operator.

For problems 1,3 and 4, it is necessary to screen out products that have a problem marking. The occurrence of these problems implies a complete stop of the production line movement and reporting to the operator about the problem.

III. Overview of existing approaches

Despite the existing interest in the autonomy of production processes and the indisputable advantages that its implementation entails, tasks similar to those described are solved in a large number of cases with the participation of a person. The operator simply checks a part of the production periodically and randomly. This approach has disadvantages:

- only a small part of the production passes inspection, so there is a possibility that the defective marking will be missed;
- the speed of a person's reaction to an emergency situation may be insufficient;
- a person may not notice a small difference between the checked marking and the template one;
- the manual verification work is monotonous.

Existing projects are based on hardware solutions, for example, on the use of special sensors [4].

Such solutions implement marking recognition, but have a number of important disadvantages:

- Unstable recognition quality, depending on the conditions under which the recognition is performed. Since the production line moves quickly, the necessary conditions for high-quality recognition are often not met;
- Necessity to purchase specialized software to configure sensors.

Thus, such solutions create additional difficulties in operation, which are manifested in the need, in addition to selective manual control of product quality, to control the recognition system itself.

IV. Proposed approach

The proposed approach is to use a pipeline structure of separate neural network modules, each of which solves its own subproblem of marking recognition.

The task of this pipeline is to detect the marking, determine its type and recognize it.

Let us stop on the system architecture in more detail.

A. Formulation of the marking detection problem

As mentioned earlier, in the process of recognizing product markings, additional tasks are solved related to the correct application of such markings. The first task is to determine the presence of marking on the product and the type of marking. The second task is to determine the presence of marking distortions that arise during the printing process, the absence of its parts, etc. And finally, the third task is the actual recognition of the marking, which is carried out in different ways depending on its type (for example, by detecting individual digits that make up the marking, or by recognizing the corresponding Data Matrix code). After completing these tasks, the output information is generated (the date of production of the product, the number in the consignment, etc.) and the correctness of these data is determined in accordance with a predetermined template.

However, it should be noted that in the process of solving these problems, the following problems arise:

- **High speed video stream**. Since the video stream speed is 76 frames per second, the processing time for each frame is about 13 milliseconds. It should be noted that this time is not enough to launch a complex neural network architecture.
- **Impossibility of correct direct detection of digits**. In addition to the digits contained directly in the marking, digits applied to other objects can get into the frame, for example, the production line itself or its parts. In addition, it should be noted that the image enters the neural network with a reduced resolution, which leads to the difficulty of recognizing small objects (digits).
- **Possibility to change the orientation of marking**. During the movement of products along the production line, it is possible to rotate the marking at an arbitrary angle, which leads to a significant degradation in recognition.

The listed problems can be solved architecturally. So, the first of these problems is solved by skipping insignificant frames in which the product is not in the middle of the frame. This allows us to increase the time interval required for image processing by the neural network. Thus, it is necessary to assess the significance of the frame. This can be done with a simple model with a short processing time.

The second problem is solved by decomposing the detection problem into separate subtasks. For example, at the beginning, the product is found, then the marking on the product and, finally, separate digits.

Finally, the third problem is solved by estimating the angle of rotation of the marking in order to present it in the maximum horizontal position.

The presence of several subtasks involves the use of a group of models, each of which does its own part of the work.

B. Recognition module architecture

Based on the previously formulated tasks and problems solved by the recognition module, we list the main subtasks.

1) Assessment of product position
2) Product and marking detection
3) Defining the type of marking
4) Rotate markings for horizontal orientation
5) Marking recognition
6) Assembling the marking and checking it

It should be noted that in our implementation, almost every subtask uses its own neural network model. This allows you to easily modify the system, improve individual modules, change them, and add new ones.

The architecture of the ready-made recognition module is shown in Fig. 3.

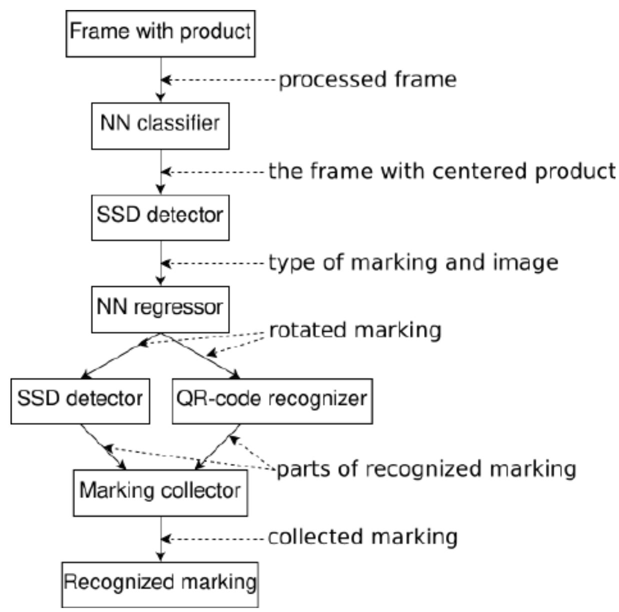


Figure 3. Structure of the marking recognition module

Let us briefly describe the purpose of each individual neural network model and their role in the overall architecture.

The first model is a convolutional classifier that determines the significance of the frame for subsequent analysis. In this case, the most significant is the frame in which the product is closest to the center of the frame. Four main classes of positions were defined as the distance of the product from the center of the frame. Class 1 describes the minimum distance. Only frames of this class are involved in the subsequent stages of processing and analysis. Classes 2 and 3 describe the average and maximum distance. Class 4 is used when there is no product in the frame (empty line).

The architecture of the classifier used is shown in Fig. 4. It consists of 5 layers and has 4 output neurons according to the number of classes that determine the position of the product in the frame. All layers use the ReLU activation function except for the 3rd and last layers. They use linear and softmax activation functions, respectively. Also applies max pooling after the first and second convolutional layers with stride equal to 2.

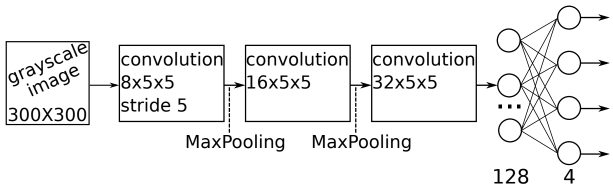


Figure 4. Classifier structure for bottle position estimation [2]

If the frame has been assigned to class 1, it is passed on to the next model.

The second model is a detector and searches for products and markings in the frame. Here, an SSD network [5] based on the MobileNet v1 [6] classifier was chosen as the architecture.

Independent product and marking detection automatically identifies the missing label situation. To do this, it is enough to check the logical condition for the absence of marking in the presence of the product itself in the frame. If a defect is detected, the system notifies the operator about it.

It should be noted that this detector is used to detect different types of markings. Since the SSD model can be used to detect objects of different classes, it was decided to use it to determine the type of markings, which does not require any special analysis procedures.

As a result, if a defect in the marking was not detected, the image of the marking (in its original size) and information about its type are transferred to the next model.

The third model is a regressor used to estimate the angle of rotation of the marking. This model returns the angle by which the marking must be rotated to achieve a horizontal orientation of the image. This conversion improves the quality of subsequent digit recognition. After rotation, the image of the marking is transferred to the model for analyzing the corresponding type of marking (in our implementation, these are models for analyzing a Data Matrix code or alphanumeric marking).

To analyze the alphanumeric marking, the SSD-MobileNet v1 detector is also used, which detects separate digits in the marking. The current implementation uses a non-neural network model to analyze the Data Matrix code. The applicability of a neural network for such an analysis may be the subject of further research.

Further, the assembly of the recognized marking is carried out and the recognition result is returned.

The principle of operation of the neural network component using the example of alphanumeric marking is shown in Fig. 5.

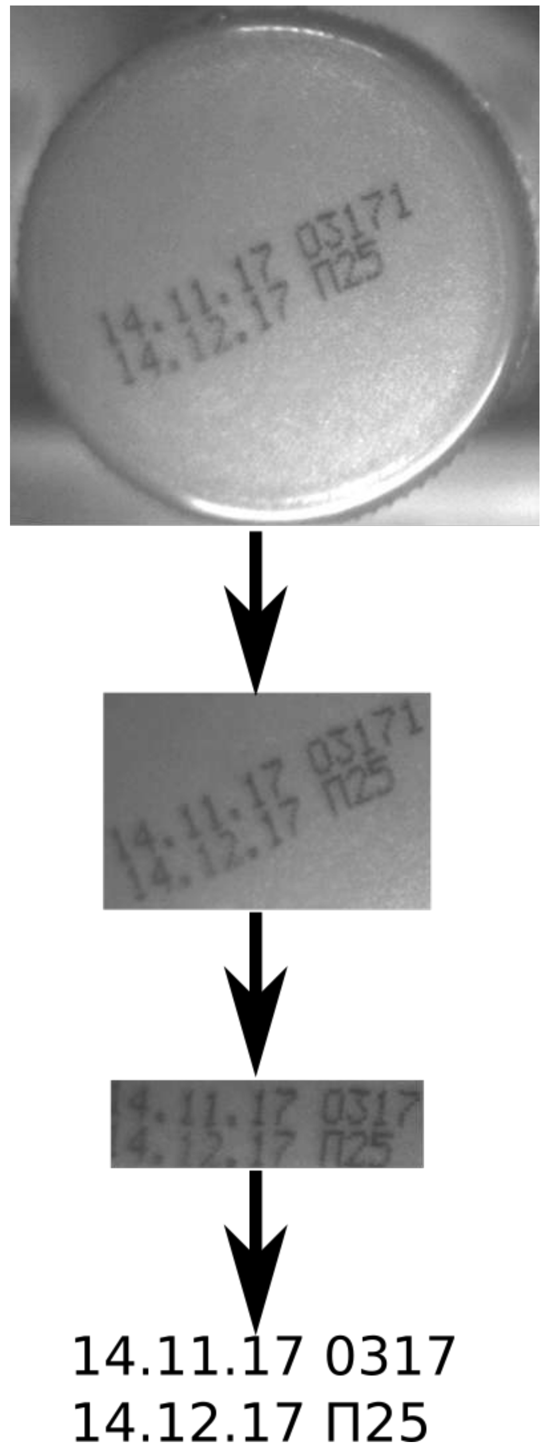


Figure 5. How the system works

C. Training datasets

In the process of preparing neural network models, we used several different training datasets:

- Dataset for training the classifier
- Dataset for training the detector of markings and products (determining the type of marking)
- Dataset for training the rotation angle regressor
- Dataset for training the digit detector

Dataset for training the classifier. To create a dataset, the Faster R-CNN [7] model was used (based on the pretrained ResNet50 [8] classifier). This model has better performance indicators than SSD-MobileNet, but is inferior in speed. It was used to automatically labeling the available data (mainly video files of the production process) according to the distance of the product from the center of the frame. The metric used was the Euclidean distance from the center of the product to the center of the frame. Thus, four classes of images were formed, which were used for the subsequent training of the convolutional classifier. The total sample size was 6189 images, 1303 of which constituted the test dataset.

Dataset for training the detector of markings and products (determining the type of marking). To form this dataset, we used manually labeled images from the classifier training dataset and additional images of generated Data Matrix codes. The total dataset size was 815 images, 163 of which constituted the test dataset.

Dataset for training the rotation angle regressor. When creating this dataset, the images of markings rotated at arbitrary angles were used. The total dataset size was 59385 images, of which 11877 constituted the test dataset. The images of markings obtained from the dataset for training the detector of markings and products were taken as a basis.

Dataset for training the digit detector. To create this selection, a dataset of house numbers SVHN [9] was used, as well as labeled markings. The variant of the SVHN dataset was used, which included 33402 images in the training part and 13068 in the test part. The dataset size of labeled markings was 419 images.

D. Results

After training classifier, the final recognition accuracy was 93.27%.

Both detectors (products/markings and separate digits) were trained on the basis of pre-trained models.

The use of the SSD model allows achieving detection efficiency of 99% (mAP = 0.99) for product detection and marking and 92% (mAP = 0.92) for separate digits. In addition, the processing speed makes it possible to successfully detect markings in the video stream at 76 frames per second. The results of the recognition efficiency of separate digits are presented in the table I.

Table I
EFFICIENCY OF DETECTING INDIVIDUAL CLASSES OF DIGITS

Class label	AP
0	0.9218
1.	0.9107
2.	0.9354
3.	0.9286
4.	0.9265
5.	0.9137
6.	0.9274
7.	0.9167
8.	0.9646
9.	0.8975
mAP	0.92429

The results for the detector of markings and the detector of separate digits are shown in Fig. 6 and 7.

Figure 6. Detected product and marking

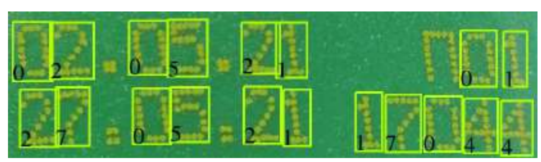


Figure 7. Detected digits in marking

CONCLUSION

This paper discusses the development of an intelligent component for recognizing product marking based on a neural network approach. The advantage of the proposed solution is the use of the neural networks pipeline, which allows easy switching between individual models. It should be noted the universality of the proposed approach, which is manifested in the simplicity of performing the work on recognizing arbitrary markings on a variety of products. It is enough to integrate a new module with the required functionality and the system will start using it. The models we use are efficient, which makes it possible to operate the system on a fast moving production line.

The direction of further work can be chosen to improve the existing results of recognition of alphanumeric markings, as well as to study the application of a neural network approach for QR code recognition.

Acknowledgment

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help and valuable comments. In particular, Vladimir Adamovich Golovko, Vladimir Vasilyevich Golenkov, Daniil Vyacheslavovich Shunkevich and Mikhail Vladimirovich Kovalev.

References

[1] GS1 DataMatrix – Electronic sign. Available at: https://datamark.by/voprosy-otvety/gs1-datamatrix (accessed 2021, Jun) (In Russian)

[2] V. Golovko, A. Kroshchanka, M. Kovalev, V. Taberko, and D. Ivaniuk "Neuro-Symbolic Artificial Intelligence: Application for Control the Quality of Product Labeling," Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2020, pp. 81-101.

[3] Milk products – Electronic sign [Electronic resource]. Access mode: https://datamark.by/markirovka-molochnoy-produktsii (accessed 2021, Jun) (In Russian)

[4] Omron documentation. Available at: https://robotics.omron.com/browse-documents. (accessed 2021, Jun)

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. Berg "SSD: Single Shot MultiBox Detector," in arXiv, arXiv:1512.02325v5 [cs.CV], Available at: https://arxiv.org/abs/1512.02325 (accessed 2021, Jun).

[6] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in arXiv, arXiv:1704.04861v1 [cs.CV]. Available at: https://arxiv.org/abs/1704.04861 (accessed 2021, Jun)

[7] S. Ren, K. He, R. Girshick, and J. Sun "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in arXiv, arXiv:1506.01497v3 [cs.CV]. Available at: https://arxiv.org/abs/1506.01497 (accessed 2021, Jun)

[8] K. He, X. Zhang, S. Ren, and J. Sun "Deep Residual Learning for Image Recognition," in arXiv, arXiv:1512.03385v1 [cs.CV]. Available at: https://arxiv.org/abs/1512.03385 (accessed 2021, Jun)

[9] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng "Reading Digits in Natural Images with Unsupervised Feature Learning," NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.

Нейросетевой компонент системы распознавания маркировки продукции на производственной линии

Крощенко А. А., Иванюк Д. С.

Данная работа посвящена аспектам разработки нейросетевого компонента системы распознавания маркировки на производственном предприятии ОАО "Савушкин Продукт". Данная система предназначена для эксплуатации с различными типами маркировок (буквенно-цифровая, Data Matrix) и может быть легко дополнена новыми типами по мере необходимости. Каждый из модулей предложенной системы заменяем и может независимо изменяться и улучшаться. Нейросетевые модели, используемые в текущей реализации системы, отличаются быстротой работы и могут быть с успехом использованы на производственной линии с высокой скоростью конвейера.

Received 09.07.2021

Principles and possible ways of building an intelligent system of integral medicine

Rostovtsev V.N.
RSPC of Medical Technologies of the Ministry of Health of the Republic of Belarus
Minsk, Belarus

Kobrinskii B.A.
Federal Research Center "Informatics and Management" of the Russian Academy of Sciences
Moscow, Russia

***Abstract*—The conceptual framework of an intelligent system of integral medicine in the medical-technological basis of functional spectral-dynamic diagnostics is considered and the ways of its building are determined.**

***Keywords*—intelligent system, diagnostics, integral medicine**

I. Introduction

Under the concept of integral medicine, let us understand the model of iatric thinking based on the integration of knowledge from the main methodic approaches of world medical science and practice.

This is, firstly, the convergence of various fields: oriental (traditional Chinese, Indian and Tibetan medicine), allopathic, homeopathic, naturopathic (the usage of natural pharmaceuticals, including bioactive accessory food substances), physiopathic (isopathic, frequency-resonance and spectral-dynamic methods) ones.

In addition, integral iatric thinking necessarily includes two more dimensions. The second dimension is the integration of knowledge of the health-improving (health-protective), preventive and curative branches of medicine. And the third dimension is the integration of knowledge about the main etiological levels of pathology, including genetic, morphological, biochemical, functional and mental ones.

II. Medical-technological basis

For the next generation of physicians to be able to successfully complete their tasks in the paradigm of integral medicine, a single medical-technological basis is required. Today, spectral-dynamic technology claims to be the single medical-technological basis of integral medicine [1]. This is determined not only by the technological advantages (simplicity, rapidity and passivity of the diagnostic procedure) but also by the possibilities of practically unlimited spreading of the spectrum of diagnostic markers and the creation of problem-oriented marker bases that facilitate the work of a physician and are a necessary condition for the building of automated systems for completing various tasks based on new algorithms developed by physicians.

The development of the theory, methods and technological means of integral medicine is a new large-scale field in the theory and practice of medicine. It is important to clarify that this process preserves the existing structure of medical activities. In other words, the question is not about all physicians becoming specialists in integral medicine, but about the fact that physicians of various specialties will gain access to its methods that can radically improve the quality of health-improving, preventive and curative care.

In practice, the methods of integral medicine can become available to physicians only if there is an intelligent system that provides full-scale usage of the capabilities of integral medicine.

Currently, the main obstacle in the way of building an intelligent system of integral medicine is the differentiation of conceptual constructs of different fields of medicine. The overcoming of this is possible only on the way of ensuring semantic compatibility and semantic convergence of subsystems in the process of designing intelligent systems [2]– [4].

III. Key terms

Before identifying a means of overcoming these obstacles, let us consider the correlation of the concepts "intelligence", "task" and "diagnostics".

The tools of thinking are systems of concepts and systems of images. The imaginative and conceptual constructs contained in the memory of the subject of thinking are the corresponding systems of their knowledge. The world cognition by a child begins through images, which are then joined by concepts.

Images and understanding of problems can be intuitive, that is, the understanding of problems is not carried out at the level of argumentation, and the resulting formulations may be unprecise. In this case, the communicative moment necessary for the transfer of knowledge obtained at an intuitive level, using associative relations, may be difficult. Thus, it is possible to talk about two ways or two levels of thinking – intuitive and analytical ones, the last of which includes the structuring of concepts and the nature of their connections.

In the humanities and natural sciences, which belong to poorly structured areas of knowledge, decision-making at the initial stage is often not only, and sometimes not so much, argumentative as intuitive. Intuition and argumentation as a type of reasoning are two sides of the same phenomenon. Since the knowledge accumulated in the form of experience is often intuitive and cannot be described [5]. According to A. Bergson [6], "intelligence is carved in intuition", and intuition, according to K. G. Jung [7], is a psychological function that provides the transfer of perception to the subject in an unconscious way. At the same time, intuitive thinking is "fueled" by imaginative representations. The so-called "internal images" of objects, phenomena, situations, that exist as a stimulus, in the absence of a directly affecting prototype, can act as generalizing ones for imaginative phenomena that correspond to a certain situation [8], [9]. In medicine, intuition helps to search for diagnostically significant traits or their combinations (syndromes) and can contribute to the choice of a rational way of the patient examination, which optimizes the diagnostic process that includes mental intuitive-imaginative hypotheses [5].

Intuitive thinking, which is present in humans, has not yet been implemented in technical systems. Although sometimes image recognition using neural network technologies is called intuitive recognition, it is actually the result of computational procedures on numerous neurons of multilayer perceptrons (Deep Learning). However, fuzzy logical-linguistic (linguological) models [10], which should include intuitive-imaginative-logical systems [5], can be implemented using fuzzy inference algorithms. As a basis for this, imageries can serve, accompanied by factors of confidence of experts, which, in turn, can include intuitive representations of experts. This allows transforming a fuzzy imagery into a relatively ordered one – from a typical or classical image (archetype) to the least similar ones, which are especially difficult to identify [11].

Analytical thinking is present in biological systems and is implemented (modeled) in intelligent technical systems. The communicative aspects of "thinking" can be traced in multi-agent systems and systems that use ant colony optimization [12]– [15]. To formalize mental concepts, it was proposed to use extensions of modal logics [16], [17]. This approach is based on the syntax of the variant of the multi-modal logic of multiplicative time proposed by the authors and on semantics in the form of a set of possible worlds. The result is the formalization of the dynamics of the environment and the behavior of an intelligent agent.

The intuitive and analytical levels of thinking are a means of intellectual support for the operation of a biological or technical system in a specific environment and in the context of a certain value system, explicit or not explicit one. The operational environment of the system can have several levels, which is especially noticeable when referring to manipulators and robots. There are two main levels in the biological system. The first level is the sensory environment, the second level is the thinking environment, which is divided into visual active thinking (innate one, that refers to the subject area), concrete-objective thinking (completion of tasks concerning a real object), abstract-logical thinking, which allows completing creative tasks.

In biological systems, communications are mediated by sensory systems, including, mainly, the visual, auditory and tactile systems of a human, and in cyber physical systems, it is electricity-measuring, low-frequency electromagnetic, infrared, optical, X-ray, biochemical and other sensors.

Intelligent software is, in fact, an analytical tool for a biological or technical system. From this, the following definition can be formulated:

INTELLIGENCE is a tool for cognition, understanding, solving problems and completing tasks.

It should be noted that the fundamentally important characteristics of intelligence are the ability to recognize the essential in the data, the generation of a sequence (purpose – plan – action), the ability to select assumptions relevant to the purpose, reasoning (obtaining consequences from assumptions), decision-making through argumentation [18].

The completion of the task is always preceded by its understanding, which can be implicit (intuitive) or explicit (analytical and communicative), including fixed in the architecture or in the algorithms of the technical system. Explicit understanding can and should be based on both declarative and procedural knowledge. At the same time, finally, a human, including the creator of a technical system, has an understanding formed on the basis of a conceptual construct. The consequence of this fact is the high value of a tight definition of concepts. For example, for the understanding of tasks, the tight definition of the concept of the task itself, given by Yu. L. Yershov and K. F. Samokhvalov, effectively helps [19].

A TASK is something, for which a criterion for distinguishing a completion from a non-completion is defined. Note that, in contrast to the definition of the concept of a task, a PROBLEM is something, for which a criterion for distinguishing a solution from a non–solution is not defined. The definition of such a criterion translates the problem into the rank of a task. The quality (accuracy) of the found criterion for distinguishing a solution/completion from a non-solution/non-completion depends on the level of understanding of the problem.

The understanding of the concept, idea, problem, task, message, event, phenomenon, system, etc. (conditionally generalized, object) is carried out in the actual discourse as a set of relevant, that is, having a sufficiently high rank in the value system, semantic fields of object classes. That is, the actual discourse (late Lat. discursus – reasoning, argument) forms a system of understanding of the object.

There are three levels of understanding of the object.

The first level of understanding can be considered as the identification of the belonging of an object to the semantic fields of a set of object classes that are suitable by association.

The second level of understanding is the integration (consolidation) of the obtained as a result of identification of belonging, that is, semantic identifications, into an integral whole in the current discourse. However, it should be noted that the subjective irrelevance of some objectively significant semantic field of the general discourse leads to an incomplete or incorrect and even false understanding of the object.

The third level of understanding is the identification (including the formulation) of the meaning of the essence of an object as an invariant of its integral understanding in a certain set of contexts. The limiting case of the third level of understanding is a tight definition of the meaning or essence of an object. Thus, the definition of the concept of a task according to [19] led to the need to revise some of the foundations of mathematics. This suggests that the natural language conceptual base is always primary, and the subsequent levels of formalization using artificial languages, including the language of mathematics, are secondary.

In general, the understanding of "what it really is" is possible only in the discourse of the laws of human nature, including the laws and determinants of health. This statement is correct in relation to almost all objects, with a few exceptions of purely technical, restricted professional issues. It is obvious that a sufficiently complete and noncontradictory understanding of tasks and problems by natural or artificial intelligence is a necessary condition for their correct solution and completion.

The tasks of recognition and classification in poorly structured areas of knowledge in artificial intelligence systems are completed using heuristic and logical algorithms. Next, let us focus on the class of diagnostic tasks.

DIAGNOSTICS is the identification of the belonging of an object or process to a certain semantic category. It is based on sensory, imaginative, computational and verbal data (exclusively verbal data can be found in the diagnostics of mental diseases in medicine, the comparison of images and their coordinates in astronomy).

As a semantic category, in medicine, the positions of nosological classification (classification of diseases) or disease exposure (prenosological diagnostics) are most often used. In technical diagnostics, this can be, for example, a classification of failure risk levels.

As sources of sensory data for medical diagnostics, equipment based on the fixation of electrical measurements, low-frequency electromagnetic waves, infrared or X-ray radiation and data from other sensors can serve.

Regardless of the source of the sensor signals, the input data for diagnostic systems is usually a parametric vector or matrix, in particular, an image. In both cases, neural network, logical and logical-statistical recognition algorithms, as well as their combinations, are applicable.

IV. Prospects of intelligent diagnostic systems

Medicine is in urgent need of the development of intelligent diagnostic systems (IDS). The history of existing IDS has convincingly demonstrated their high efficiency and low potency of widespread usage. A similar situation occurs with new IDS developments. This contradiction is caused not only by organizational and financial issues but also by the complexity of integrating intelligent decision support systems into medical information systems due to technological incompatibility, the main component of which is the problem of semantic convergence.

The real possibility of overcoming this contradiction is provided by the principles of the organization of the Open Semantic Technology for Intelligent Systems (OSTIS) [20]. On this basis, the prospects of ontological design, production and operation of semantically compatible hybrid intelligent computer systems are being opened.

The prospect of using the OSTIS Technology to create medical IDS can be seen in the following example.

Over the past two decades, the technology of functional spectral-dynamic diagnostics (FSD-diagnostics) has been developed using a medical spectral-dynamic complex [1]. This technology is based on recording a low-frequency electromagnetic signal from the patient's skin surface using a passive electrode (FSD-sensor) and allows simultaneously obtaining diagnostic information on all body systems in the amount of tens of thousands of diagnostic characteristics. The most valuable quality of FSD-diagnostics is the possibility of early detection of infectious and common noncontagious diseases [21].

The mass character of early detection of diseases will be provided by the technology of automatic FSD-smart-diagnostics [22]. This technology will make it possible to use a personal smartphone (or other means of Internet communication) for automatic FSD-diagnostics of infectious and chronic noncontagious diseases at the early stages of their development for individual prevention and early treatment. The set of essential interacting automatic diagnostic systems requires their semantic compatibility, and therefore the OSTIS Technology best corresponds to the tasks of implementing smart diagnostics [23]. And most importantly, that FSD-diagnostics uses markers that take into account the features of the diagnostics of the main fields of medicine, including allopathic, naturopathic, isopathic and homeopathic ones.

V. Conclusion

The above considerations allow assuming that the combination of FSD-diagnostics technology (in combination with other sources of diagnostic data) and the OSTIS Technology can serve as an effective basis not only for creating IDS for a wide range of common infectious and noncontagious diseases but also for building an intelligent system of integral medicine.

References

[1] Medical spectral-dynamic complex [Electronic resource] Access mode: http://www.kmsd.su. – Access date: 09.09.2009.

[2] Jung, J., Choi, I., Song, M. An integration architecture for knowledge management systems and business process management systems // Computers in Industry. 2007. Vol.58. Pp. 21–34.

[3] Golenkov V. V., Gulyakina N. A. Semantic technology of component design of systems controlled by knowledge / / Open semantic technologies for intelligent systems. 2015. No. 5. pp. 57–78. (in Russian)

[4] Kobrinskii B.A., Yankovskaya A.E. Convergence of Applied Intelligent Systems with Cognitive Component // Open Semantic Technologies for Intelligent System. OSTIS 2020 / Golenkov V., Krasnoproshin V., Golovko V., Azarov E. (eds) // Communications in Computer and Information Science, vol 1282. Springer, Cham, 2020. P. 34–47.

[5] Kobrinskii B.A. Avtomatizaciya razrabotki pol'zovatel'skih interfeysov s dinamicheskimi dannymi // VI National Conference on Artificial Intelligence with international participation: Conf. Vol. 1. – Pushchino, 1998. pp. 7–14.

[6] Bergson A. Creative evolution. – M., "CANON-press", "Kuchkovo pole", 1998. – 384 p.

[7] Jung K. Psychological types. – St. Petersburg, Moscow: Juventa, Progress-Univers, 1995. – 716 p.

[8] Ananyev B. G. Psihologiya chuvstvennogo poznaniya. – Moscow: Nauka, 2001. – 279 p.

[9] Gostev A. A. Psihologiya vtorichnogo obraza. – Moscow: Institute of Psychology of the Russian Academy of Sciences, 2007. – 512 p.

[10] Pospelov D. A. Logiko-lingvisticheskie modeli v sistemah upravleniya. – M.: Energoatomizdat, 1981.

[11] Kobrinskii B. A. Sub'ektivnost' i ob'ektivnost' v mire mediciny: Vzglyad s pozicii predstavleniya znaniy dlya gibridnoy intellektual'noy sistemy / / "Integrated models and soft computing in artificial intelligence". Collection of scientific w. of the X-th International Scientific and Practical Conference (IMSC-2021, Kolomna, May 17-20, 2021). In 2 volumes. Vol. 1. Smolensk: Universum, 2021. pp. 55–67.

[12] Tarasov V. B. Ot mnogoagentnyh sistem k intellektual'nym organizaciyam: filosofiya, psihologiya, informatika. – Moscow: Editorial URSS, 2002. – 352 p.

[13] Gorodetskii V. I. Samoorganizaciya i mnogoagentnye sistemy. I. Modeli mnogoagentnoy samoorganizacii / / Izvestiya RAS. Theory and control systems. 2012a, No. 2, Pp. 92–120.

[14] Gorodetskii V. I. Samoorganizaciya i mnogoagentnye sistemy. II. Prilozheniya i tekhnologiya razrabotki // Izvestiya RAS. Theory and control systems. 2012b. № 3, Pp. 102–123.

[15] Iskusstvennye immunnye sistemy i ih primenenie / Edited by D. Dasgupta. Translated from English – Moscow: FIZMATLIT, 2006. – 344 p.

[16] Rao A.S., Georgeff M.P. Formal model and Decision Procedures for Multi-Agent Systems. Technical note 61, Australian Artificial Intelligence Institute, 1995a. – 54 p.

[17] Rao A.S., Georgeff M.P. BDI Agents: From Theory to Practice // Proceedings First International Conference on Multi-Agent Systems (ed. V. Lesser). AAAI Press/The MIT Press. 1995b. Pp. 312–319.

[18] Finn V. K. Intellekt, informacionnoe obshchestvo, gumanitarnoe znanie i obrazovanie: Ponyatijnye i logicheskie osnovaniya iskusstvennogo intellekta. Gumanitarnoe znanie i kognitivnye issledovaniya v informacionnom obshchestve. Obrazovatel'nye programmy dlya iskusstvennogo intellekta. – M.: LENAND, 2021. – 464 p.

[19] Yershov Yu. L., Samokhvalov K. F. O novom podhode k metodologii matematiki / Yu. L. Yershov, K. F. Samokhvalov // In the book.: Zakonomernosti razvitiya sovremennoj matematiki. Metodologicheskie aspekty. M.: Nauka, 1987. – P. 85 – 106.

[20] Golenkov V. V., Gulyakina N. A. Principy postroeniya massovoj semanticheskoj tekhnologii komponentnogo proektirovaniya intellektual'nyh sistem // Open semantic technologies for intelligent systems (OSTIS-2011): materials of the International Scientific and Technical conf. Minsk, February 10–12, 2011). – Minsk: BSUIR, 2011. pp. 21–58.

[21] Rostovtsev, V. N. Reshenie problemy rannej diagnostiki /V. N. Rostovtsev // General Practitioner's Handbook. Journal of Research and Practice. – 2016. – №04.– P. 10–15.

[22] Rostovtsev, V. N. Tekhnologicheskaya koncepciya avtomaticheskoj smart-diagnostiki / V. N. Rostovtsev // Issues of organization and informatization of healthcare. – 2019. – № 4 (101). P. 57–59.

[23] Rostovtsev, V. N. Principles of ostis-system of automatic diagnosis design / V.N. Rostovtsev, O.S. Rodionova // Open Semantic Technologies for Intelligent Systems OSTIS–2018. – 2018. – P. 341–346.

Принципы и возможные пути построения интеллектуальной системы интегральной медицины

Ростовцев В.Н., Кобринский Б.А.

В работе рассмотрены концептуальные основы и определены пути построения интеллектуальной системы интегральной медицины в медико-технологическом базисе функциональной спектрально-динамической диагностики.

Изложенные в работе соображения позволяют считать, что сочетание технологии ФСД-диагностики (в сочетании с иными источниками диагностических данных) и технологии OSTIS может служить эффективной основой не только для создания ИСД для широкого спектра распространенных инфекционных и неинфекционных заболеваний, но и для построения интеллектуальной системы интегральной медицины.

Received 07.06.2021

Decision support system for breast cancer screening

Kayeshko Aliaxey
Intellectual Information Technologies
BSUIR
Minsk, Belarus
ondister@gmail.com

Efimova Aleksandra
Intellectual Information Technologies
BSUIR
Minsk, Belarus
sashaefimova007@gmail.com

***Abstract*—Currently, decision support systems in radiation mammology focus on the detection and classification of neoplasms, despite the fact that the real work of a radiologist does not imply a diagnosing. Computer vision systems use a black box model and do not explain the results of work, which is unacceptable in medicine.**

The proposed hybrid approach, combining the work of computer vision subsystems and a problem solver based on graphodynamic sc-memory, can make a structured research report based on the detected anomalies and production output. This approach will reduce the time of the doctor-radiologist.

Also, for the first time, the decision support system of an X-ray technician is presented to improve the quality of breast styling. The system will improve the quality of the X-ray image and, accordingly, improve the evaluation of the study.

***Keywords*—Dicom, Ostis, Cad, Digital Mammography, Birads**

I. Introduction

Breast cancer is one of the most frequently diagnosed malignancies among women. So, in the United States in 2018, its share among all malignancies detected in women was about 30% [1]. Currently, mammographic screening remains the most effective method of early detection of breast cancer [2]. The goal of breast cancer screening is to reduce mortality by detecting a tumor before its clinical manifestation. The widespread use of this method makes it possible to detect breast neoplasms at an early stage, which improves the prognosis for the patient's recovery. However, the result in the population can be achieved under the conditions of ensuring the quality of diagnostics and sufficient coverage of the target group.

Due to the significant number of patients referred for screening mammography, the X-ray laboratory assistant and the doctor have limited time to perform and evaluate the study [3]. Therefore, the result of diagnostics is influenced by the time factor, the quality of equipment, the organization of the process and the qualification of staff.

Lack of time is a serious problem with a wide coverage of screening. To conduct the study, it is necessary to obtain 4 images: two projections for each laterality. In this case, the X-ray technician must make the correct placement of the breast on the deck, otherwise the image will be considered unsuitable for interpretation. On average, one study takes about 20 minutes. The doctor, in turn, must interpret 4 images and make a study report with a conclusion on the BiRads scale [4]. As a result, it takes at least 10 minutes. Thus, the study of one patient lasts at least 20 minutes.

Currently, X-ray screening of breast cancer in women is carried out using digital radiological mammographs. Usually, the technology of automatic exposure control is used, which allows you to exclude the possibility of incorrectly set exposure parameters of the image. The detectors of such devices allow you to shoot 16-bit images up to 5000*5000 and higher with a physical pixel size of the matrix from 0.04 mm. Such images take up about 50 MB in memory. However, for screening, as a rule, 4 images are produced: two projections of each laterality. Thus, the study takes about 200 MB in memory.

Images are not stored on a digital mammograph-the device is part of the radiological information system (RIS), which contains at least one picture archiving and communication system (PACS). In turn, RIS is part of the hospital information system (HIS), which contains demographic and clinical information about the patient, which can be used by a doctor to compile a report.

The doctor's workplace is also included in the RIS and consists, usually, of a research viewer and a gateway to HIS.

The screening process can be organized on the basis of a hospital or with the help of mobile mammographic complexes. When using the mobile option, the doctor can evaluate the studies only when they are moved to PACS, which can happen only after a few days.

During screening studies, the obtaining and interpretation of images are almost always separated in time. In this case, the X-ray laboratory technician may make a mistake and make an incorrect set-up or perform a study with incorrect exposure parameters, which will make the study unsuitable for further interpretation.

The qualification of an X-ray technician is extremely important: in case of incorrect breast placement for the study, the images cannot be evaluated. But the most important qualification is that of a radiologist-mammologist. Due to the insignificant dynamic range of tissue densities and as a consequence of the low image contrast, the detection of pathology requires considerable experience from a doctor. The complexity of image interpretation generates hypo and hyperdiagnostics, that are, errors of the first and second kind. Often, it is possible to distinguish pathologies only with a retrospective comparison of images. Since the 2010s, the workplaces of doctors of some manufacturers have been equipped with a computer diagnostics system (CAD), which allows determining neoplasms on images. However, their effectiveness is quite low due to the large number of false positives cases [5].

Thus, the following approaches to improving the quality of breast cancer screening can be identified:

- Automatic assessment of the quality of breast placement at the laboratory assistant's workplace;

- Prediction of possible pathology at the laboratory assistant's workplace during screening based on mobile complexes;
- Automated compilation of a structured research report at the doctor's workplace.

II. Suggested system architecture

To automatically assess the quality of breast styling at the laboratory assistant's workplace, it is necessary to solve the following tasks:

- Determine the location of the breast nipple outline;
- Determine the outlines of the chest muscle for the MLO projection;
- Determine the type of X-ray density of the breast;
- Develop an algorithm for determining the quality of laying on the scale "Unsatisfactory-Good-Excellent"

The main limiting factor in the development of this subsystem is the speed of operation. The laying assessment procedure is integrated into the workflow of obtaining an image on an X-ray machine. And the evaluation stage should not exceed 20 seconds.

The subsystem should also have the ability to integrate any mammographic device into the diagnostic process. It is worth noting that the component must be fault-tolerant, and its failure should not affect other parts of the system.

Thus, the subsystem consists of a server and a client part and interacts via the grpc protocol. Dicom storage is used as a container for images. The subsystem architecture is shown in Fig. 1.

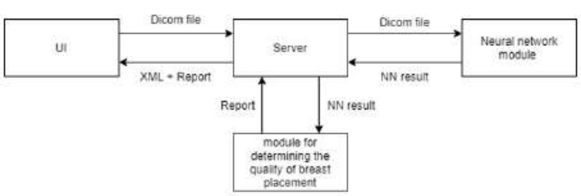


Figure 1. Architecture of the subsystem for assessing the quality of breast placement.

The purpose of the system for predicting possible pathology at the laboratory assistant's workplace is to early identify patients who may need additional research. Thus, the system can offer the laboratory assistant to additionally perform mammography with tomosynthesis, and a structured report on automatically found artifacts can also be included in the results of the study.

The purpose of the system of automated compilation of a structured research report at the doctor's workplace is to reduce the time for manual compilation of the report and the translation of the report into a machine-readable form.

A system capable of automated creation of a mammographic examination protocol must match the following criteria:

- The entire study should be provided to the system for input: several projections (images) for each laterality (left, right), their metadata, patient metadata (age, information about the hormonal background).
- Communication with the system should be carried out in a language based on BiRads.
- The system must have knowledge about radiation diagnostics in mammology: statistics on the occurrence of pathologies and their symptoms, the compatibility of symptoms, etc.

The study (4 images and a study report) is supplied to the input of the handler, which sends images to the recognition modules of diagnostic criteria. Each criterion (artifact) recognizes a separate recognition module. Based on the found criteria, the handler forms a request to the knowledge processing machine for verifying and supplementing search results. The knowledge processing machine for differential diagnostics can request additional data (from the user or another data source). The response of the knowledge processing machine is translated by the research handler into a machine - and human-readable report in a subject-dependent language based on BiRads.

The schematic diagram of this system is shown in Fig. 2.

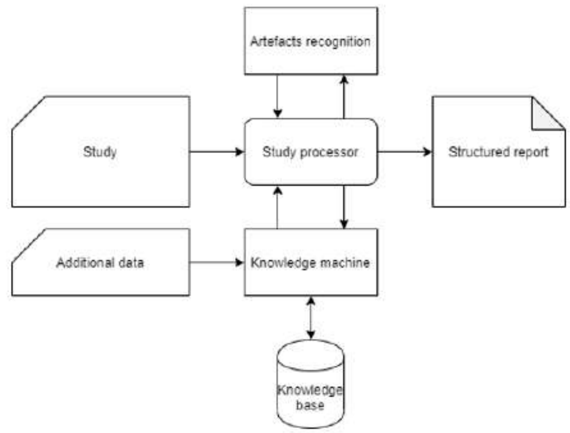


Figure 2. Schematic diagram of the decision support system of a radiologist.

III. Data preparation

The main limiting factor in the development of this subsystem is the insufficient amount of training data. To create a report, you need to mark up images for classification or segmentation by 21 criteria. Doctors should mark up images using professional medical monitors. In this regard, a markup language based on the Birds lexicon was developed. The markup language was built into the Dicom viewer MedXView and all the markup metadata was stored in Dicom storage. Currently, 800 studies have been collected with an acceptable quality of breast styling, all of them were marked up by doctors. At the same time, various pathologies were found only in 84 studies (10.5%). Thus, the initial data is sufficient only for the full implementation of the breast styling quality assessment system at the laboratory assistant's workplace.

IV. Implementation of the system of automatic assessment of the quality of breast placement

The tasks of the subsystem for automatic assessment of the quality of stacking are to determine the location of the outline of the breast nipple, to determine the outlines of the pectoral muscle for the MLO projection, to determine the type of X-ray density of the breast, to determine the quality of styling on the scale "Unsatisfactory-Good-Excellent".

To implement this task, 2 modules were developed. The module that controls the launch of neural networks and the processing of the results obtained. And the module that determines the quality of the placement.

During development, we used grpc technology to interact with the client. The subsystem of automatic assessment of stacking is a service.

The definition of the laying quality varies depending on the projections. At the moment, two algorithms have been implemented for CC and MLO projections. After the image is processed by the neural network and the coordinates of the nipple and muscle are received (for the MLO projection), the stacking assessment algorithm is started.

For the CC projection, the stacking is considered excellent if the nipple is located close to the center of the image vertically.

For the MLO projection, the styling is considered excellent if the nipple and the lower border of the muscle are on the same line or very close.

The neural network for determining the boundaries of the nipple correctly finds it in 84% of cases. The neural network, by determining the density, gives the correct result in 82% of cases. The neural network for determining the boundaries of the muscle as a result gives the correct result in 96% of cases.

V. Implementation of the system of automated compilation of a structured report

Taking into account the fact that there is not enough training data on the images of the report compilation system to implement the subsystem for recognizing diagnostic criteria, the system was partially implemented.

So, the input-output subsystem and the decision-making subsystem were implemented.

The decision-making subsystem consists of a knowledge processing machine and a knowledge base.

The tasks of the decision-making subsystem are the description and use of medical knowledge to generate forecasts of nosological forms in patients, verification of the output of each of the recognition modules, selection of a conclusion on the BiRads scale, conclusion of a hypothesis about the disease.

The task of the knowledge base is to formalize knowledge in the field of mammology. Medicine is a complex subject area and its knowledge model is difficult to describe in the form of a relational model. OSTIS technology is used to represent the model of knowledge about mammology [6].

The knowledge base consists of subject and utility ontologies. The subject ontology describes the BiRads language and the hierarchical structure of artifacts (diagnostic criteria) and diseases. In turn, the utility one is the basic ontology of IMS OSTIS.

The template for describing the patient's study was developed using a graphical representation of the SC-code. The description of the study consists of different parts:

- from the part of the patient's passport data,
- from the part of the image metadata description
- from the part describing all the diagnostic criteria found by the artifact recognition subsystem.

The passport data of the patient at the moment consists of the nodes of the hormonal status of the patient and his age. In the future, it will be possible to add information about previous studies and about the studies of relatives, which will affect the calculation of the probability of a particular disease.

The description of the image metadata consists of information about the projection, width and height, laterality, and so on. This information is taken from Dicom files.

Artifacts are divided into laterality artifacts and artifacts that have a specific location in the image. When processing an image, the neural network finds all the artifacts and their attributes and uses tags and values to describe them. The information is then translated into a graph view. Artifacts have properties that are also described in the graph structure. For example, the formation has margins, shape, density, size, distance from the nipple. For example, the total breast density can be considered an artifact of laterality.

Many nosological forms have been described. They are divided into normal, benign, suspicious, and potentially suspicious. Each nosological form has a probability distribution of occurrence by age. The nosological form has features, they are also artifacts, which are formalized according to the same template as the study.

The probabilities of occurrence of the nosological form are very rarely described in the literature. Therefore, we had to use the real experience of doctors - to get several invaluable consultations from specialists in their field. Also from the literature [7], descriptions of nosological forms and probabilities of occurrence of each artifact and its properties in this form were taken.

The domain ontology uses static and dynamic ontologies. We have filled static ontologies with immutable data about diseases, artifacts and their properties. The dynamic ones are filled in during the work and form a description of the patient's study based on DICOM.

The problem solver is a graphodynamic sc-machine (memory uses a semantic network as a model of knowledge representation), consisting of two parts:

- graphodynamic sc-memory;
- sc-operations systems.

The system of operations is agent-oriented and is a set of sc-operations, the condition for initiating which is the appearance of a certain structure in the system's memory. In this case, the operations interact with each other through the system memory by generating constructs that are the initiation conditions for another operation. With this approach, it becomes possible to ensure the flexibility and extensibility of the solver by adding and/or removing a certain set of operations from its composition.

There are currently seven agents:

- probability counting agent;
- an agent who creates a report on the patient's passport data;
- the agent who creates the research report;
- an agent that generates a general report;
- agent for generating the study structure
- report verification agent
- agent for verifying the size of the formation

As a result of the agents work, a list of nosological forms and their probabilities of existence in this study is obtained, sorted in descending order.

A BiRads-based language is also used to transfer information between subsystems and process research. The markup language is represented in both SC-code and XML.

The decision-making subsystem receives an XML file with the result of the recognition modules. After that, a graph structure is generated in semantic memory, the formations are validated and the modules are checked. Patient reports, studies are created and the probabilities of the presence of nosological forms are calculated. A structured report is created. In this case, the doctor can edit the created report and save it to PACS.

VI. Conclusion

Currently, decision support systems in radiation mammology focus on the detection and classification of neoplasms, despite the fact that the real work of a radiologist does not imply a diagnosing. Computer vision systems use a black box model

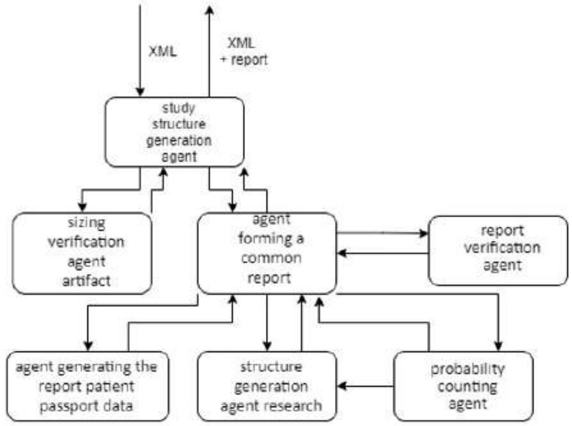


Figure 3. Interaction of agents of the structured report generation subsystem.

and do not explain the results of work, which is unacceptable in medicine.

The proposed hybrid approach, combining the work of computer vision subsystems and a problem solver based on graphodynamic sc-memory, can make a structured research report based on the detected anomalies and production output. This approach will reduce the time of the doctor-radiologist.

Also, for the first time, the decision support system of an X-ray technician is presented to improve the quality of breast styling. The system will improve the quality of the X-ray image and, accordingly, improve the evaluation of the study.

References

[1] Siegel RL, Miller KD, Jemal A. Cancer statistics, 2018. CA: A Cancer Journal for Clinicians. 2018; no 68, pp.7–30.

[2] Miller AB, Wall C, Baines CJ, Sun P, To T, Narod SA. Twenty five year follow-up for breast cancer incidence and mortality of the Canadian National Breast Screening Study: randomised screening trial. Bmj. 2014; pp.348-366.

[3] . Biesheuvel C, Barratt A, Howard K, Houssami N, Irwig L Effects of study methods and biases on estimates of invasive breast cancer overdetection with mammography screening: a systematic review. Lancet Oncol , 2007, no 8(12), pp. 1129–1138

[4] BiRads. Available at: https://www.acr.org/Clinical-Resources/Reporting-and-Data-Systems/Bi-Rads // (accessed 2021, May)

[5] Pasynkov D.V., Egoshin I.A., Kolchev A.A., Kliouchkin I.V., Busygina O.V. Diagnostic Value of 1st and 2nd Generation Computer Aided Detection Systems for Mammography: a Comparative Assessment. Medical Visualization. 2017, no 1, pp. 90-102.

[6] V.V.Golenkov, N.A.Gulyakina, Printsipy postroeniya massovoi semanticheskoi tekhnologii komponentnogo proektirovaniya intellektual'nykh sistem [Principles of building mass semantic technology for component design of intelligent systems], Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2011, pp 21-58

[7] Fisher, U., Baum, F., Luftner-Nagel, S.Direct Diagnosis in Radiology: Breast Imaging. 2020

Система поддержки принятия решений при проведении скрининга рака молочной железы

Каешко А.И., Ефимова А.А.

В настоящее время системы поддержки принятия решений в лучевой маммологии концентрируют внимания на выявлении и классификации опухолей несмотря на то, что реальная работа врача-рентгенолога не подразумевает постановку диагноза. Системы компьютерного зрения используют модель черного ящика и не объясняют результаты работы, что неприемлимо в медицине.

Предложенный гибридный подход, совмещающий работы подсистем компьютерного зрения и решателя задач на основе графодинамической sc-памяти может составлять структурированный отчет исследования на основе обнаруженных аномалий и продукционного вывода. Такой подход позволит сократить время врача-диагностика. Так же впервые представлена система поддержки принятия решений рентген-лаборанта для улучшения качества укладки молочной железы. Система позволит улучшить качество рентгеновского изображения и соответственно улучшить оценку исследования.

Received 01.06.2021

Chest Pathologies Analysis System Based On X-Ray Images Using Neural Network

Andrew Naumovich
Faculty of Applied Mathematics and Computer Science
Belarussian State University
Minsk, Belarus
andrew.naumovich@yandex.ru

***Abstract*—The paper deals with the problem of chest pathologies recognition based on images obtained from protable fluorographic devices. An approach based on the neural network technology is proposed. A complex algorithmic solution is delevoped. The algorithm is implemented as a specialized software package.**

***Keywords*—neural network, recognition, supervised learning, X-Ray image**

I. Introduction

The epidemiological situation in the country has revealed the necessity of population medical screening. This standard mean allows one to identify lung pathologies at an early stage quickly. Portable digital fluorography equipment is often used for medical screening. Such a type of the equipment makes it possible to obtain high-quality digital images with lung tissues visible. Therefore, the task of effective medical image processing technologies development is currently urgent [1], [2].

II. The problem of chest pathologies recognition

Algorithmic solutions to that problem exist [3], [4]. They are primarily based on X-Ray pathology classification using deep learning. Authors of the approach use DenseNet neural network for that purpose. The evaluation results are considered to be suitable for pratical usage.

However, during the analysis of existing solutions a few problems were discovered.

The first of them is that the algorithmic solution doesn‘t take into account the specifics of a particular medical device. Images taken with diverse devices can be very different and thus can be misclassified.

The second problem is purely technical in nature. Any medical device is a highly specialized device. Therefore, running e.g. a neural network on such a device is almost impossible.

The third one is connected with the interpretation of the recognition results. It is necessary for specialists-radiologists to validate their diagnosis. The implementation of that feature requires the development of specialized procedures for organizing effective medical image processing.

Thus, we can conclude that the existing algorithms actually perform the task of flourographic image analysis. However, this is not enough to make a final verdict.

III. Solving the problem

The development of the software package focused on portable equipment can be become a productive solution to the problem. It should not only increase the efficiency of the pathology diagnosis process, but also automate the entire process as much as possible [5]. The algorithmic solution should allow one to perform (in conditions of limited resources) image analysis quickly. In addiction to that, iterpretation and results visualization is required. Moreover, it is necessary to solve the non-trivial problem of technological compatibility of all heterogeneous components included in the software package.

The following architecture and composition of the software package is proposed. It is based on neural network data processing technology [6]. The complex consists of the following main modules:

- Algorithmic module.
- Database with reference images.
- Module for visualization and interpretation of results
- Calibration module.
- User interface.
- Expert advisor interface.

The overall architecture of the software package is shown in Fig. 1.

Due to the limited scope of paper, only the algorithmic module and database module of the software package will be considered.

IV. Algorithmic module

Let X be an input image and D - medical equipment information. As the result of the classification, each image X is assigned a vector of probabilities $P(X)$ of the presence of pathologies from the set T. With the help of the calibration module and the information D vector $P(X)$ is transformed into vector $P'(X)$ which, in turn, is used for a final conclusion. In addition to the probabilities of pathologies, the algorithmic module

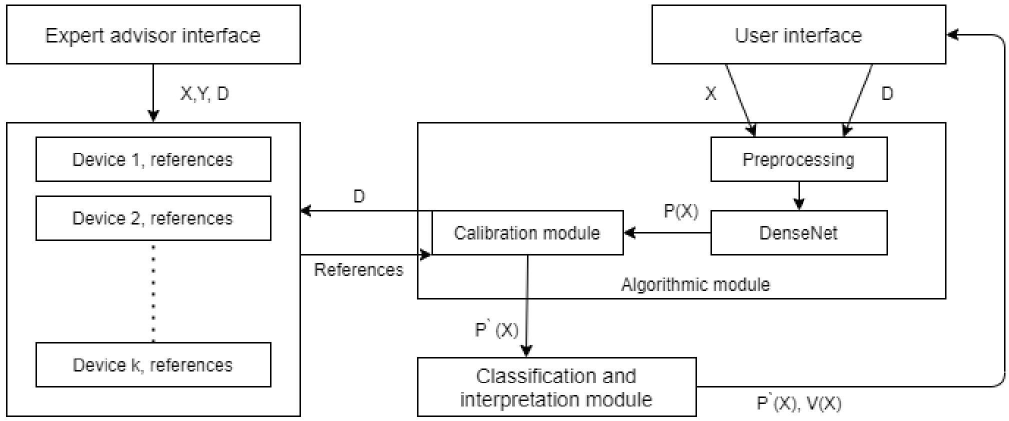


Figure 1. Software package architecture

should provide the user with the ability to interpret the predictions made.

Thus, the process of operation of the algorithmic module can be represented in the form of sequential execution of severals tages: pre-processing , classification of pathologies, calibration of predictions and interpretation of results. It is necessary to consider some of steps of the algorithm.

The key objective was to choose neural network architecture for image analysis. AlexNet, ResNet, EfficientNet and DenseNet architectures were considered. To measure their perfomance RSNA dataset was used and the results are presented in Table I.

Table I
MODEL SELECTION

	AlexNet	*ResNet50*	*EfficientNetB4*	*DenseNet121*
Accuracy	0.889	0.931	0.949	0.936

DenseNet121 was chosen for image encoding [7], [8] despite the fact that EfficientNet shows better results in classification. DenseNet121 has much fewer parameters and shows higher throughput capacity.

A grayscale image of size 384×384 is fed into the network. As a result of the network operation, two tensors are constructed. The first one is a vector of probabilities of pathologies. The second is the feature map of the last convolutional layer. This feature map is then used to visualize the results. A schematic architecture of the network is presented in Fig. 2.

The prediction calibration module is based on the threshold selection method. The appropriate threshold is selected for each pathology label (taking into account the information D). For this purpose, information about medical equipment and reference images is used, which is stored in the database module.

The interpretation and visualization procedure operates feature maps obtained from the last convolutional layer

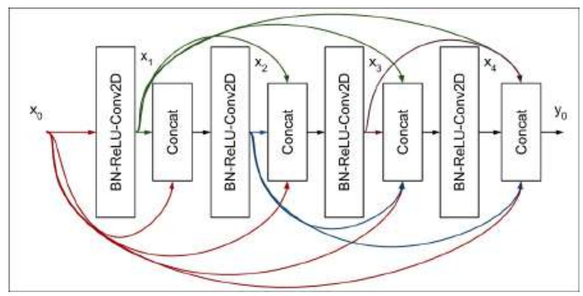


Figure 2. DenseNet architecture

of the network. Heatmaps of pathologies are constructed using the GradCAM algorithm.

V. Database module

The module is designed to store images from various devices and annotations for these images. Additionally, information about medical equipment must be stored.To ensure high speed of the algorithmic module, the images should be structured according to the medical equipment they belong to.

As medical screening is performed continuously, many new images appear with an unverified diagnosis made by the algorithm. Therefore, it is proposed to present the database module in the form of 2 databases: Hotbase and Warehouse. Warehouse stores data validated by an expert. Hotbase contains model's predictions that wait for validation. After validation completion data from Hotbase flows into Warehouse and can be used for model finetuning.

To organize the validation process, the expert interface is introduced.

Fig. 3 shows schema of the database module.

VI. Implementation details

The software package is implemented in Python using "tensorflow" and "tensorflow.keras" machine learning libraries. These libraries provide a framework for quick

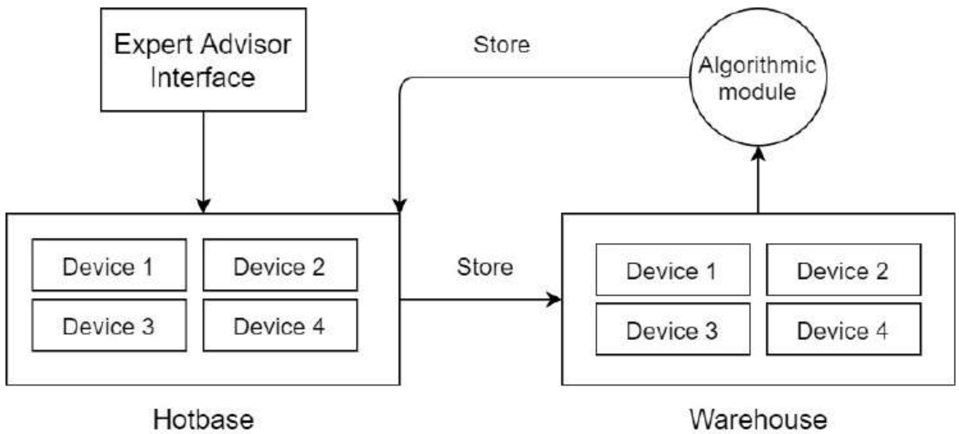


Figure 3. Database schema

building and training of the neural networks. For these purposes, “tensoflow“ mathematical operations are represented as a computational graph with the possibility of their automatic differentiation.

A separate problem is the organization of effective interaction between the user and the neural network. To solve this problem, we suggest using Tensorflow Serving. To call the model and get the results, we propose an approach based on the joint use of containerization and the HTTP protocol.

VII. Experiments and evaluation

The effectiveness of the approach proposed is confirmed by the results of experiments. Two X-Ray images datasets are used: RSNA and NIH [9].

RSNA dataset is a collection of annotated X-Ray images. The total size of the dataset is around 30000 samples. There are three classes available for classification: Normal, Lung Opacity, Not Normal / No Opacity. For simplicity only the first two classes were used. In addition to anomaly labels dataset's annotation provides bounding box for a pathology region which were not used in our research.

Evaluation results for the RSNA dataset are shown in Table II. The overall classification quality is 0.950. The test set includes 6045 samples, classes are balanced.

Table II
RSNA evaluation results

	Precision	***Recall***	***F1***
Normal	0.947	0.971	0.959
Pneumonia	0.956	0.919	0.937

NIH Chest X-Ray dataset is comprised of approximately 112 thousands of images from over 30 thousands of patients. The dataset contains 14 different lung patholigies that may occur at the same time. However, some pathologies were not actually found in the dataset (no samples of that class) and therefore excluded from the classification report.

The main difficulty we faced with that dataset is images diversity (different brightness, contrast, labels at the image). Moreover, the dataset is highly imbalanced which makes the training process more complicated.

To prevent the neural network from overfitting and to expand input data distribution artificially an extensive data augmentation is used. It includes the following transformations:

- Center crop.
- Random horizontal flip.
- Random rotation.

DenseNet network has been training for 10 epochs, 4500 steps each with batch size 8. Adam optimizer was used with default parameters $\alpha = 10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$.

The evaluation results on NIH test subset (22420 samples) are shown in Table III. As a calibration mechanism for this data set, we use threshold selectionmethod performed at the validation part of the dataset.

Table III
NIH evaluation results

	AUC-ROC	***F1***	***F1 (calibrated)***	***Threshold***
No Finding	0.792	0.740	0.770	0.001
Atelectasis	0.816	0.397	0.420	0.7994
Consolidation	0.797	0.207	0.219	0.9790
Infiltration	0.708	0.387	0.426	0.001
Pneumothorax	0.885	0.383	0.422	0.959
Edema	0.883	0.222	0.237	0.9590
Emphysema	0.896	0.287	0.347	0.979
Fibrosis	0.799	0.092	0.110	0.979
Effusion	0.883	0.556	0.555	0.599
Pneumonia	0.791	0.068	0.096	0.979
Cardiomegaly	0.891	0.301	0.351	0.979

An example of a pathology heatmap produced by GRAD-CAM algorithm is shown in Fig. 4.

Figure 4. Heatmap example

VIII. Conclusion

The paper presents a software package for X-Ray image processing from portable fluorographic devices. A complex algorithmic solution is proposed, which makes it possible to automate the process of image analysis. At the same time, it shows a good quality of prediction. In addition, it provides the ability to interpret the received predictions and has a high speed of operation. In view of the above, we can assume that the approach proposed in this paper has good practical prospects.

One of the problems mentioned is still unsolved. It is connected with the diversity of X-Ray images retrieved from different medical devices. To address that problem Domain Adaptation techniques can be used.Finally, the neural network can benefit from using bounding boxes for pathology regions. For that purpose Visual Image Transformer and Mask-RCNN architecture can be used.

References

[1] Krasnoproshin V.V., Vissia H.E.R.M. Decision-Making and Big Data. Las Nuevas Areas del Poder Economico Mundial. XII Acto Internacional de la Real Academia de Ciencias Economicas y Financieras. Barcelona: Royal Academy of Economy and Financial Sciences, 2017, pp. 105-120

[2] Krasnoproshin V., Obraztsov V., Vissia H. Solution of applied problems: formalization, methodology and justification. World Scientific Proceeding Series on Computer Engineering and Information Science, Vol. 3. "Computational Intelligence in Business and Economics". London, World Scientific, 2010, pp. 57-64

[3] Rahaman, Md Mamunur et al. 'Identification of COVID-19 Samples from Chest X-Ray Images Using Deep Learning: A Comparison of Transfer Learning Approaches'. 1 Jan. 2020, pp. 821–839

[4] CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learn-ing. Pranav Rajpurkar [et al.] [Electronic resource]. Available at https://arxiv.org/pdf/1711.05225.pdf (accessed 2021 March)

[5] V. Golenkov, N. Guliakina, V. Golovko, V. Krasnoproshin Artificial Intelligence Standardization Is a Key Challenge for the Technologies of the Future. Communications in Computer and Information Science. Volume 1282: "Open Semantic Technologies for Intelligent System". Springer. 2020, pp. 1-21

[6] V. A. Golovko, V. V. Krasnoproshin. Neural network technologies of data processing: a textbook. Minsk, BSU, 2017. 264p.

[7] Densely connected Convolutional Networks. Gao Huang [et al.] [Electronic resource]. Available at https://arxiv.org/abs/1608.06993 (accessed 2021 April)

[8] Hongyu Wang, Yong Xia. ChestNet: A Deep Neural Network for Classification of Thoracic Diseases on Chest Radiography. [Electronic resource]. Available at https://arxiv.org/abs/1807.03058 (accessed 2021 May)

[9] ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-supervised Classification and Localization of Common Thorax Diseases // Xiaosong Wang [et al.] [Electronic resource]. Available at http://dx.doi.org/10.1109/CVPR.2017.369 (accessed 2021 April)

Система анализа патологий грудной клетки на основе рентгеновских изображений с использованием нейронной сети

Наумович А.И.

В условиях пандемии для раннего обнаружения патологии легких часто используют портативные флюорографические устройства. Из-за высокой загруженности специалистов-рентгенологов актуальной является задача разработка эффективных технологий обработки и анализа изображений, ориентированных именно на такие устройства. В рамках работы предлагается программная система, архитектура и состав позволяет эффективно решать поставленную задачу. Она состоит из 6 основных модулей: алгоритмического модуля, базы данных, модуля визуализации и интерпретации, а также модуля калибровки. Алгоритмическое решение основано на нейросетевой технологии обработки данных с использованием фреймворка DenseNet. Прототип программной системы реализован на языке Python с использованием библиотек машинного обучения Tensorflow и Keras. Эффективность предложенного подхода демонстрируется на наборах рентгенографических снимков из базы данных RSNA и NIH.

Received 09.06.2021

Integrated Medical Information and Decision-Support System development based on shared metamodel definition

Aliaksandr Kurachkin
faculty of radiophysics and computer technologies
Belarusian State University
Minsk, Belarus
alex.v.kurochkin@gmail.com

Vasili Sadau
faculty of radiophysics and computer technologies
Belarusian State University
Minsk, Belarus
sadov@bsu.by

Aliaksandr Halavatyi
faculty of radiophysics and computer technologies
Belarusian State University
Minsk, Belarus
alex.halavatyi@gmail.com

***Abstract*—Developing integrated knowledge-based decision-support systems with persistent storage for medical use usually requires creating multiple program modules with a non-generic implementation that includes a lot of duplicate and non-generalized components for describing the dataset the system is designed to operate on. As a solution, this paper proposes a semantic metamodel definition for medical information systems based on extensible entity and attribute descriptor format, along with an integration framework that enables projecting external data to common format, simplifies data access by generating schema definitions and APIs for accessing persistant storage, and eliminates the need for manual user interface development by procedurally generating form-based and list-based views.**

***Keywords*—decision-support systems, medical information systems, intelligent diagnosis, expert systems**

I. INTRODUCTION

Medical information systems are an integral part of modern healthcare all around the globe. Generally, any kind of information system operating on patient or medical research data can be considered a medical information system; however, there are several problems specific to these kinds of systems: patient data handling, predictive and intelligent diagnosis assistance, and system integration [1].

Working with patient data generally requires isolated self-hosted database solutions with strong security measures to provide confidentiality, since healthcare data is considered sensitive and private. On the other hand, data sources themselves are generally numerous, loosely connected, non-standardized, denormalized and weakly structured, up to the point where it is possible to have duplicate or even conflicting information regarding the same patient.

Predictive and intelligent diagnosis assistance refers to the problem of using various decision support algorithms and predictive models. These methods are usually based either on rules and strict expect knowledge formalization, or on data analysis and supervised machine learning algorithms. Either way, both kinds of systems require a uniform way to access various kinds of data available regarding a specific type of medical research or specific pathology to evaluate performance relative to existing data, and to train the model in case of supervised machine learning.

Medical information system integration refers to the problem of adding new types of decision support models and handling data source changes in a way that is transparent, while retaining the simplicity of data access and generating predictions by medical staff using an integrated user interface. Ideally, adding a new kind of predictive model should be as simple as implementing a set of contracts within a predefined framework, in such a way that simple descriptive representation of the system is sufficient to automatically connect to existing data sources, provide necessary data access for predictive and intelligent diagnosis assistance, and present a user interface that can be used to both access relevant data and generate predictions based on it.

This paper proposes an integration framework for medical information systems based on descriptive semantic metamodel definition, consisting of 4 main parts: metamodel definition format itself, formal projection definition for adapting existing data sources to metamodel-compliant format, DDL generator for adapting metamodel definition to persistent data store and generating basic data access API, and user interface generation algorithm that can be used to autogenerate form-based and list-based views for data access based on metamodel. The framework itself provides a set of extension points at various stages: bootstrap stage to hook into initialization lifecycle events and register metamodel definitions nad projections, data link stage to modify the process of accessing a specific persistent storage, interface generation stage to implement custom user interface logic, and decision-support algorithm initialization stage to add custom predictive or knowledge-based models and integrate them with user interface.

II. Metamodel definition

Data model usually refers to definitions basic entities and attributes present in a set of data. For example, in medical information system, an entity may refer to general patient data, and consist of attributes such as name, sex, birthdate, address of residence, etc. In turn, a metamodel refers to metadata definitions that can be used to define data models, i.e., data model allows to describe the dataset with specific entity and attribute definitions, and metamodel allows to describe data models using generalized descriptors [2], [3]. Semantic metamodel is a type of metamodel that is used to describe knowledge base schemas in terms of semantic concepts and relationships [4].

Proposed metamodel format is based on attribute descriptors that, in turn, are grouped together under an entity descriptor.

Attribute descriptor contains the following information:

- attribute identifier – string-based identifier that must conform to variable identifier rules (should consist of alphanumeric characters and underscores, and should not start with numeric character)
- attribute name – human-friendly string name for this attribute, suitable for use as user interface field label
- attribute description – optional string value that contains extended description of a specific attribute, suitable for use as user interface field hint or description
- attribute type and attribute metadata – one of the predefined types for this specific attribute and additional type information
- visibility flag – a flag indicating whether this particular attribute is present on form representations and is able to be edited by user

The following types and metadata points are supported:

- string – string value, e.g., name, diagnosis, etc.
  - single-line or multi-line
  - validation regular expression
- number – floating-point or integer numeric value, e.g., blood glucose level, CT item voxel density, etc.
  - floating point precision
  - minimum value
  - maximum value
- boolean – a simple true/false value, e.g., presence or absence of a specific marker
  - descriptions for true and false values
  - preferred display style – as checkbox or as two radio buttons
- single-value categorical – a value defined as a single choice of multiple options, where each option is represented by string-based value and optional identifier, e.g., type of stroke, sex, etc.
  - options defined as string values or id & value tuples
  - preferred display style – as single-value drop-down or as radio button set
- multiple values categorical – a value defined as zero or more choices of multiple options, where each option is represented by string-based value and optional identifier, e.g., ASPECTS scale visible changes, or multiple related markers
  - options defined as string values or id & value tuples
  - preferred display style – as multiple-value drop-down or as checkbox set
- date, time or date with time – a value represented as a UNIX timestamp
  - minimum date
  - maximum date
  - minimum time
  - maximum time
- attachment – an attribute representing a file with unspecified format
  - required extension
  - maximum file size
- single reference – an attribute representing reference to another entity, which is mapped to a multiple reference from the other entity for one-to-many relationship
- multiple reference – an attribute representing a collection of references to another entity, which can be mapped either to a single reference for many-to-one relationship, or to multiple references from the other entity for many-to-many relationship

Attribute metadata definition is extensible, i.e., the set of properties described above can also include any non-standard definitions that can be later handled using various framework extension points. For example, it is possible to provide additional “type” metadata for string fields for implementing complex validation scenarios like e-mail validation, and then add appropriate validation handler that is able to parse required metadata and provide necessary enhancements at runtime.

Entity descriptors include:

- entity identifier – string-based identifier that must conform to variable identifier rules
- entity name – human-friendly string name for this entity
- entity description – extended description of the purpose and use cases for specific entity type
- mutability – an integer value indicating preferred handling for attribute modifications: 0-mutability

entities are accessed and modified in-place, while entities with mutability of $n > 0$ are modified by adding an entry to modification log, up to n entries, and accessed by reading latest version

- visibility flag – a flag indicating whether this particular entity type is visible as a separate entity on user interface – this can be useful to hide utility entities like many-to-many link tables when mapping to relational databases
- attribute set – a collection of attribute descriptors for this particular entity

An example of entity definition is as follows:

```
{
  "identifier": "OctScan",
  "name": "Optical Coherent Tomography
      scan protocol",
  "description": "OCT scan protocol,
      containing results from parsed
      tomography scan report and values
      from analyzing regions of image
      obtained by OCT scanner.",
  "mutability": 0,
  "visible": true,
  "attributes": [
    {
      "id": "eye",
      "name": "Eye",
      "description": "Which eye is
          examined during the scan",
      "type": "boolean",
      "typeMetadata": {
        "trueDescription": "OS",
        "falseDescription": "OD",
        "displayStyle": "radio"
      }
    },
    {
      "id": "octTemp",
      "name": "Temporal OCT",
      "description": "Retinal thickness
          in temporal side, as measured
          by OCT",
      "type": "number",
      "typeMetadata": {
        "precision": 3
        "min": 0
      }
    },
    {
      "id": "patient",
      "name": "Patient",
      "description": "Patient ref",
      "type": "ref",
      "typeMetadata": {
        "referenceType": "Patient"
      }
    }
  ]
}
```

Entity metadata definition is also extensible and can include any number of additional properties; various extension points throughout the framework can be used to access these properties and implement custom logic. Metamodel is represented as a set of entity descriptors available in the system.

For the purposes of universal serialization and usage in JavaScript language, the proposed metadata format is implemented as YAML or JSON document. Because of the extensible nature of YAML representation, it is also possible to define additional attributes that can be handled in extension points.

Metamodel can be passed to the framework as definition file in JSON or YAML format, or passed directly to the framework runtime during the bootstrap stage. Existing data sources may be adapted to common format by projecting individual data points to common metamodel format [5].

III. Adapting metamodel definition to database schema definition

Metamodel entity and attribute descriptors are designed to serve a storage-agnostic way of defining application schema. At the same time, it is possible to create database-specific adapters that convert metamodel-based definitions to compatible database schema definitions, and define an appropriate data access layer abstractions for specific platform.

Primary usage scenario of proposed metamodel is organizing access to centralized data storage, as explained earlier, that uses graph schema as a source. As such, an adapter for Neo4j graph database is implemented [5].

Since Neo4j database is schema-optional, it is not necessary to create schemas prior to manipulating actual data. However, metamodel can be used to translate API requests to database queries. For example, for OctScan entity, request for retireval of entity list can be triggered by API call, and metamodel definition can be used to translate the request to appropriate Neo4j Cypher query like this:

```
MATCH (x:OctScan)
RETURN (x.eye, x.octTemp)
```

The infromation about appropriate fields is taken directly from metamodel. Moreover, this approach allows to enable query support for any field, where own field constraints are translated to field MATCH clauses, and reference field constraints are translated to vertex-edge MATCH clauses. For example, retrieval of OctScan for specific patient with id 42 can be translated to the following request:

```
MATCH (x:OctScan)<--(p:Patient {id: 42})
RETURN (x.eye, x.octTemp)
```

In order to facilitate access to data storage itself, API endpoints can be used. Typical CRUD endpoints for data access can also be generated automatically based on metamodel definition. Server-side API route registration and handling is implemented using express framework for Node.JS runtime.

As mentioned earlier, metamodel can be adapted to other types of persistant storage solutions, including knowledge bases, using information in metamodel as metaknowledge.

The contracts of data access layer are abstract and should be replacable with a suitable driver implementation. However, since some of the more exotic database functions cannot (and should not) be abstracted away, API also provides an extension point for accessing native underlaying database connection, while expecting "external" calling code to be transformed to the format compliant with metamodel definitions.

Besides working as API data access proxy and database mapping layer, server-side APIs also call any validation rule that are defined for the attributes in metadata. Validation API is also extendable, which means it's possible to register custom validators that would be triggered based on specific information present in attribute metadata.

IV. User interface generation

Metamodel contains sufficient information to automatically generate a suitable form-based user interface for creating and modifying a single entity, and create paged list views for working with multiple entities [3].

In order to generate form interface, each attribute of a specific entity is mapped to a specific form control. For example, string fields are represented by single-line or multi-line input fields, depending on field metadata, while categorical fields are represented using radio buttons, checkboxes, etc.

References are represented as a special control type that can be used to add refernce to existing entity, create new entity in-place, or remove reference. Multiple references are represented as inline lists.

List interface uses a simple table representation, with columns corresponding to individual metamodel attributes. Column order and visibility can also be adjusted. Columns of most types also usually support custom sorting.

Generated user interface is automatically connected to API data access points for data retrieval and modifications. Using auto-generated interface allows to skip UI development entirely and integrate generated interface directly, and is also guaranteed to correspond to metamodel definition, thus making it much less error-prone. Custom extension points exist that allow to modify and create new rules for mapping attributes to specific control types.

Validation rules defined for each attribute can also be duplicated on client side – this way, validation error and hint appears as soon as the user finishes editing the field or attempts to submit the form. It should be noted that client-side validation does not replace server-side validation completely, since it would still be possible to submit incorrect data by directly accessing the API, so it's used only to enhance user experience.

V. Conclusion

Creating medical information systems and integrating them into existing infrastructure can be greatly simplified by using proposed metamodel-first approach. Using a single metamodel definition across data access APIs, persistent storage schema definitions and user interface generation can greatly simplify rapid prototyping of various medical information systems, as well as supplement the integration of new and existing decision-making models and knowledge-based solutions. The extensibility of the framework allows to adapt it to various types of medical diagnosis, while unified projected data representation greatly enhances the capabilities for interoperability with knowledge sources and allows to create training and validation datasets for various purposes.

References

[1] B. S. Abu-Nasser. “Medical Expert System Survey,” International Journal of Engineering and Information Systems (IJEAIS), vol. 1, iss. 7, pp. 218–224, 2017

[2] C. Gonzalez-Perez, B. Henderson-Sellers, “Metamodelling for Software Engineering,” Wiley, 219 p., 2008.

[3] A. Kurochkin. “Integrating medical data management and decision-making systems with common metamodel,” International Journal of Open Information Technologies, vol. 8, no. 12, pp. 49–53, 2020.

[4] T. Tokuda, Y. Kiyoki, H. Jaakkola, N. Yoshida, “24. Information Modelling and Knowledge Bases XXV (Frontiers in Artificial Intelligence and Applications)”, IOS Press, 336 p., 2014.

[5] A. Kurachkin, V. Sadau. Agregatsiya i indeksirovanie neskol'kikh istochnikov dannykh na osnove grafovoi modeli v bazakh dannykh meditsinskikh ekspertnykh sistem [Aggregation and indexing of multiple data sources based on a graph model in databases of medical expert systems]. *Informatika [Informatics]*, 2020. pp. 25-35

Разработка интегрированных медицинских информационных систем поддержки принятия решений на основе общей метамодели

А. В. Курочкин, В. С. Садов, А. И. Головатый

Разработка интегрированных основанных на знаниях систем поддержки принятия решений с долговременным хранилищем для использования в медицине требует создания нескольких программных модулей с собственной реализацией, которая включает множество дублирующихся и необобщенных компонентов для описания тех данных, которыми оперирует система. В качестве решения этой проблемы в работе предлагается описательное задание в виде семантической метамодели для медицинских информационных системах, в основе которого лежит расширяемый формат дескрипторов сущностей и атрибутов, а также интеграционную среду, которая позволяет приводить внешние данные к общему формату, упрощает доступ к данным благодаря генерации описаний схемы и программных интерфейсов для доступа к долговременному хранилищу, а также устраняет необходимость в ручной разработке пользовательского интерфейса благодаря процедурной генерации представлений на основе форм и списков.

Received 31.05.2021

Evaluation of the quality of functioning of a telecommunications company based on Data Mining technology

Globa Larysa, Novogrudska Rina, Moroz Anastasia
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
Kyiv, Ukraine
lgloba@its.kpi.ua, rinan@ukr.net

***Abstract*—The paper presents an approach to assessing the quality of a telecommunications company, which can be used to predict the churn of subscribers from a telecom operator. The originality of the approach lies in the use of such mathematical methods that make it possible to determine the main set of parameters due to which specific subscribers are inclined to change their mobile operator. The approach is based on data mining, namely the method of cluster analysis, which most fully allows assessing the quality of telecommunications companies functioning. Mathematical modeling of predicting customer churn using methods of association rules, decision trees and bagging has been carried out. The parameters characterizing the interaction of a mobile operator with end users are described. The parameters that have the greatest influence on the client's decision to refuse the services of a mobile operator have been determined.**

***Keywords*—Telecom operator, Quality assessment, Customer churn, Data Mining, Modeling.**

I. Introduction

The billing model, in which the collection and processing of data was carried out by one specialized automated billing system that perfors only finensial calculations, does not satisfy the needs of the telecom operator. It is being replaced by models where a separate system is involved in collecting, processing and preparing data. The main reasons for the development of such a system are:

- The market demands from the operator a variety of offered tariff plans and service packages, quick commissioning of services, and therefore carrying out various analyzes of the accumulated data and according to various criteria.
- Telecom operators are forced to provide services on dissimilar equipment, which significantly complicates the collection of statistics through a variety of accounting data formats.
- Large telecom operators are expanding geographically to include regional companies. This leads to a sharp increase in heterogeneous equipment and the volume of accounting data.

At the same time, there is a need for centralized accounting of the provided communication services in the conditions of operation of several different billing systems in the regions. In the course of their development, operators developed their own data preprocessing systems. The complexity of the operation of such systems lies in the need for their constant revision in connection with the emergence of new services, new equipment, and changes in the data formats coming from the equipment. In this case, the operator is forced to use the resources of a team of highly qualified programmers to constantly improve the system. A standard situation is when the data preprocessing is carried out by various systems at a telecom provider company.

With the modern development of telecommunications, systems that calculate the consumed services by users provided on the basis of one or another telecommunications equipment have received the same development. Such systems are usually called billing systems [1]. Their main purpose is to display the number of services consumed by the subscriber and write off funds in accordance with the cost per service unit. Based on this, it is possible to give a definition: billing system (BS) is a software complex that records the volume of services consumed by subscribers, calculates and debits funds, according to tariffs. In terms of relationships with content providers, the billing system must provide the following capabilities: firstly, it must guarantee the transmission of information to partners about the use of their services by specific subscribers, and secondly, bill the end consumer for using the service (or the content provider independently issues an invoice).

In the context of the rapid development of the communication network, replacement and addition of switching equipment, the provision of new types of services, it becomes expedient to switch to an industrial solution of data preprocessing problems. Products of this class have long been successfully used by large operators in countries with developed communication infrastructure. They are called Mediation systems. Mediation systems represent the level between the network infrastructure and OSS / BSS (Operation Support System / Business Support System) systems. The main purpose of Mediation systems is to transform data received from network elements into information that can be interpreted by the billing system and other business systems of the operator. Also, such systems are called previous billing systems.

In the context of sustainable development of telecommunications, as well as the presence of a sufficient number of provider companies in the telecom services market, important tasks are to assess the quality of telecom operators' functioning and develop a strategy to prevent subscriber churn. However, for this it is necessary to increase the level of intellectualization of OSS / BSS systems and to include in their structure tools for processing and analyzing large volumes of accumulated data.

II. State of art and background

Currently, the usual methods of increasing the loyalty of old customers and attracting new customers (mass advertising, traditional marketing, low prices) do not produce the desired positive result. That is why concepts that allow personalized sales of goods and services are becoming a priority all over the world.

An example is Customer Relationship Management (CRM) – customer relationship management [2]. CRM is not a technology or software product. This is a business strategy based on a customer-oriented [3] approach, that is, we can say that the main task of CRM is to increase and improve the efficiency of business processes, directly aimed at attracting new and retaining existing clients [4]. CRM systems are most effective in those areas of business where it is possible to accumulate a large amount of useful information about each client. Here, the CRM strategy is used primarily to combat customer churn. In telecommunications, this term refers to the process of enticing customers from one telecom operator to another, or simply the outflow of customers from the operator [5]. The annual rate of customer churn reaches 25 – 30%. Operators who have this indicator is maximum, will not be able to get a return on investment in new subscribers, since it takes about three years to get back the funds spent on replacing each lost client with a new one, that is, it takes about three years to acquire clients [6, 7].

Currently, OSS / BSS have gained active development to support the operational and business activities of telecom operators, with the help of full or partial automation of these activities. Operational activities include processes that interact mainly with network equipment and the network, such as: accounting and planning of network resources, management and provision of services, management of quality characteristics of services. Business activities include customer-centric processes such as processing and invoicing, collecting payments, proposing new products, and much more [8].

Thus, both OSS / BSS systems and CRM systems, although they allow to some extent process big data of telecom operators, it is necessary to integrate them and supplement them with methods of deeper analysis and structuring of such data to solve the problem of assessing the effectiveness of the telecom operator's functioning, and also predicting the churn of customers from the operator.

A. Data Mining in telecommunications

Today, the Data Mining solution [4] is widely used by telecommunications operators, network operators and Internet providers in order to improve the processing of unstructured large amounts of data for making better decisions in their business. Analytics enables telecommunications providers to significantly improve the cost-effectiveness of their service delivery.

Data Mining models are positioned on a dataset with known results and are used to predict results from other datasets. These are classification models (they describe the rules by which the description of an object can be attributed to one of the classes) and sequence models (they describe functions by which it is possible to predict the change in continuous numerical parameters).

In telecommunications, it is very important to identify categories of customers with similar stereotypes for using services, as a result, a pricing policy is being developed; analysis of failures; prediction of peak loads. Telecom systems generate an extremely large amount of data that needs to be processed and analyzed. Thus, data mining is becoming a very important and widely implemented part of business processes in the systems of a telecom operator [9].

Data Mining methods make it possible to effectively solve the problems of structural engineering design of innovative technical systems in telecommunications [2]. These methods have much in common with methods for solving problems of classification, diagnostics and pattern recognition. Nevertheless, one of their main distinguishing features is the function of interpreting the patterns that form the basic rules for including objects in equivalence classes. Therefore, such methods are becoming more common today. Developing intelligent data processing for telecommunications companies is essential to:

- Reducing the computational complexity of big data processing methods for the operator to provide services to a subscriber with a given quality of service;
- Predicting the risks that may arise from the operation of the telecommunications system;
- To be able to identify faults in the system and find out the cause of their occurrence.

Some of the existing algorithms can be adapted to compute large distributed information arrays. At the same time, serious difficulties can arise in the visual presentation of the results – due to the huge amount of information entering the input, the number of different reports at the output increases dramatically. For their convenience, new mathematical methods are needed, which are fundamentally different from the report generators used for traditional storage. In this regard, it is relevant to use just such mathematical methods that make it possible to move from obtaining data to obtaining information, and from it to obtaining knowledge about the process and patterns of providing services to subscribers and, thus, increasing the efficiency of managing this process.

III. The method of clustering big data of a telecom operator

In data mining tasks, using cluster analysis, a complex construction of data for classification is created, patterns are identified, hypotheses are formed and tested, etc. In addition, cluster analysis is often used to identify data that ”stands out” from others, since such data correspond to points located at a distance from any cluster. Cluster analysis is used to compress and summarize data. A method for clustering big data of a telecom operator is proposed, the use of which allows assessing the quality of functioning of telecommunication companies [10]. The method uses decision trees, association rules, and bagging to predict customer churn from a carrier. The method proposed in the study allows you to go through all the stages of preparation, processing and analysis of big data of a telecom operator. The method includes next stages:

- Subject domain analysis.
- Problem statement.
- Data preparation.
- Model development.
- Model checking.
- Model selection.
- Model preparation.
- Model updating.

The following systems are responsible for the first three stages: analysis of the subject area, statement of the problem, preparation of data: billing, OSS / BSS. The following steps are performed by the CRM system.

All of the above stages together give a high-quality result in solving the problem of customer churn in the telecommunications sector. This question is very important for doing business and determining the new course of the company.

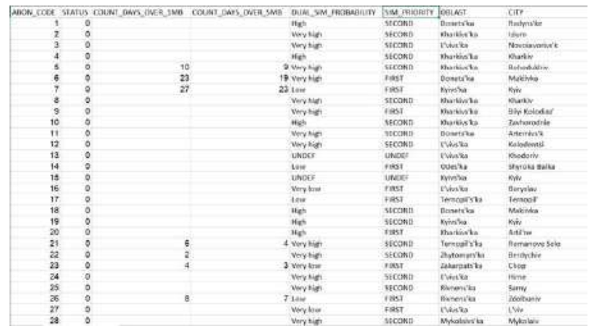
The first thing to do is to analyze the telecommunications structure. Data that will be processed in the future are collected from network equipment. Further, in order to solve the problem of determining the cause of customer churn, it is necessary to identify which factors influence this, which data components are of the greatest importance.

After collecting the data, they need to be structured. To do this, the ”raw” data must go through the following steps: verification, correction, consolidation, filtration, separation, routing, representation, distribution.

The next stage is modeling.

IV. Modeling and predicting the churn of subscribers of a telecom operator

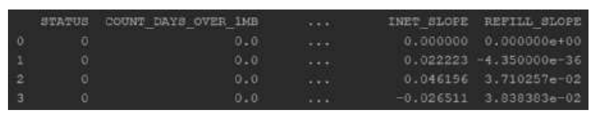
The input data for the modeling were real customer data obtained from one of the leading telecom operators in Ukraine (Fig. 1). We will call such data ”raw”. There are spaces in the data table, which is invalid for analysis. Gaps can introduce additional prediction errors.



ABON_CODE	STATUS	COUNT_DAYS_OVER_1MB	COUNT_DAYS_OVER_5MB	DUAL_SIM_PROBABILITY	SIM_PRIORITY	OBLAST	CITY
1	0			High	SECOND	Donets'ka	Rodyns'ke
2	0			Very high	SECOND	Kharkivs'ka	Izium
3	0			Very high	SECOND	L'vivs'ka	Novoiavorivs'k
4	0			High	SECOND	Kharkivs'ka	Kharkiv
5	0	10	9	Very high	SECOND	Kharkivs'ka	[illegible]
6	0	23	19	Very high	FIRST	Donets'ka	Makiivka
7	0	27	23	Low	FIRST	Kyivs'ka	Kyiv
8	0			Very high	SECOND	Kharkivs'ka	Kharkiv
9	0			Very high	FIRST	Kharkivs'ka	Bilyi Kolodiaz'
10	0			High	SECOND	Kharkivs'ka	[illegible]
11	0			Very high	SECOND	Donets'ka	Artemivs'k
12	0			Very high	SECOND	L'vivs'ka	[illegible]
13	0			UNDEF	UNDEF	L'vivs'ka	Khodoriv
14	0			Low	FIRST	Odes'ka	Shyroka Balka
15	0			UNDEF	UNDEF	Kyivs'ka	Kyiv
16	0			Very low	FIRST	L'vivs'ka	Boryslav
17	0			Low	FIRST	Ternopil's'ka	Ternopil'
18	0			High	SECOND	Donets'ka	Makiivka
19	0			High	SECOND	Kyivs'ka	Kyiv
20	0			High	FIRST	Kharkivs'ka	Artil'ne
21	0	6	4	Very high	SECOND	Ternopil's'ka	Romanove Selo
22	0	2		Very high	SECOND	Zhytomyrs'ka	Berdychiv
23	0	4	3	Very low	FIRST	Zakarpats'ka	Chop
24	0			Very high	SECOND	L'vivs'ka	Hirne
25	0			Very high	SECOND	Rivnens'ka	Sarny
26	0	8	7	Low	FIRST	Rivnens'ka	Zdolbuniv
27	0			Very low	FIRST	L'vivs'ka	L'viv
28	0			Very high	SECOND	Mykolaivs'ka	Mykolaiv

Figure 1. Initial modeling data.

For the correct execution of the process of analyzing the dataset, you need to fill in the gaps that are present in the input table. To do this, use the search function for gaps and fill them with zeros (Fig. 2). Thus, when simulating, we get a high probability of truthful results. As a result, we will receive ”preliminary prepared data”.



	STATUS	COUNT_DAYS_OVER_1MB	...	INET_SLOPE	REFILL_SLOPE
0	0	0.0	...	0.000000	0.000000e+00
1	0	0.0	...	0.022223	-4.350000e-36
2	0	0.0	...	0.046196	3.710257e-02
3	0	0.0	...	-0.026511	3.838383e-02

Figure 2. Pre-prepared data for modeling.

When modeling customer behavior in order to predict subscriber churn the random forest method was used, with the help of which the modeling is carried out. The random forest is one example of ensemble classifiers. The modeling was performed in the Python programming language. In the course of the study, a sample of subscribers inclined to outflow from the given probabilities was determined (Table 1)

Table I
Subscribers who are prone to churn

№	prob_true
1977	0.9333
2696	0.9167
2708	0.9233
2924	0.9041

The work uses Data Mining methods: associative rules, decision trees and bagging in order to increase the efficiency of prediction by increasing the accuracy of the probabilities of subscriber churn.

It was found that the bagging method allows you to get more accurate results, by 7% compared to the method of association rules and up to 1% better than the value of the metric of the decision tree method, the accuracy of the results obtained is 78.84%.

Also, in the process of modeling, the problem of identifying the factors that most affect the decision of the subscriber to switch to the services of another telecommunications company was solved. The modeling revealed the following parameters that most affect the churn of customers, they are presented in table. 2. These parameters are:

- duration of using the services of the telecommunications company;
- the number of days during which the subscriber does not use the services;
- average value of all days during which the subscriber does not use services;
- average value of all active days outgoing calls to numbers of other mobile operators.

Table II
IMPACT OF PARAMETERS ON THE CUSTOMER CHURN PROCESS

$N^{\underline{o}}$	importance	labels
3 2	0.075648	Duration of using the services of the telecommunications company
2 3	0.068619	The number of days during which the subscriber does not use the services
2 1	0.060132	Average value of all days during which the subscriber does not use services
2 2	0.055328	Average of all active days
1 9	0.045268	Outgoing calls to numbers of other mobile operators

The simulation revealed the subscribers who are most susceptible to the transition to the services of another telecommunications company. Table 3. shows a sample of subscribers prone to churn from a mobile operator. The churn rate percentage is shown next to each subscriber. It can be concluded that, based on the consideration of these data, it is possible to influence individual users and provide each of them with those services in which the subscriber is most interested. This way you can prevent customer churn.

Table III
SUBSCRIBERS WHO ARE PRONE TO CHURN

$N^{\underline{o}}$	prob_true
13788	0.99256
13859	0.99121
17185	0.98286
11595	0.95789
10842	0.94865

Conclusion

A telecom operator collects large amounts of data in the course of its work; for full processing, this data must be structured. Large amounts of data are processed using systems: OSS / BSS, billing, CRM. Correct storage, structuring and analysis of this data in aggregate with high accuracy allows predicting the behavior of the subscriber and his use of the operator's services, as well as the possibility of the subscriber's transition to another operator.

Mathematical modeling for predicting customer churn was performed using the methods of association rules, decision trees and bagging. Such methods allowes to get more accurate results, by 7% compared to the method of association rules and up to 1% better than the value of the metric of the decision tree method, the accuracy of the results obtained is 78.84%. Also, as a result of the study, patterns and factors were identified that most affect the subscriber's decision to refuse the services of a telecom operator.

The use of data obtained in the course of research is especially effective at the stage of concluding an agreement with a client, which allows you to build relationships with a client in the most beneficial way for the company.

References

[1] Nechiporuk, DV, Features of Data Mining Technology, Don State Technical University, Rostov-on-Don, Russian Federation.

[2] Skorbota, V. S., "How Ukrainians choose a mobile operator," (2017) [Online]. https://biz.nv.ua/experts/skorbota/kak-ukraintsy-vybirajut-mobilnogo-operatora-1930612.html.

[3] Dyakonov PV, Handbook on the use of the PC MATLAB system, Moscow: "Fiz-Matlit", (1993).

[4] Palmov, SV Comparison of the capabilities of various methods of Data Mining technology in the analysis of personal traffic. // XII Russian scientific conference of professors and teaching staff, researchers and graduate students, Samara, PDATU, (2005), thesis, pp. 285–287.

[5] Sasaki, Y., The truth of the F-measure, 26th October, (2007)

[6] Scholl, F., Deep Learning in R, St. Petersburg: Peter Publishing House, (2018).

[7] Kaftannikov, IL, Parasich, AV, Features of the use of decision trees in problems South Ural State University, Chelyabinsk.

[8] ALEXANDER, Schill; LARYSA, Globa; TETIANA, Kot. Applying business process modeling method when telecommunication services development. In: 2011 21st International Crimean Conference "Microwave & Telecommunication Technology". IEEE, 2011. pp. 457–458.

[9] Globa, L.S., Novogrudska, R.L., Koval, A.V.: Ontology Model of Telecom Operator Big Data. Proceedings of IEEE International Black Sea Conference on Communication and Netwoorking (BlackSeaCom), 1–5 (2018). doi:10.1109/BlackSeaCom.2018.8433710.

[10] Y Buhaienko, LS Globa, A Liashenko, M Grebinechenko. Analysis of clustering algorithms for use in the universal data processing system. In: Open Semantic Technologies for Intelligent Systems. Research Papers Collection, Issue 4, Belarusian State University of Informatics and Radioelectronics, Minsk, 2020, pp. 101–104

Оценка качества функционирования телекомуникационной компании на основе технологии Data Mining

Л. Глоба, Р. Новогрудская, А. Мороз

В статье представлен подход к оценке качества функционирования телекомуникационной компании, который может быть использован для предсказания оттока абонентов от оператора связи. Оригинальность подхода заключается в использовании таких математических методов, которые позволяют определить основной набор параметров, из-за которых конкретные абоненты склонны к смене мобильного оператора. Подход основан на data mining, а именно методе кластерного анализа, который в наиболее полной мере позволяет оценить качество функционирования компаний телекоммуникационной связи. Проведено математическое моделирование предсказания оттока клиентов с помощью методов ассоциативных правил, деревья решений и bagging. Описаны параметры, характеризующие взаимодействие оператора мобильной связи с конечными абонентами. Определены параметры, оказывающие наибольшее влияние на решение клиента об отказе от услуг мобильного оператора.

Received 14.05.2021

Interactive and intelligent tools of the GeoBazaDannych system

Valery B. Taranchuk
Department of Computer Applications and Systems
Belarusian State University
Minsk, Republic of Belarus
taranchuk@bsu.by

***Abstract*—In this paper, from the perspective of creating and maintaining geological or geoecological models, methodological and technical issues, ways of developing the system GeoBazaDannych [1], expanding its functionality by including data mining modules of the Wolfram Mathematica computer algebra system are considered. In particular, the examples illustrate the tools for preparing control sets of geodata for validation, testing and evaluation of related neural network models. It is shown how the adopted architecture, the implemented concept of constructing the system GeoBazaDannych, allow expanding the functionality by including additional software components. Examples illustrate the variants for choosing the best clustering algorithms.**

***Keywords*—system GeoBazaDannych, intelligent adaptation of digital fields, clustering**

I. Introduction

The features of solving the problems of developing and implementing computer-based geological and geoecological models with the means of their adaptation and self-adjustment, the main approaches to processing, analysis, interpretation of the data used and obtained are noted in [2] – [5]. The mentioned publications provide several basic solutions to the issues of preprocessing, intelligent analysis of geodata by means of the computer system GeoBazaDannych. It is emphasized that at this stage, data mining is among the priority areas of research and development, the corresponding classes of systems for its implementation are listed. The results and methodological recommendations of cluster analysis of geodata obtained with the environment of the system GeoBazaDannych are discussed below.

II. General information about cluster analysis

The solution to the problem of cluster analysis (segmentation) [6], [7] is the partitions that satisfy the accepted criterion. The criterion is usually a functionally formalized set of rules for determining the levels of differences in partitions and groupings (the objective function). In data mining, segmentation can be used as an independent tool for making decisions about data distribution, for monitoring characteristics and subsequent analysis of data sets of certain clusters. Alternatively, cluster analysis can serve as a preprocessing stage for other algorithms. Segmentation is also used to detect atypical outlier objects (values that are "far" from any cluster), in other words, it is a novelty detection, such objects may be more interesting than those included in clusters. In addition, cluster analysis, unlike most mathematical and statistical methods, does not impose any restrictions on the type of source data under consideration. An important advantage of cluster analysis is that when it is performed, it is possible to divide objects not only by one parameter, but by a set of features. In addition, cluster analysis, unlike most mathematical and statistical methods, does not impose any restrictions on the type of source data under consideration. It is well known that cluster analysis is widely used in many fields, in particular, in computer systems for pattern recognition, image analysis, information retrieval, data compression, computer graphics, bioinformatics, machine learning. The following are representative examples and the cluster analysis tools implemented in the system GeoBazaDannych environment are noted.

III. Brief information about the software system GeoBazaDannych

The interactive computer system GeoBazaDannych is the complex of intelligent computer subsystems, mathematical, algorithmic and software for filling, maintaining and visualizing databases, input data for simulation and mathematical models, tools for conducting computational experiments, algorithmic tools and software for creating continuously updated computer models. GeoBazaDannych's subsystems allow you to calculate and perform expert assessments of local and integral characteristics of ecosystems in different approximations, calculate distributions of concentrations and mass balances of pollutants; create permanent models of oil production facilities; generate and display thematic maps on hard copies. The main components of the system GeoBazaDannych [1]:

- the data generator Gen_DATv;
- the generator and editor of thematic maps and digital fields Gen_MAPw;

- modules for organizing the operation of geographic information systems in interactive or batch modes;
- the software package Geo_mdl – mathematical, algorithmic and software tools for building geological models of soil layers, multi-layer reservoirs; modules for three-dimensional visualization of dynamic processes of distribution of water-soluble pollutants in active soil layers;
- software and algorithmic support for the formation and maintenance of permanent hydrodynamic models of multiphase filtration in porous, fractured media;
- the integrated software complex of the composer of digital geological and geoecological models (GGMD).

IV. What is the novelty of the presented results

To explain the novelty of the results presented in this paper, we note that [4], [5] provide examples of interactive formation of digital models of geological objects in computational experiments that meet the intuitive requirements of the expert. Examples of approximation and reconstruction of the digital field, its interactive adaptation by means of the system GeoBazaDannych were discussed. The examples of approximation and reconstruction of the digital field, its interactive adaptation by means of the system GeoBazaDannych and evaluation of the accuracy of results using the tools of the GGMD complex illustrate the unique capabilities of the developed methods and software. In [2], [3], the results of the use of artificial neural networks in the analysis and interpretation of geospatial data are presented and discussed, the possibilities of obtaining and visualizing errors are described. This paper discusses variants and provides tools for implementing cluster analysis of geodata in the environment of the system GeoBazaDannych; recommendations are given for choosing the optimal parameters of classification algorithms when dividing the studied objects and features into groups that are homogeneous in the accepted sense. Particular additions are discussed and illustrated with examples.

It is important that the corresponding additions to the system GeoBazaDannych, new instrumental content are implemented within the framework of the concept of computer model development adopted and actively developed in recent years (see, for example, [1], [8]), the basis of which is integration into software packages, complexes of modules of computer algebra systems, ensuring functioning in one environment, a single interface of current software modules and extensions.

V. Tools, examples of cluster analysis of geodata

The examples below are calculated with the data [3], from the two surfaces considered there, *zSurfB* is selected for illustrations. Recall that the simulated surface (a reference for evaluating the accuracy of numerical experiments, approximate calculations by various methods) has a complete mathematical description, for clarity, Fig. 1 shows the isolines (contour lines) of the *zSurfB* levels.

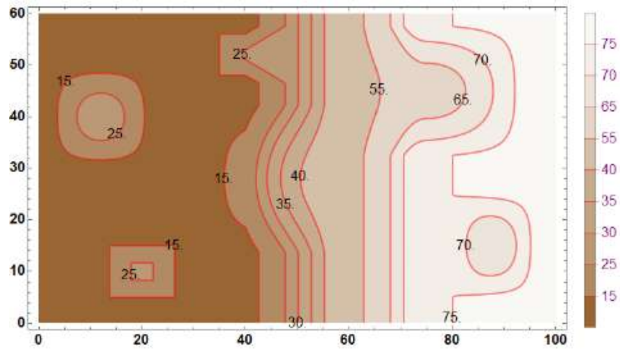


Figure 1. Contour map of the reference surface *zSurfB*.

The corresponding scheme of their placement is shown in Fig. 2, where the isolines of the reference surface and the one reconstructed in Wolfram Mathematica are also given (the Interpolation method, InterpolationOrder = 1). Data for demonstrations of methods and algorithms of intellectual analysis are obtained by simulation of measurements, the corresponding data set – the points of measurements of the level of the restored surface, representing (in fact) a scattered set of points, are interpreted as data on observation profiles.

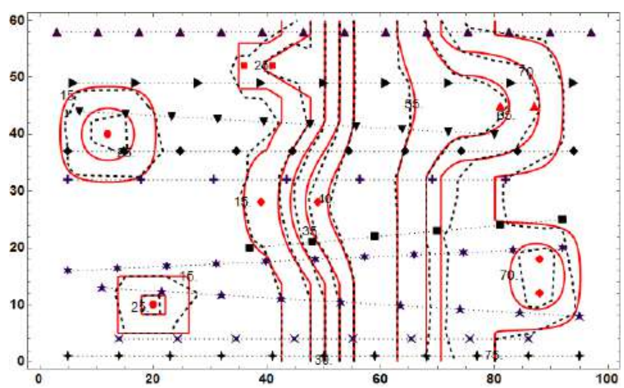


Figure 2. A scheme of points with level measurements, a map of isolines of the reference and reconstructed surfaces.

VI. Effects of the accepted clustering method

Cluster analysis allows for many different types of clustering techniques/algorithms to determine the final result. Determining the number of clusters is one of the most important problems of segmentation. In a broader sense, this is the problem of initializing the algorithm:

selection of optimal values of control parameters, evaluation functions used, metrics, stopping conditions, etc. In the examples below, a priori information is used, the number of clusters is initially set to 7. Why so much – it is taken into account that in the initial data, measurements were carried out for a surface that included: the base surface and 6 different distortions of it with individual positioning of perturbations. In the illustrations (to remind the data source), the isolines of the reference surface are given in red dotted lines. Below are the results that illustrate the features of the most commonly used clustering algorithms.

The effects of the accepted clustering method are illustrated by the schemes in Fig. 3. Clustering in the examples of this series was considered only for pairs of coordinates, i.e. the relative position of the points of the scattered set was taken into account, moreover, the FindClusters function with different criteria was used in the program module, the norm in the examples of the series Fig. 3 was calculated using the DistanceFunction EuclideanDistance metric.

Generally speaking, the corresponding software application included in the system GeoBazaDannych from the Wolfram Mathematica allows variants of the clustering method (Criterion function): Automatic, Agglomerate, DBSCAN, Gaussian mixing, Jarvispatrick, KMeans, KMedoids, Neighborhood, Optimization, SpanningTree, Spectral [9]. What segmentation methods are used in the calculations are recorded in the headers of the diagrams. Representative clustering options are shown, namely Automatic (Wolfram Mathematica automatically selects the method, the Wolfram Language will automatically try to pick the best method for a particular computation), k-means (k-means clustering algorithm [10]), k-medoids (splitting into medoids [11]), Spectral (spectral clustering algorithm [12]). These results are quite indicative. At the same time, taking into account the reference and the digital field of the original, we can consider the clustering option by the Spectral method as preferable.

VII. The impact of the metric

In the examples discussed above, as well as in this series of results, the similarity or difference between the classified objects is established depending on the metric distance between them. The issues of measuring the proximity of objects have to be solved with any interpretation of clusters and various classification methods, moreover, there is an ambiguity in choosing the method of normalization and determining the distance between objects. The influence of the metric (DistanceFunction) is illustrated by the diagrams in Fig. 4. The results presented in this series are obtained by means of the corresponding software application included in the GeoBazaDannych from the Wolfram Mathematica, which allows different options for setting DistanceFunction (Possible settings for Method). In the Wolfram

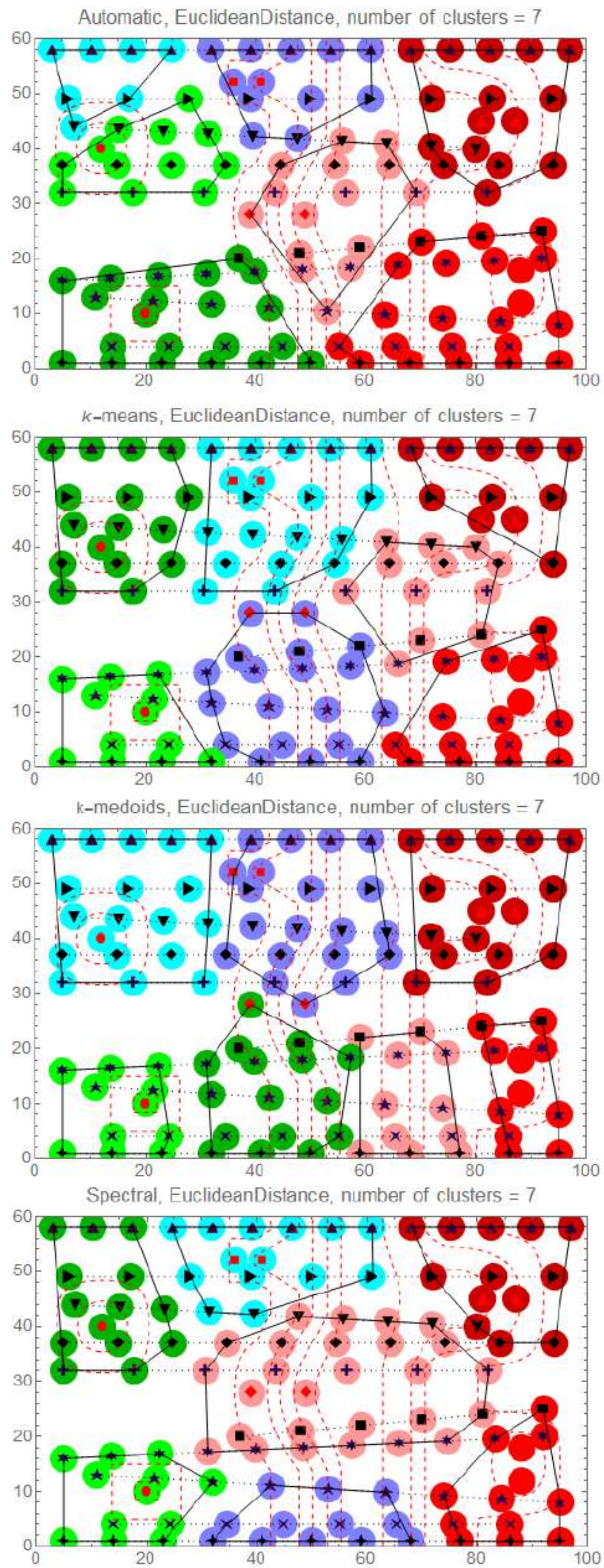


Figure 3. Clustering methods.

Mathematica system, different measures of distance or similarity are convenient for different types of analysis. The Wolfram Language provides built-in functions for many standard distance measures, as well as the capability to give a symbolic definition for an arbitrary measure. In particular, the following metric variant sare available for analyzing digital data [13]. The algorithmic features of the listed metrics can be clarified in the articles [14], [15]. As in the examples above, clustering algorithms were considered only for pairs of coordinates, i.e. the relative position of the points of the scattered set was taken into account, the Spectral method was used.

What methods of DistanceFunction are used in calculations is recorded in the headers of the schemes in Fig. 4. Representative variants are shown, namely ChessboardDistance [16], CosineDistance (a measure of similarity between two non-zero vectors of an inner product space), ChebyshevDistance (a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension), EuclideanDistance (the length of a line segment between the two points):

It follows from the above results that for the considered configuration of data points, taking into account the digital field of the original, clustering options using Spectral EuclideanDistance methods can be considered preferable.

VIII. Influence of the number of clusters

As noted above, one of the most important problems of segmentation is determining the number of clusters. The series of illustrations in Fig. 5 shows the results calculated by Spectral EuclideanDistance methods with the number of clusters 6 and 8; 7 clusters are shown in Fig. 4.

IX. The effect of accounting for values in points

In the results considered and shown in Fig. 3, Fig. 4 and Fig. 5, the similarity or difference between the classified objects is established depending on the metric distance between them. In other words – in the results presented in this series, the algorithms take into account not pairs *Xi*, *Yi*, but triples – *Xi*, *Yi*, *Zi*. Fig. 6 shows classification options using the Wolfram Mathematica ClusterClassify function (use data to create a function to classify new data into clusters), which allows clustering not only taking into account the coordinates of the points of the scattered set, but also the values in them.

From the above results, it follows that for the data class under consideration, taking into account the values at points does not give an additional positive effect in the implementation of clustering.

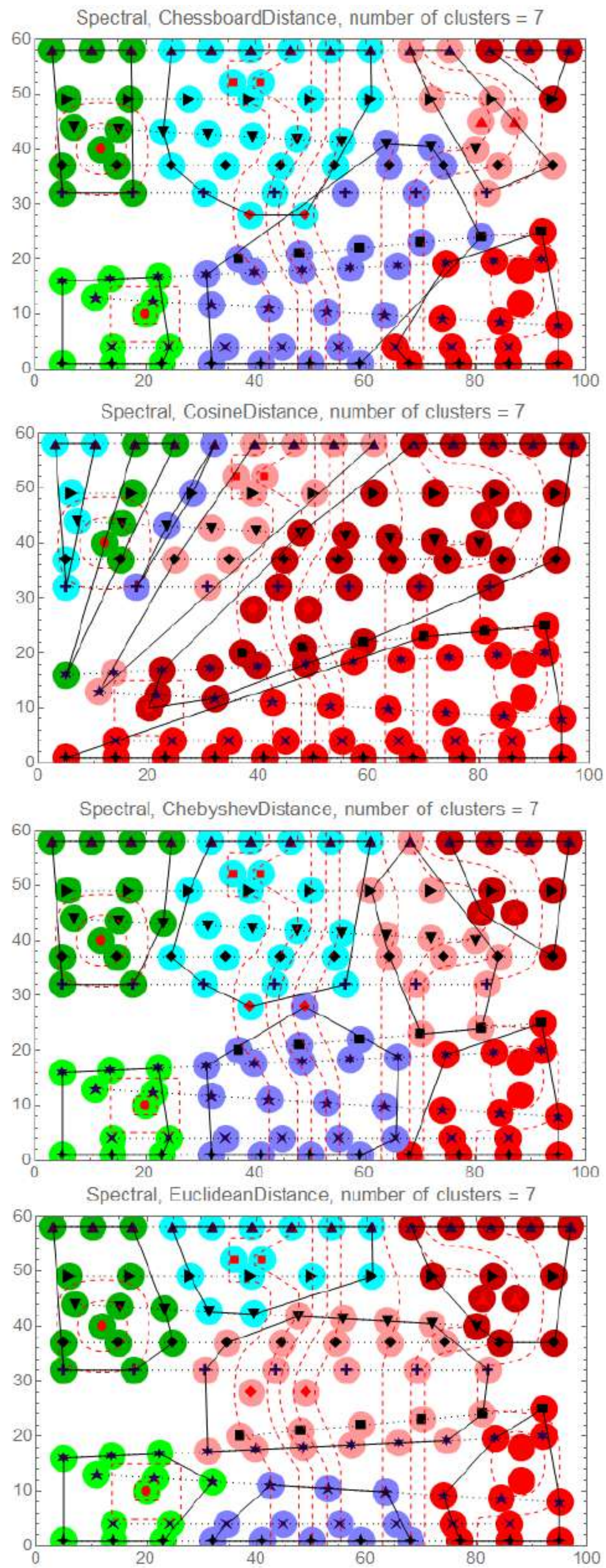


Figure 4. Influence of DistanceFunction.

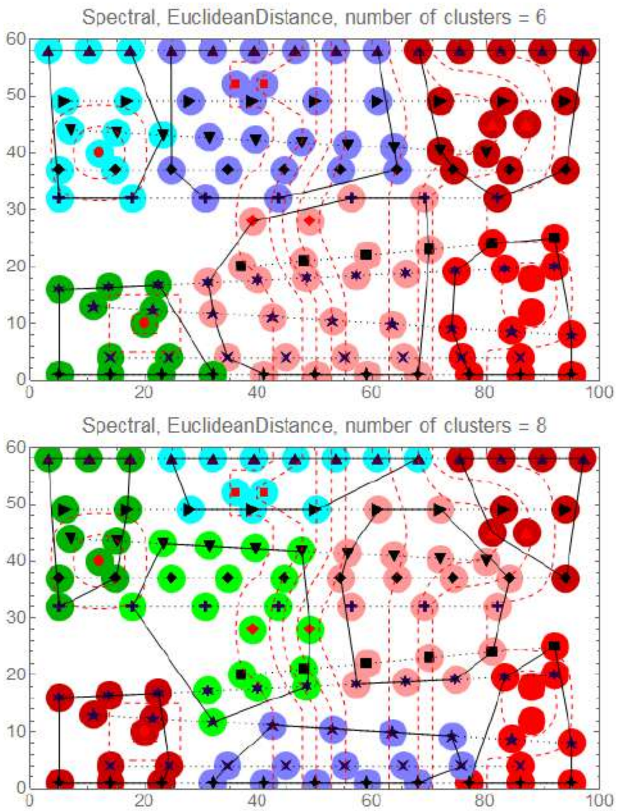


Figure 5. Influence of the number of clusters.

X. Conclusion

The article deals with the issues of instrumental filling and the use of the interactive computer system GeoBazaDannych. The results of clustering of a representative data set of a typical model of a geological object are presented and discussed.

References

[1] V. B. Taranchuk, Komp'yuternye modeli podzemnoi gidrodinamiki. Minsk: BGU, 2020. 235 p., (in Russian)

[2] V. B. Taranchuk, "Examples of the use of artificial neural networks in the analysis of geodata", Open semantic technologies for intelligent systems, vol. 3, pp. 225–230, 2019.

[3] V. Taranchuk, "Tools and examples of intelligent processing, visualization and interpretation of GEODATA", Modelling and Methods of Structural Analysis. IOP Conf. Series: Journal of Physics: Conf. Series Vol. 1425 (2020) 012160. – P. 9.

[4] V. B. Taranchuk, "Examples of intelligent adaptation of digital fields by means of the system GeoBazaDannych", Open Semantic Technologies for Intelligent Systems, vol. 4, pp. 243–248, 2020.

[5] V. Taranchuk, "Interactive Adaptation of Digital Fields in the System GeoBazaDannych", Communications in Computer and Information Science. Book series Springer, vol. 1282, 2020, pp. 222–233, https://doi.org/10.1007/978-3-030-60447-9_14.

[6] D. Tupper Charles, "Concepts of Clustering, Indexing, and Structures", Data Architecture, 2011, pp. 241–253, https://doi.org/10.1016/B978-0-12-385126-0.00013-9.

[7] B. S. Everitt, S. Landau, M. Leese, D. Stahl. Cluster Analysis. 5th Edition, John Wiley & Sons, 2011, 360 p.

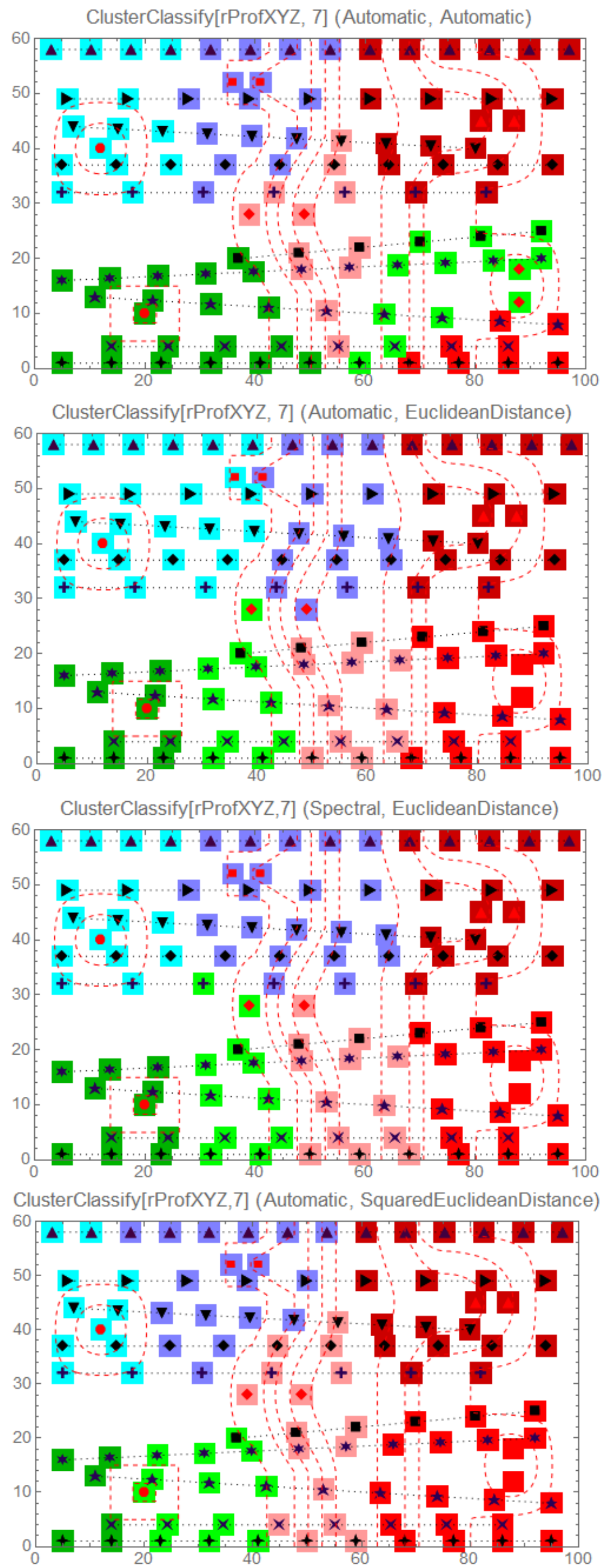


Figure 6. Influence of DistanceFunction.

[8] V. V. Sobchuk, O. V. Chichurin, I. V. Kalchuk , T. V. Zhigallo. Solving problems of analysis and Differential Equations by means of computer algebra Mathematica: textbook. Lutsk: Volyn National Academy of Sciences. Lesya Ukrainka University, 2021, 414 p., (in Ukrainian).
[9] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, Arthur Zimek, "Density-based Clustering", WIREs Data Mining and Knowledge Discovery, 2011, 1 (3): 231–240. doi:10.1002widm.30.
[10] H. Bock, "Clustering methods: a history of k-means algorithms", in: Selected contributions in data analysis and classification. Springer, Berlin, 2007, pp. 161–172.
[11] H. Park, C. Jun, "A simple and fast algorithm for k-medoids clustering", Expert Syst. Appl., 2009, 36(2), 3336–3341.
[12] U. von Luxburg, M. Belkin, O. Bousquet, "Consistency of Spectral Clustering", Annals of Statistics. 2008, 36(2), 555–586.
[13] Distance and Similarity Measures. https://reference.wolfram.com/language/guide/DistanceAndSimilarityMeasures, (accessed 2021, Jun).
[14] E. Amigó, J. Gonzalo, J. Artiles, et al., "A comparison of extrinsic clustering evaluation metrics based on formal constraints", Inf Retrieval 12, 2009, 461–486. https://doi.org/10.1007/s10791-008-9066-8.
[15] P. Grabusts, "The Choice of Metrics for Clustering Algorithms", Environment Technology. Resources, DOI:10.17770/etr 2011vol 11 70–76.
[16] W. Zhu, C. Ma, L. Xia and X. Li, "A Fast and Accurate Algorithm for Chessboard Corner Detection", 2009, pp. 1-5, doi: 10.1109/CISP.2009.5304332.

Интерактивные и интеллектуальные средства системы ГеоБазаДанных

В.Б. Таранчук

В статье рассматриваются вопросы инструментального наполнения и использования интерактивной компьютерной системы ГеоБазаДанных. Представлены и обсуждаются результаты кластеризации представительного набора данных типичной модели геологического объекта.

Received 31.05.2021

Complex Human-Machine Petroleum System Improve by Causation Approach

A.A. Bezrodny
Belarusian State University
Minsk, Belarus

A.M. Korolenok
Gubkin University
Moscow, Russia

A.F. Rezchikov
Trapeznikov Institute of Control Science
Moscow, Russia

***Abstract*—Most of modern systems are complex human-machine ones. To improve them one needs to use analysis, simulation, and optimization. One of the possible ways is to consider so-called inner languages of systems taking into consideration their nature and peculiarities. Causation approach in this case provides description of various origin element systems and decision-making in ill-defined situations. In the article below appropriate structure models and algorithms are presented for petroleum supply system as example.**

***Keywords*—algorithm, complex system, knowledge, causal-and-effect, petroleum supply**

I. Introduction

Complex human-machine systems (further – CHMS) contain automation and manpower control circuits with many elements and links between them and require scientific approach for improvement. They work under directions of parent systems, satisfy consumer needs, consider possibilities of competitors and suppliers, follow regulations, etc. These demands, influences, restrictions (factors) may be formalized as element sets that permit to create goal areas E_{ef} in parameter space R to which a system image trajectory should come.

For a large-scale distributed CHMSs task to improve is formulated as to bring key performance indicator (KPI) K to extremum at factors G in Δt by developing structures $S = \langle X, U, GR\rangle$ and selecting actions $\langle C, A, X, U\rangle$

$$K(S, \langle C, A, X, U\rangle, \Delta t) \to extr \qquad (1)$$

where X – set of control means and U – relations between them, C – control functions, A – algorithms, GR – structures. In given formulation the task usually could not be resolved because of diversity of components, non-linearity and difficulty of precise description in mentioned parameter spaces due to high dimension.

Petroleum supply system as an important energy source for transport [1] may be considered as CHMS good example. Existing methods to resolve practical tasks are not enough because of the system size and development, that causes needs to search new ones.

II. Causation approach

For systems with insufficient information about their structure and behavior (ill-defined ones) it is recommended [2] to use some knowledge languages, describing logics both in linguistic (semantical) form and good enough for simulation and common use.

The most important regularity of a system is historicity or development in time [3]. These ideas are formulated in all fields of knowledge and practical activity: For any objects, processes, events and phenomena (objects) previous ones exist as origins, causing changes and connecting with other objects, that is also true for subsequent objects for which considered is origin in its turn. They have both «casual future and past». Causality principle it is necessary to understand as a part of the Law of Nature to Time and Space [4] – finite speed of signal propagation, impossibility to influence on the past – that is also related to sequence of stages in lifecycle. But sometimes relative simplicity of cause-and-effects is the reason for «refusal» as from «earlier» is not exactly gone «because of» [5].

There are the following properties of cause-and-effect or casual interaction:

– «cause» assumes groups of objects (here better events) genetically inter-connected;
– there are no events without causes and effects, varied and interconnected,
– absolute causes and effects are absent or unachievable;
– goals are also derived from the structural causal definition [6].

As future states of a system goals are achieved at mentioned factors that may be named as condition(s) 1. Results (effects) of interaction changes the system and surroundings that forms new conditions (s) 2. Achieve of an elementary goal is modeled as follows:

- system element is at stage SA with factors GA, KPI K, resources WA under control CA;
- functions and algorithms, contained in the kernel of a causal cell, brought it from A to B;
- at SB with factors GB and control CB corrected accordingly to the goal achieving degree ($\|GA - -GB\|$) new K^* (after interaction) and output resources flow WB appear (Fig.1);
- decomposition of G, W, C, St is widely used and theoretically and practically approved.

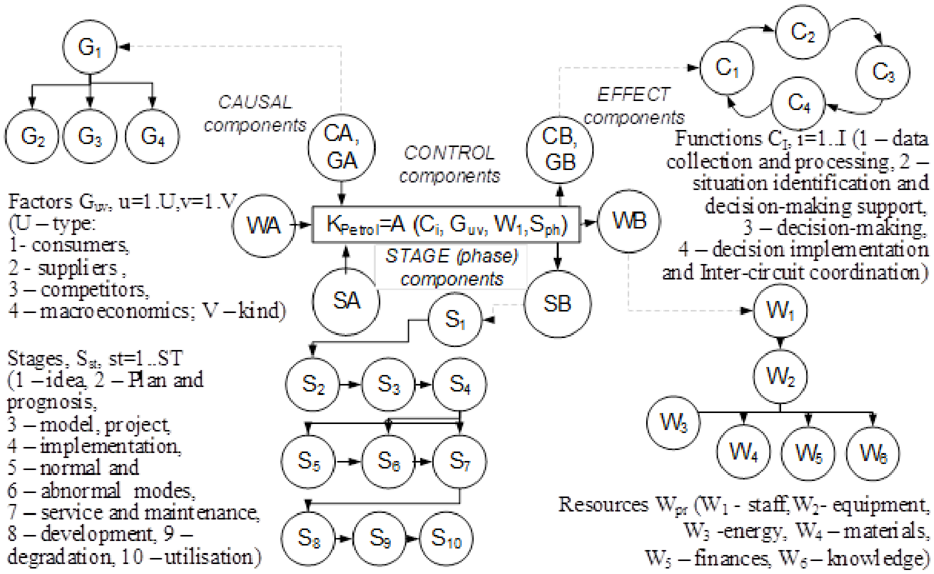


Figure 1. Structure of Cause-and-effect relation to choose development direction of petroleum supply

On Fig. 1 P_j processes (j=1..J), H_k control periods (k=1..K), X_{pq} – means (p=1..P – type, Q=1..Q – level), U_{pq} – relations, Ph_{ph} – phases (ph=1..PH).

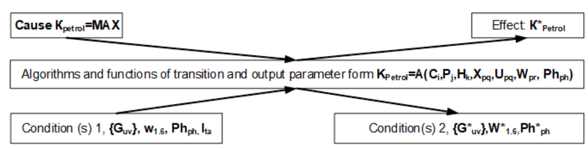
For non-elementary cases cause-and-effect (CE) complexes are formed using finite state machine algebra [7] and interaction simulated accordingly to sphere process models. To operate with the cells some set of operations OC (oc=1..OC) is formed: unification (model creation), decomposition (structuring), intersection/Cartesian products (multicircuit control), complements (extra- and interpolation of parameters for parts with most trustable data), composition (developing of the system model by process approach, etalon models) and substitution (synthesis optimal structures).

The solution in non-elementary situation is achieved by formulation of general, intuitively understandable or proved casual relations, its components decomposition to the level perceptible for involved using models of the sphere, practice checking, feedback and correction. If there is under-determined situation parts with the most trustable data are selected, for which same procedures are realized, CE-cells for the whole system are developed and optimization tasks are resolved considering known models with given accuracy of data. Results are step-by-step improved while the system is developed or new data appear, information is put in knowledge data base (DB). Since there are no restrictions on types of functions and algorithms, it is possible to describe various origin element system.

III. Algorithms and Diagrams to improve petroleum supply as complex system

Petroleum supply is supposed [1] consisting of stations to serve clients and terminals to distribute products, nets to provide the objects work and companies as legal entities.

Therefore, model of CE-cell on the first level of decomposition accordingly to the goal «profit satisfying clients» [8] may be formulated as achieving maximal K (profits/costs at alternatives) by control structure formation and selection of control actions, i.e.:



General algorithm to select development way of CHMSs is shown on Fig. 2, where:

- main factors * are determined by correlation analysis, specification the goals depends on factors G, flow chart of processes and characteristics X and U;
- KPI K comprises mentioned items for all hierarchical levels and sub-systems;
- petrol DB or at least investigation of the works done before should be;
- t^* - desirable time, within which the system should achieve the goal area;
- elementary control tasks F_{ijk} and circuits are Cartesian products in matrix form;
- $QG_n = K_2/K_1$ – thresholds between type of system stages (phases) at transition from time 1 to time 2 with ΔQG_n and $\Delta t_{dG,n}$ as deviation and t_{dG} – time of G_{uv} changes.

Identifying phase of the system (see Fig. 3) and appropriate sets of processes, structures and resources

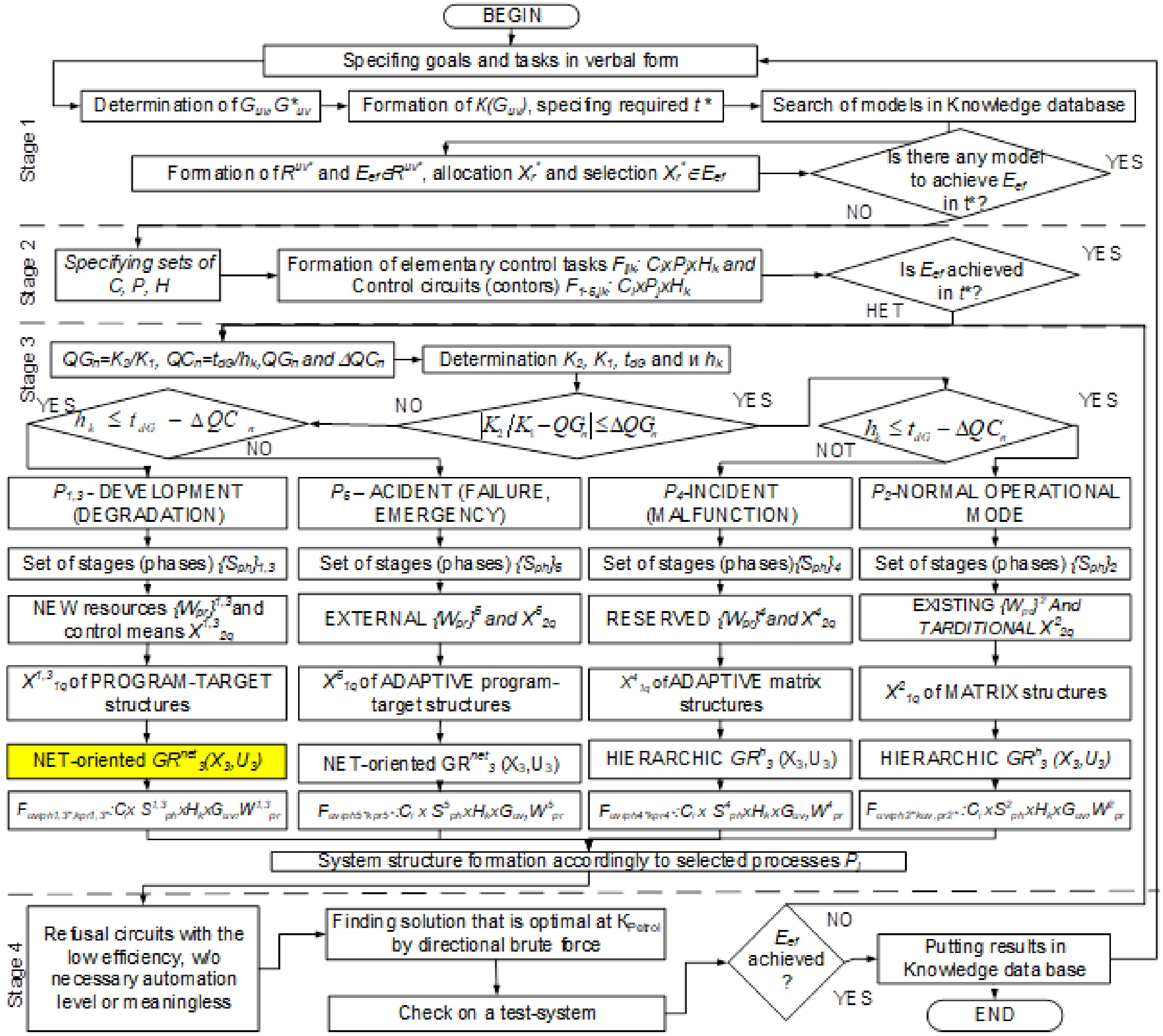


Figure 2. Informational-logic diagram to select CHMS development variant

elementary control tasks are determined as follows

$$P_{1,3} - F_{pqiphkpruv} : X^{1,3}_{pq} \times C_i \times S^{1,3}_{ph} \times H_k \times W^{1,3}_{pr} \times G_{uv},$$

$$P_2 - F_{pqiphkpruv} : X^{2}_{pq} \times C_i \times S^{2}_{ph} \times H_k \times W^{2}_{pr} \times G_{uv},$$

$$P_{4,5} - F_{pqiphkpruv} : X^{4,5}_{pq} \times C_i \times S^{4,5}_{ph} \times H_k \times W^{4,5}_{pr} \times G_{uv}.$$

On Fig. 3 figures on ribs mean:

- development/degradation - 1 – development control, 2 – transition between stable stages, 3 – feedback to control system (CS), 4 – corrections or corrective actions
- normal mode - 1 – influence of surroundings (at perturbation) or 1’ – system restrictions (at deviation), 2 – oscillation around stable stage, 3 – feedback to CS, 4 – correction;
- abnormal mode - 1 – influence of surroundings or 1’ – system restrictions, 2 – transition to an unstable stage, 3 – feedback to CS, 4 – correction, 5 – transition to an unstable stage (cascade), 6 – feedback to CS, 7 – correction, 8 – coming back or 8’ – system liquidation.

Control means X_{pq} act accordingly to principles and procedures during periods with relative costs, that structure presented in matrixes M_{1-4} respectively (M_1 for Pr_4 for short).

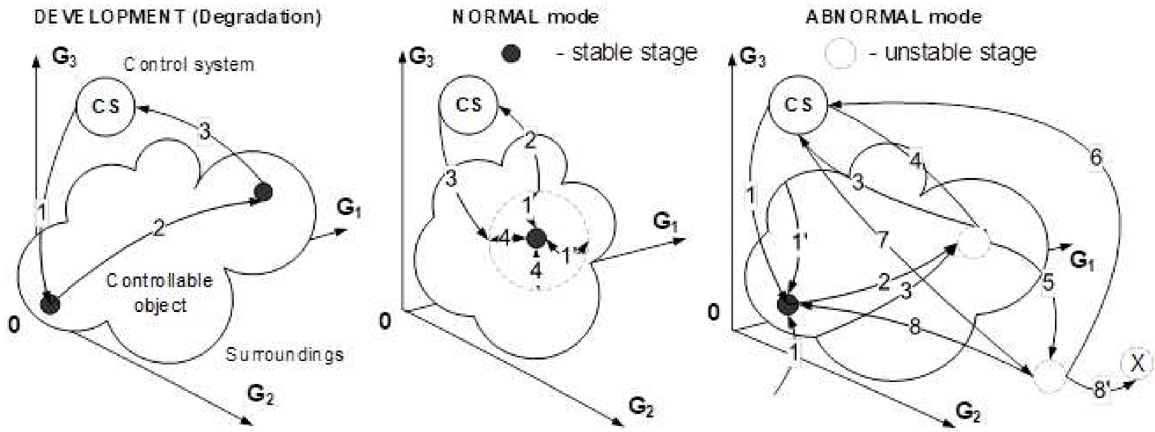


Figure 3. Phases of a system development

$$M_1 \quad \begin{array}{cc} & S_{ph=1} \\ \begin{array}{c} X_{11} \\ X_{12} \\ X_{13} \\ X_{21} \\ X_{22} \\ X_{23} \end{array} & \begin{pmatrix} - \\ - \\ - \\ 0{,}5 \\ 0{,}2 \\ 0{,}2 \end{pmatrix} \end{array} \quad \begin{array}{cc} & S_{ph=2} \\ \begin{array}{c} X_{11} \\ X_{12} \\ X_{13} \\ X_{21} \\ X_{22} \\ X_{23} \end{array} & \begin{pmatrix} 1 \\ 0{,}5 \\ 0{,}1 \\ 0{,}01 \\ 0{,}01 \\ 0{,}01 \end{pmatrix} \end{array}$$

$$M_2 \quad \begin{array}{cc} & \begin{array}{cccccc} X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & IS \end{array} \\ \begin{array}{c} X^C_{11} \\ X^C_{12} \\ X^C_{13} \\ X^C_{21} \\ X^C_{22} \\ X^C_{23} \end{array} & \begin{pmatrix} Pr_{3,4} & 0 & 0 & Pr_{2-4} & 0 & 0 \\ - & Pr_{2-4} & 0 & Pr_{2-4} & 0 & 0 \\ 0 & - & 0 & Pr_{2,3} & Pr_{3,4} & Pr_{1,2} \\ Pr_{2,4} & 0 & - & Pr_3 & Pr_3 & 0 \\ 0 & Pr_{1,2} & 0 & - & 0 & 0 \\ 0 & 0 & 0 & Pr_4 & - & Pr_{1,2,3} \end{pmatrix} \end{array}$$

$$M_3 \quad \begin{array}{cc} & \begin{array}{ccccc} h_1 & h_2 & h_3 & h_4 & h_5 \end{array} \\ \begin{array}{c} X_{11} \\ X_{12} \\ X_{13} \\ X_{21} \\ X_{22} \\ X_{23} \end{array} & \begin{pmatrix} 0{,}0 & 0{,}1/n & 0{,}2/3 & 0{,}3/3 & 0{,}4/3 \\ 0{,}05 & 0{,}2 & 0{,}3 & 0{,}3 & 0{,}15 \\ 0{,}33 & 0{,}33 & 0{,}33 & 0 & 0 \\ 0{,}1 & 0{,}1 & 0{,}2 & 0{,}4 & 0{,}2 \\ 0{,}1 & 0{,}2 & 0{,}4 & 0{,}2 & 0{,}1 \\ 0{,}5 & 0{,}2 & 0{,}1 & 0{,}1 & 0{,}1 \end{pmatrix} \end{array}$$

$$M_4 \quad \begin{array}{cc} & \begin{array}{cccccc} X_{12} & X_{13} & X_{21} & X_{22} & X_{23} & IS \end{array} \\ \begin{array}{c} X^C_{11} \\ X^C_{12} \\ X^C_{13} \\ X^C_{21} \\ X^C_{22} \\ X^C_{23} \end{array} & \begin{pmatrix} DPGS & P & DPGC & DP & DP & 0 \\ - & DPGS & 0 & DPGC & DP & P \\ 0 & - & 0 & DPGS & DP & DP \\ 0 & DPGC & - & DPGC & DP & 0 \\ 0 & 0 & 0 & - & DPGC & DP \\ 0 & 0 & 0 & 0 & - & DPGC \end{pmatrix}^{Pr_4} \end{array}$$

In M_1: principle D – control at system deviation, P – at perturbation from surroundings, GC – goal change by decision makers, the lower the level the more monitoring and following, the upper – analysis, and decision making accordingly to data collected (Pr_4).

In M_2 – procedure Pr_1 – continuous and Pr_2 – discrete monitoring, Pr_3 – quasiprogramming control, Pr_4 – control with collection (analysis) of data, C – controlling X_{pq}.

In M_3 H_k – control time periods (K=1..5, 1 – monitoring by X_{2q} or automation, 2 – interruptible monitoring, 3 – tactical, 4 – operative and 5 – strategic periods), 1 – full (0 – non-) participance of a mean (manager's time proportionally, others deal with only one).

In M_4 for X_{pq} at p=1 (human, q=1 – manager, 2 – deputy manager, 3 – specialist) and at p=2 (technical, q=1 – server, 2 – work station, 3 – controller), quantity of levels is example, IS – inactive system, Sph=1 – development and Sph=2 – operation.

In the adjacency and incident M_{1-4} «0» - absence of interaction and «-« - its impossibility, the higher level of X_{1q} the lower interaction with IS and X_{pq} at q=2 and 3. All of matrix elements are quantitively derived from data of real working objects, conditional and in particular are some parameters of the model also. The matrix multiplication is possible

$$M_1 \times M_2 \times M_3 \times M_4^{Sph=1,2}$$

The synthesis of control system structure by convolutions is done considering efficiency, meaning, level of automation:

- C-convolution (synthesis) as integration of control functions alongside control circuits and designation more C_i to the smaller number of X_{pq};
- P-convolution as the same for functions C_i belonging to various circuits of processes P_j;
- H-convolution as the same for control functions C_i on various time periods.

GR, GR_1 – GR_4 mention in (1) determine graphs of, correspondingly, infra-system (non-active and needed control), control (1), decision making (2), organization-technical (3) and information (4) systems. It is supposed [9] that the models are enough to describe a whole system. The task to form structures and choose control actions using system causal approach is resolved by the informational logic diagram on Fig. 4 [10].

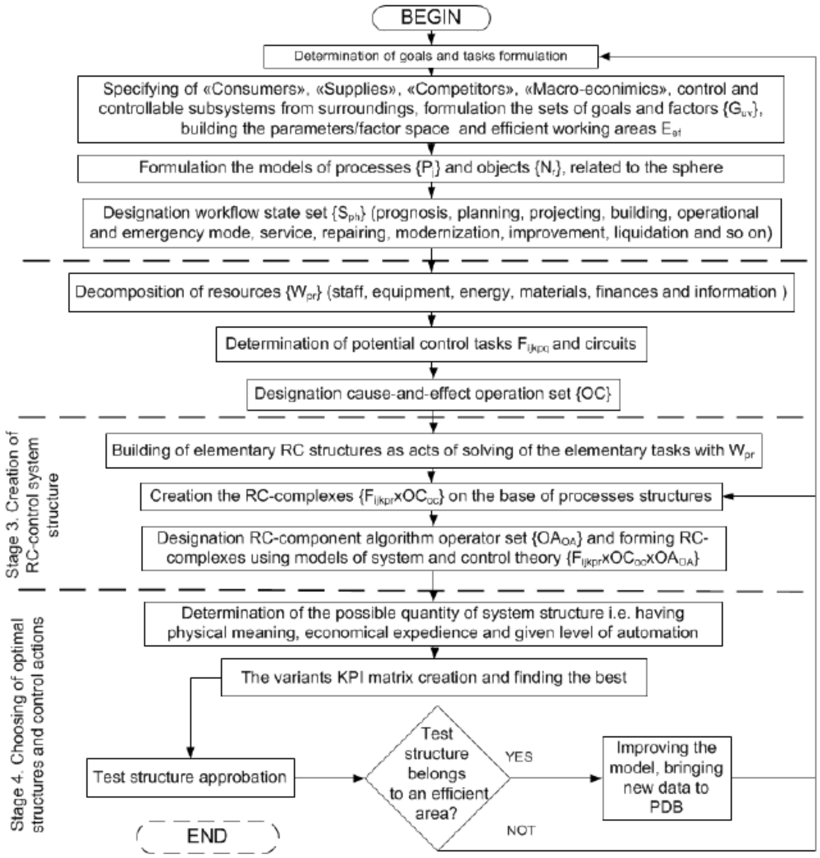


Figure 4. Information-logic diagram to form structure and choose control actions using causal approach

IV. Discussion

As a result of the approach application it was developed the complex of inter-related structure models and algorithms to impove petroleum supply systmes (Fig. 5). Normal font marks previousle obtained models, bold – newly ones, gray color – under investigation.

In 1998-2004, it was numerically proved multi-product dispenser station structure, created algorithm for service stations location at maximum transportation flow points on the basis of automation system for part with the most trustable information.

In 2005-11 there were found optimal station parameters for small and medium towns and inter-city roads (3-4 and 9-10 stations with no less than 5 cross-roads between them and every 75 km +/- 25 km, respectively). For minimum client queque and station downtime is the structure of two dispensers with all of the fuels proposed, outdoor payment terminals provide 10 % higher productivity considering necessity to enter customers inside the station building to sale non-fuel goods. Moreover there were prepared efficient system structures to security, automation, card and loyalty (3 regions, sales increace in 6 times), staff preparation (2 training center), etc [10].

In 2012-19 in some regions of Russia and CIS states there were changed technical maintanance systems with cost reduction in 3-15 % at better service. Also logistics automation control system (tank-trucks on-board gauges camera, GPS, smart-sealing) was developed and implemented. Finally, theoretical foundations of petroleum product traceability were developed with implementation in Turkey and tests in Sant-Peterburg [11].

V. Conclusion

The systematic causal-and-effect approach proposed to improve complex human-machine systems is characterized by co-synthesis of controllable and control systems, descision-making in case of not enough trustable data from systems and surroundings, possibility to match objects, processes, events and phenomena of various nature and so on.

For petroleum supply important for economy and requiring continuous improvement structure models and control algorithms were done with practical results achieved.

Adequatnes of models and algorithms and approach as whole is cofirmed by the proximity of the known and developed models on the similar feasible regions, reliability of results by statistical data for more than 20 years of observation, validity of conclusions by results

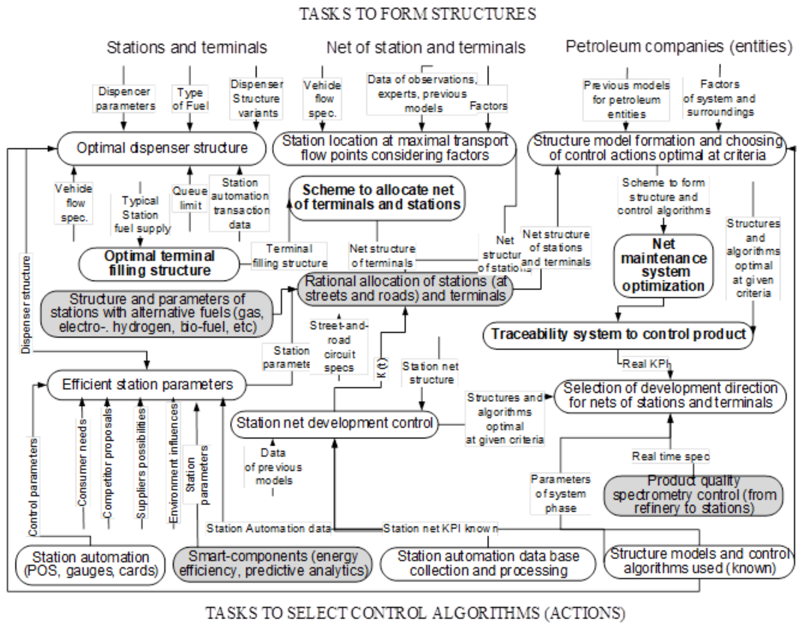


Figure 5. Interaction of structure models and control algorithms done to improve service nets

of approbation and successful applications that permit to use them in other spheres.

References

[1] F.A. Davletiyarov and others. *Nefteproductoobespechenie. [Petroleum Supply]*. Moscow: Matematika, 1998. 662 p (In Rus).

[2] Pospelov D.A. *Logiko-lingvisticheskie modeli v systemah upravleniya [Logico-linguistical models in control systems]*. Moscow: Energoizdat, 1981. 232 p. (In Rus).

[3] Forrester. J. Principles of systems. Cambridge: Wright Allen Press, 1960.

[4] Fok V.A. *Teoriya Prostranstva, vremeni i tyagoteniya [Theory of space, time and gravity]*. Moscow: Fizmatgiz, 1961. 568 p. (In Rus).

[5] Rezchikov A.F. and others. Problem of preventing crucial event combination development in large-scale systems. Advances in Intelligent Systems. 2020. T. 1224. P. 274-280.1.

[6] R. Acoff. On Purposeful Systems: An Interdisciplinary Analysis of Individual and Social Behavior as a System of Purposeful Events, Aldine-Atherton: Chicago. 1972, 272 p.

[7] Rezchikov A.F., Tverdokhlebov V.A. *Prichinno-sledstvennie modeli proizvodstvennih system [Causal models of industrial systems]*. Saratov: «Scientific book» Publishing house, 2008. 137 p. (In. Rus).

[8] Kotler, Philip. Principles of marketing / Philip Kotler, Gary Armstrong. 9^{th} ed. 496 p.

[9] Tsvirkun A.D. *Struktura sloznih system [Structure of complex systems]*. Moscow: Nauka, 1981. 345 p. (In Rus).

[10] Alexey Anatolievitch Bezrodny, Anatoliy Mikchailovitch Korolenok, Aleksandr Fedorovitch Rezchikov, Algorithms of Cause-and-effect Approach to Increase Service Net Efficiency, Engineering and Applied Sciences. Vol. 5, No. 2, 2020, pp. 34-40. doi: 10.11648/j.eas.20200502.11.

[11] Bezrodny A. A., Tszin V., Yunushev R. R., Korolenok A. M. Postroenie system proslezivaemosti v setyah AZS i neftebaz [Traceability system formation in petroleum supply]. *Nauka i tehnologii truboprovodnogo transporta nefti i nefteproduktov [Oil and Oil Products Pipeline Transportation]*. 2020;10(6):642-653 (In Rus).

Улучшение сложной человеко-машинной нефтяной системы с помощью причинно-следственного подхода

А.А. Безродный, А.М. Короленок, А.Ф. Резчиков

Большинство современных систем представляют собой сложные человеко-машинные системы. Для их улучшения необходимо использовать анализ, моделирование и оптимизацию. Один из возможных способов – рассмотрение так называемых внутренних языков систем с учетом их природы и особенностей. Причинно-следственный подход в этом случае обеспечивает описание различных систем элементов происхождения и принятия решений в нечетко определенных ситуациях. В данной статье в качестве примера представлены соответствующие структурные модели и алгоритмы для системы нефтесервиса.

Received 01.06.2021

Hybrid artificial neural networks for component design of space telemetry processing systems

Alexander Doudkin
United Institute of Informatics Problems of the National Academy of Sciences of Belarus
Minsk, Belarus
doudkin@lsi.bas-net.by

Yauheni Marushko
United Institute of Informatics Problems of the National Academy of Sciences of Belarus
Minsk, Belarus
marushkoee@gmail.com

Sergey Zolotoy
Unitary Enterprise Geoinformation Systems of the National Academy of Sciences of Belarus
Minsk, Belarus
gis@gis.by

Xiangtao Zheng
The Key Laboratory of Spectral Imaging Technology Xi'an Institute of Optics and Precision Mechanics Chinese Academy of Sciences
Shaanxi Province, China
xiangtaoz@gmail.com

***Abstract*—This paper describes a software system of neural network control of space telemetry data for malfunction diagnosis of spacecraft subsystems. This system is used for testing of intelligent technologies for processing information about a spacecraft subsystems state, prediction and detection of irregularities of the spacecraft subsystem modes. The information obtained from on-board data sources on space communication channel is used for processing.**

***Keywords*—neural network, telemetry, spacecraft, diagnosis.**

I. Introduction

To facilitate the development of intelligent systems (ISs), it is important to ensure the reusability of ISs components. A number of design support software is known, such as SSADM, Meris, TDD, Gherkin [1]–[3]. It allows to create component-oriented, service-oriented data processing systems. Their main drawback is the lack of flexibility of the designed tools in the sense of their dynamic reconfiguration in the face of changing requirements during the software life cycle [4], [5]. The solution of this problem lies in the use of components that have the property of adapting to changing operating conditions. Their development is provided by technologies based on the ontological approach to design, on the representation of the design process by semantic networks, in particular, open semantic technologies [6]–[9]. The presence of knowledge integration tools in these development technologies reduces the process of developing a new, more advanced IS to teaching the existing one.

It is shown that neural networks (NN), which are one of the most powerful and dynamically developing tools for intelligent information processing, can be effectively used as components of applied systems. Through training, NNs allow take into account the characteristics of specific data processing components. There are a number of examples of using NNs in onboard intelligent decision support systems for controlling a complex dynamic object and diagnosing its state [10]–[14]. Their main advantage is provided by their machine learning and self-learning ability, as well as by their high degree of parallelization of processing [15].

This paper solves the problem of applying a NN approach to construct systems for operational monitoring and assessing the state of spacecraft (SC) for remote sensing of the Earth (ERS) during their operation on Earth's orbit. The input data for processing are telemetry (TM) ones, which include measurements of physical quantities characterizing the position of the SC, environmental parameters, the state of the SC equipment, subsystems and processes, transmission of the results of these measurements, registration and processing of the received TM data in flight control centers.

The complexity of the TM analysis consists in the processing quantities that are heterogeneous in physical nature and dynamic characteristics and introduce a certain uncertainty in decision-making. Thus, new monitoring methods are needed that can detect anomalies in TM data. The increased complexity of on-board systems, processing and analysis of TM data in a continuous process accompanied by noise in the information flow by non-deterministic sources of interference leads to the fact that the existing deterministic control algorithms do not provide reliable identification of abnormal modes due to partial loss of diagnostic information. The NN acts here as an apparatus for formalizing complex algorithms for information transformation. An increase in the accuracy of the analysis can be obtained by taking into account features of analyzed objects and subsystems in the NN structure. The best way to do this is to develop of hybrid (combined) NN architectures or to build ensembles of NNs (ENN).

Considering the above, some tasks are relevant. Firstly, it is necessary to develop NN models that increase the accuracy of identification and prediction of the states of subsystems of spacecraft. Secondly, it is necessary to develop a technique for TM analysis using models based on ENNs trained for individual modes of operation of SC subsystems, which ensures processing of the entire set of TM parameters of the SC, with the possibility of additional training in case of work in a non-stationary environment. The software implementation of the technique will reduce the cost of monitoring the state and behavior of the SC subsystems, since this ensures the effective use of software and hardware to solve the problem of increasing its survivability due to the rational planning of TM sessions.

II. TELEMETRY SYSTEMS

Space telemetry is a set of technologies that makes it possible to measure physical quantities characterizing the state of objects or processes, transfer the results of these measurements, register and process the received data. A telemetry system (TMS) is a part of the command and measurement systems of flight control centers.

The nature and volume of measurements are determined by the tasks of the spacecraft and can be single, constant, periodic, as required by the measurement program. The paper considers the TM of a small ERS spacecraft of the Canopus type. The Belarusian spacecraft (BSC) also belongs to this type [16].

Measurement data is transmitted to Earth, consumed locally, or both, depending on the situation.

A domestic space TM practices two-level measurements and cyclic polling of sensors, determined by the measurement program. Part of the data without any onboard processing is completely transferred to Earth, where it is processed. The other part is processed on site.

A target TM represents data from scientific equipment and remote sensing means. In the ERS spacecraft, these are photo and video cameras and spectrometers.

An important characteristic of any space system, especially its orbital part, is reliability and resilience to failures and abnormal situations and the possibility of recovery.

The diagnostic task is to recognize the operating modes based on TM signals from the corresponding sensors. The set of recognized modes forms a set of recognizable classes. The signals are very noisy. Not the original signals of the recording equipment are fed at the NN input, but they are processed (filtered) and used in the form of a vector. The output can be also a vector specifying the probability distribution over the modes of operation of the subsystems, or the number of the most probable mode.

III. TMS ARCHITECTURE

The main conceptual characteristic of NN TMS (fig. 1) is learning ability and adaptation to various TM conditions based on simulation.

The interaction subsystem is designed to collect TM data from sensors, video and cameras, telescopes, etc., as well as data on their state, and transmit control commands.

The sensor readings analysis subsystem analyzes the state of the sensors and transmits the analysis result to the control subsystem, the hardware diagnostics subsystem and the data preprocessing subsystem.

The hardware diagnostics subsystem analyzes the current state of the sensors taking into account the existing state space. It trains and extracts knowledge about possible states of equipment and identifies emergency situations based on data on the current state of sensors and on the state space. The diagnostic result is transmitted to the control subsystem.

The data preprocessing subsystem filters and removes data redundancy.

The data storage subsystem is designed to store TM data and descriptions of all possible states.

Intellectual data processing subsystem performs NN data processing.

The packet assembly subsystem fetches TM data from the database, forms and transmits packets to the data transmit/receive subsystem.

The current state transferring subsystem prepares data on the current state of the system and its sensors.

The data transmit/receive subsystem directly interacts with the radio channel, while transmission and reception can be carried out both through a communication channel (analogue of the TCP protocol) and in the form of datagrams (analogue of the UDP protocol).

The control subsystem is designed to collect and analyze data on the state of various subsystems, as well as to generate control signals.

I. BASIC ALGORITHMS FOR INTELLIGENT PROCESSING OF TM DATA

The intelligent processing subsystem consists of the following functional blocks:

1) NN block for identifying the state of the SC subsystems.
2) NN block for predicting the state of subsystems.
3) NN block for diagnostics of SC subsystems that is designed to monitor the performance of various SC subsystems.

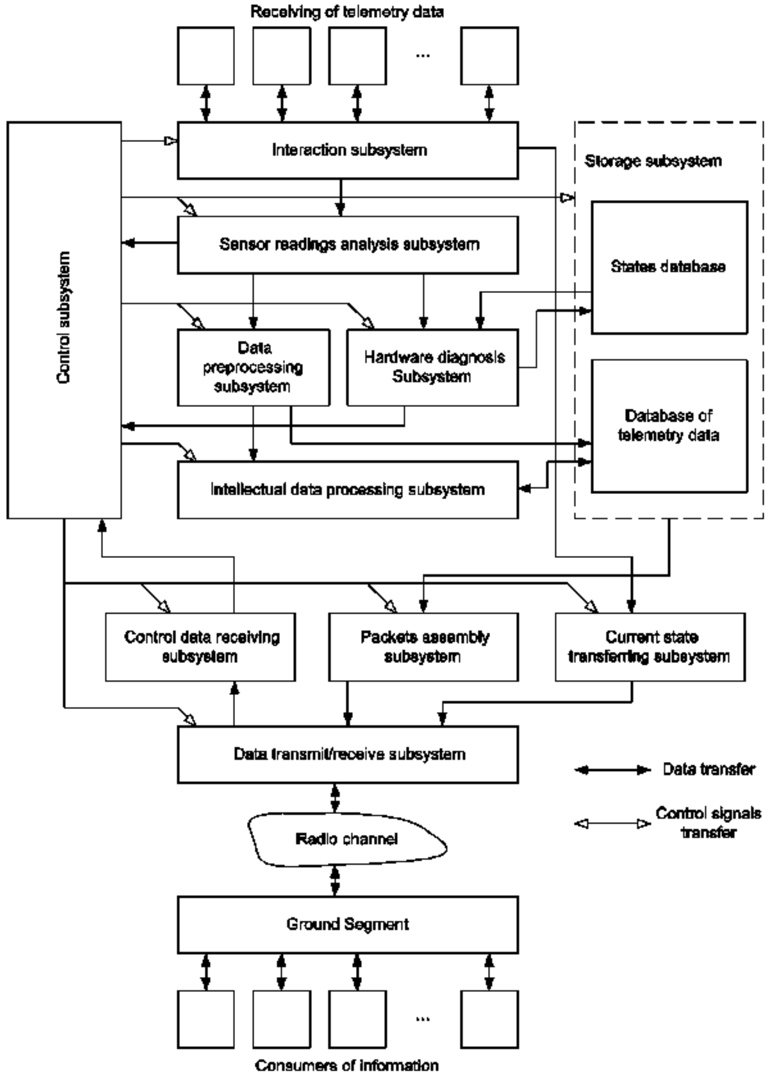


Figure 1. Block diagram of TMS

Onboard TM is a time series by its nature, including those ones with switching dynamics. A specific subsystem is characterized by a set of sensors of different types, differing in time reports (measurement frequencies), signal type, amplitudes, i.e. the time series are multidimensional. Samples of this series characterize the state of the research object at certain points in time and represent it in the space of measured features as continuous or quasi-continuous trajectories.

The type of NN depends on the types of processed signals, which are determined in turn by the specific SC subsystem and its constituent components, and the type of the problem being solved. These features are taken into account when constructing the NN model, along with taking into account the uncertainty and incompleteness of the initial information.

As a rule, the NN block consists of two parts:

- data preprocessing, which forms the input vector of NN;
- modules for constructing and training NN, recognition (calculating the output vectors of the trained NN), saving training samples in the database. Hybrid NNs are used as basic NNs.

Hybrid information processing technology involves a combination of traditional, NN and other intelligent processing methods that allow creating effective systems for processing complex structured data, when the use of only one NN method does not allow taking into account all the processing features.

Research in the field of improving the efficiency of identification and recognition based on the NN theory is carried out in the following two main areas:

1) Development of a unique, most suitable multi-layer hybrid NN model, which combines some popular NN models to effectively solve a real-life machining process. A hybrid NN model can be built from

at least two different types of NNs. The first part of the architecture is intended for preprocessing data and extracting informative features, the second is intended for making a decision in accordance with the problem being solved (segmentation, identification, classification, forecast, etc.). Various combinations of NNs are known for this purpose: multilayer perceptron, convolutional neural network (CNN), self-organizing map, long-term memory, support vector machine (SVM), recurrent NN, etc. [13], [14], [20], [21].

2) Development of ENN. These are sets of NNs that make decisions by averaging the results of individual NNs that improve the quality of identification [21]–[23]. The basic models for ENN are heterogeneous or hybrid NNs. They can be built from at least two different types of NNs.

Next, let's consider the developed base ENNs.

II. TWO-LEVEL ENN MODEL FOR PROCESSING MULTIVARIATE TELEMETRY TIME SERIES

The size N_I of the NN input layer of one ENN is determined as the product of the number of sensors in the subsystem and the time window. The size N_O of the output layer is the number of sensors in the subsystem. The size N_H of the NN hidden layer of one ENN is established when conducting an experiment with the procedure for finding the suboptimal size of the hidden layer of single NNs. Learning is carried out by the RPROP algorithm [24]. The ENN output value is generated as weighted sum of the outputs of individual NNs [25]. The weighting is repeated after specified interval of the processed time readings (dynamic weighting).

An ensemble of experts trained step-by-step on the input data (without access to the previous data) combined with a form of weighted voting for obtaining the final solution is the common of the algorithms of the drift detecting [26]–[28]. Incremental training of ENN means estimating of accuracy of all models and their ranging by accuracy at each forecasting iteration. When the error of ENN increases, the drift of the target variable is detected and a new element trained at the relevant data is added the ensemble. In this approach, we retain the model put in during the initial training and add new parameters without the problem of "forgetting". This is the way of additional training of ENN.

We solved the forecasting problem of TM data for three subsystems of the BSC. They are the power system (PS), the corrective propulsion system (CPS) and the target equipment (TE). Consequently, we generated three ENN for the TM data processing. The structure of the two-level organization of TM data processing is shown in Fig. 2.

Preprocessed TM data and the identifier of the subsystem are fed to the ENN input, which is delivered to the supervisor of the ENNs with the aid of the communication block. The supervisor generates a signal for choosing of ENN for the given subsystem and initiates the procedure of its additional training. The incremental block of additional training is responsible for preparation of the training data set and training of new elements of ENN.

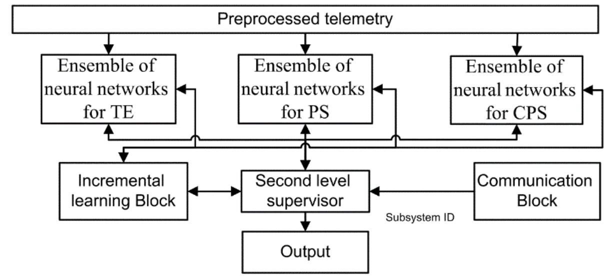


Figure 2. General scheme of telemetry data processing using ENN

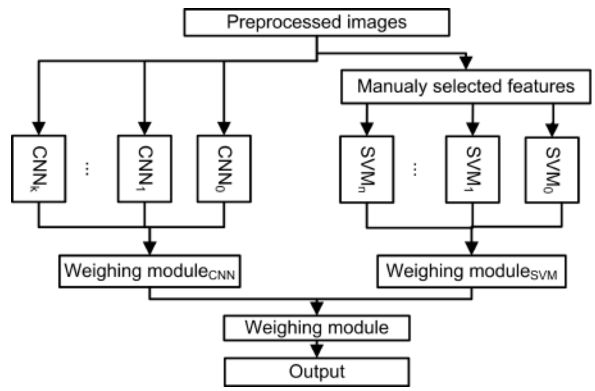


Figure 3. Ensemble of CNN and SVM models

The proposed organization of ENN in two levels implements the heterogeneity of the NN system, where the first level of the structure is represented by a set of ensembles of heterogeneous networks, and the second is represented by one generalizing module. An ensemble or a single supervisor network that processes the output values of all elements of the first level can be used as a second-level expert.

I. I. AN ENSEMBLE OF CONVOLUTIONAL NEURAL NETWORKS AND SVM MODELS

The method of support vectors is recommended to be used when working with a small set of features, so it can be chosen as the main method when forming a model from manually selected features of objects.

Thus, CNNs that receive input data directly in the form of images, and a set of SVMs that make decisions on selected features of objects can be combined into an ensemble (Fig.3).

This scheme can be modified by submitting additional features, formed without using images, directly to the input of the SVM classifiers.

CNN can be modified similarly. The network can be divided into several branches for data processing (Fig.4).

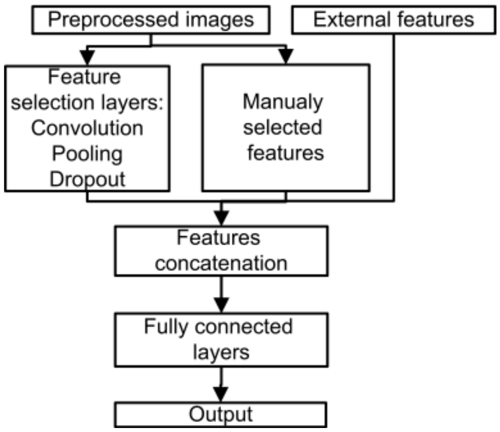


Figure 4. Hybrid CNN

One branch performs automatic feature extraction on the image using standard CNN layers, the weighting coefficients of which are determined by gradient methods during training. The other branch may include a set of predetermined preprocessing procedures, and for each input image to form an additional set of features. Also, sets of external features can be submitted to the hybrid model. This model involves two stages of training.

At the first stage, the first branch of the network is trained until sufficient accuracy is achieved, or before stopping by early stopping methods.

At the second stage, the weights of the convolutional layers of the network are fixed and the training is carried out only for fully connected layers, for which features from convolutional layers, a manual set of features, and external features come together.

CONCLUSION

A two-level model of ENNs for processing multidimensional time series of telemetry of SC subsystems is described. The experimental prototype of the software NN system was developed at the UIIP NAS of Belarus.

The proposed component design technology can be effectively supported by the OSTIS technology and basic ontology technologies for control and monitoring to describe the subject area associated with data collection using sensors and the observation (data collection) process, for example, SSN, M3, OntoSensor [29]–[32].

REFERENCES

[1] Amodeo, Learning Behavior-driven Development with javascript, Birmingham, Birmingham:Packt Publishing, 2015. 392 p.

[2] Bek, Ekstremal'noe programmirovanie: razrabotka cherez testirovanie. Biblioteka programmista, SPb, Piter, 2003. 224 p.

[3] Smart, BDD in Action: Behavior-driven development for the whole software lifecycle,NY , Manning Publications, Shelter Island,.2015. 384 p.

[4] Lavrishcheva, Software Engineering komp'yuternykh sistem. Paradigmy, tekhnologii i CASE-sredstva programmirovaniya, Kiev, Nauk. dumka. 2013. 283 p.

[5] Chainikov, Solodovnikov, Informatsionnaya tekhnologiya sinteza struktury problemno-orientirovannykh programmnykh kompleksov, Avtomatizirovannye sistemy upravleniya i pribory avtomatiki Vseukrainskii mezhvedomstvennyi nauchno-tekhn. Sbornik, 2015, no 173, pp. 24-31.

[6] Golenkov, Gulyakina, Semanticheskaya tekhnologiya komponentnogo proektirovaniya sistem, upravlyaemykh znaniyami, Mezhdunarodnaya nauchn.-tekhn. Konferentsii OSTIS, Minsk, Respublika Belarus', BGUIR, 2015.

[7] Shunkevich, Metodika komponentnogo proektirovaniya sistem, upravlyaemykh znaniyami, mezhdunarod. nauchn.-tekhn. konferentsii OSTIS,2015:Minsk, Respublika Belarus', BGUIR, 2015.

[8] Bukhanovskii, Ontologicheskaya sistema znanii i vychislitel'nykh resursov sovremennykh intellektual'nykh tekhnologii, Ontologiya proektirovaniya, 2020, vol. 10, no 35, pp.22-33.

[9] Ivashenko, Tatur Printsipy platformennoi nezavisimosti i platformennoi realizatsii OSTIS, mezhdunarod. nauchn.-tekhn. konferentsii OSTIS-2016, Minsk, 2016.

[10] Emel'yanova, Neirosetevaya sistema kontrolya datchikov uglov orientatsii i dal'nosti kosmicheskogo apparata, Programmnye sistemy: teoriya i prilozheniya,2010, no 1, pp. 45–59.

[11] Khachumov, Review of Standards and the concept of monitoring, control and diagnostics of the spacecraft tools, Software Systems: Theory and Applications, Vol. 6, no 3(26), 2015, pp. 21–43.

[12] Emelyanov, Pogodin, Neural orientation angles and distance of the spacecraft sensor control system, Software Systems: Theory and Applications, 2010, no 1(1), pp. 45-59.

[13] Kim, Choi, Jeon, Liu, A hybrid neural network model for power demand forecasting, Energies, 2019, no 12(5), p. 931.

[14] Ma, Du, Cao, Analysis of multi-types of flow features based on hybrid neural network for improving network anomaly detection, IEEE Access, 2019, vol. 7, pp. 148363-148380.

[15] Golovko, Integratsiya iskusstvennykh neironnykh setei s bazami znanii, Ontologiya proektirovaniya, 2018, vol. 8, no. 3(29), pp. 366 - 386.

[16] Vityaz, The formation of unified scientific and technological space in the Union State of Russia and Belarus within the union programs, Economical and social changes: facts, trends, forecast, 2011, 1 (13), pp. 38-42.

[17] Ganchenko, Dudkin, Inyutin, Marushko, Podenok, Programmnaya neirosetevaya sistema kontrolya kosmicheskoi telemetrii, Iskusstvennyi intellekt, 2013, no. 4, pp. 502–511, Available at: www.vscc.ac.ru/files/journal/issues/esc-2011-1-13-215f971992–en.pdf (accessed 2021, Jul).

[18] Radio Frequency and Modulation Systems–Part 1: Earth Stations and Spacecraft. Recommendation for 13, Space Data System Standards, CCSDS, Washington, Blue Book. Issue 17. Washington, D.C.: CCSDS, 2006, 208 p.

[19] Morrison, «EA IFF 85» Standard for Interchange Format Files, Electronic Arts, Available at: http://www.martinreddy.net/gfx/2d/IFF.txt (accessed 2021, Jul).

[20] Frankel, Tachida, Jones, Prediction of the evolution of the stress field of polycrystals undergoing elastic-plastic deformation with a hybrid neural network model, Machine Learning: Science and Technology, 2020, no 1(3), pp. 035005.

[21] Liu, Yang, Wang, Zhang, A hybrid neural network model for short-term wind speed forecasting based on decomposition, multi-learner ensemble, and adaptive multiple error corrections, Renewable Energy, 2021, no 165, pp. 573-594.

[22] Berkhahn, Fuchs, Neuweiler, An ensemble neural network model for real-time prediction of urban floods, Journal of hydrology, 2019, no 575, pp. 743-754.

[23] Cheng, Wu, Tao, Mei, Mao, Cheng, Random cropping ensemble neural network for image classification in a robotic arm grasping system. IEEE Transactions on Instrumentation and Measurement, 2020, no 69(9), pp. 6795-6806.

[24] Riedmiller, Braun, A direct adaptive method for faster backpropagation learning: The RPROP algorithm, IEEE International

Conference on Neural Networks (ICNN), San Francisco, 1993, pp. 586–591.
[25] Marushko, Methods of Using Ensembles of Heterogeneous Models to Identify Remote Sensing Objects, Pattern Recognition and Image Analysis, 2020, Vol. 30, No. 2, pp. 217–223.
[26] Elwell, Polikar, Incremental Learning of Variable Rate Concept Drift, MCS. of Lecture Notes in Computer Science, 2009, vol. 5519, pp. 142–151.
[27] Chernodub, Novitskii, Dzyuba, Prognozirovanie vremennykh ryadov na osnove odinochnykh neironnykh setei i komitetov neironnykh setei: sravnitel'nyi eksperiment, Nauchnaya sessiya MIFI. Sbornik nauchnykh trudov, 2011, vol. 2, pp. 192–201.
[28] Parikh, Polikar, An ensemble-based incremental learning approach to data fusion, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, 2007, vol. 37, no 2, pp. 437–450.
[29] Gyrard, Bonnet, Boudaoud, Enrich Machine-to-Machine Data with Semantic Web Technologies for Cross-Domain Applications, IEEE World Forum on Internet of Things (WF-IoT), 2014.
[30] [SSN, 2019] Semantic Sensor Network Ontology Available at: https://www.w3.org/2005/Incubator/ssn/ssnx/ssn (accessed 2021, Jul).
[31] Stavropoulos, Vrakas, Vlachava, Bassiliades,. Bonsai: A smart building ontology for ambient intelligence, 2Nd International Conference on Web Intelligence, Mining and Semantics, USA, 2012.
[32] Xue, Liu, Zeng, Yu, Shi, An Ontology based Scheme for Sensor Description in Context Awareness System, IEEE International Conference on Information and Automation, 2015.

Гибридные искусственные нейронные сети для компонентного проектирования систем обработки космической телеметрии

А.А. Дудкин, Е.Е. Марушко, С.А. Золотой, С. Чен

В статье описывается программная нейросетевая система контроля телеметрической информации для диагностики подсистем космических аппаратов. Предназначена для отработки интеллектуальных технологий обработки информации, поступающей по космическому каналу связи от бортовых источников данных о состоянии подсистем космических аппаратов, предсказания и обнаружения нарушений штатных режимов функционирования бортовых подсистем.

Описывается двухуровневая модель ансамблей нейронных сетей для обработки многомерных временных рядов телеметрии подсистем космических аппаратов. Входными данными для обработки являются измерения физических величин, характеризующих состояние аппаратуры, подсистем и процессов положение космического аппарата, параметры внешней среды, передачу результатов этих измерений, регистрацию и обработку полученных данных в центрах управления полетами. Предлагается также гибридная сверточная нейронная сеть, которая комбинирует признаки, выделенные нейронной сетью и экспертами. Оптимальные значения гиперпараметров моделей вычисляются методами сеточего поиска с использованием k-кратной перекрестной проверки. Представлена структура телеметрической системы. Предложена технология компонентного проектирования, которая может эффектифно поддержана технологией ОСТИС и базовыми технологиями онтологий для для описания и мониторинга предметной области, связанной со сбором данных с помощью датчиков и процессом наблюдения (сбора данных).

Received 24.06.2021

Software for the design of optical control systems for the manufacture of precision microelectronics equipment

A. Doudkin, A. Voronov,
V. Ganchenko
United Institute of Informatics Problems of the National Academy of Sciences of Belarus
Minsk, Belarus
doudkin@lsi.bas-net.by, voronov@lsi.bas-net.by,
ganchenko@lsi.bas-net.by

S. Avakov
Planar Joint Stock Company
Minsk, Belarus
office@kbtem-omo.by

***Abstract*—This paper describes the main functions and architecture of software control system for inspection equipment of integrated circuit on basis of computer vision. Advantages of the developed architecture are described, as well as its application for image processing of integrated circuit layouts. The system allows identifying effectively defects what it is especially important for Very large-scale integration manufacturing based on submicron technology.**

***Keywords*—image processing, VLSI, automatic layout inspection, control and measurement equipment**

I. INTRODUCTION

Contemporary means for designing electronic circuits are aimed at shortening the time required for developing and launching new products into production and reducing the cost of the mass production of digital equipment. This is possible due to computer vision systems, which are a constituent part of the modern technology of the designing and producing integrated circuits (ICs). The transition to submicron design standards and the complication of the actual circuits necessitate the development of novel approaches, methods, and algorithms for the digital processing of IC images taking into account the design and technological limitations (DTLs) in IC manufacturing.

Currently, research in the field of IC manufacturing automation is carried out in the following main directions: developing optoelectronic conversion methods, improving the pattern recognition quality [1] – [7], and designing special inspection methods [1], [8]. The principal problems for increasing the efficiency of designing and manufacturing ICs using the planar technology are as follows [9]:

- development of a unified approach to digital transformations of images of layouts;
- application of the theory of artificial neural networks (NNs) in IC data processing during the development and manufacturing of ICs [10];
- combination of technologies that in a single production cycle make it possible to design, analyze, and redesign ICs, employing both digital processing methods and NNs;
- development of parallel algorithms and means for mapping the corresponding algorithms onto multiprocessor computing structures, including supercomputers.

The equipment of ICs layout inspection is characterized by a large diversity and essentially differs in degree of complexity: from simple visual inspection tools for mass production to the most complex automatic inspection and measurement systems which are used both in R&D of new technologies and devices, as well as in the large-scale production. Automatic inspection and measurement system for defects in semiconductor wafers is already used in microelectronics [11]. Special systems for realization of critical technologies in microelectronics and precise engineering are foundation for development and further growth of modern technology.

Modern means of developing ICs are aimed at reducing time for mastering and launching new products into production, as well as reducing cost of digital equipment during its mass production. Such an opportunity is provided by technological base, including machine (computer) vision systems, which are an integral part of modern technology for the design and production of ICs.

In connection with updating new submicron design standards and increasing complexity of the ICs themselves, it becomes necessary to solve problems of processing, storing, receiving and transmitting large amounts of data obtained during lithographic process of ICs design. Original approaches for image processing allow to fully complying with conditions of submicron manufacture of Very large-scale integration (VLSI) and to reduce cost of production. The object of the study is process of critical dimensions inspection on the photomasks and VLSI layouts. The processing consists of image analysis, generating reports based on the previous analysis results, controlling the focusing system, coordinate table and other external devices, as well as synthesis of routines for the automatic operation of control equipment for monitoring of layout critical sizes.

The developed Software Control System (SCS) for equipment of ICs layout inspection is based on machine vision and provides the following functions: image preprocessing taking into account design and technological constraints; image processing and analysis with support for third-party video camera equipment; image analysis to control design and technological constraints; storage and access to data with the ability to import and export data in various formats; program synthesis for automatic operation; management of third-party mechanisms; data visualization. The main analogs of the SCS are the Olympus MicroSuite FIVE software systems from Olympus Corporation (Japan) and NIS-Elements Microscope Imaging Software from Nikon Instruments Inc. (USA). Analogs of installations for monitoring critical dimensions and their approximate cost: LEICA LWM 250 UV – 4.8 million dollars, KLA Tencor IPRO4 – 5 million dollars.

II. SCS Architecture

The developed architecture of the SCS is shown on Fig. 1. The work of the SCS is carried out as follows:

1) Initialization of user work by issuing control actions to the control system (by the user interface or loading the configuration for automatic operation).
2) Transformation of control commands, if necessary.
3) Receiving data from video camera using the appropriate SDK, convert data for processing.
4) Processing and analysis of data by appropriate subsystems.
5) Transmission of video stream and analysis results in the control system, if necessary, format conversion.
6) Development of control actions by the decision-making subsystem (decision based on the received commands, video stream and analysis results).
7) Transferring of data at the control process to the virtual data model.
8) Signal transmission when changes the state of the model from the virtual data model to the graphical user interface.
9) Request the required data graphical user interface from the virtual data model and retrieving them.
10) Transmission commands of control to equipment by the decision-making subsystem through the appropriate next path (interface, the mechanism control system and the corresponding Software Development Kit (SDK)).
11) Save the results of control in the DB.

III. Image processing

The SCS includes implementation of main function by special systems and subsystems: an image processing and analysis system, including a video camera subsystem for preparing data for use; a control system for functional linking of other systems and subsystems; the mechanism control system for generating unit control commands; graphic user interface for the user to control the functioning; subsystem of interaction with the DB for storing the results of inspection; a subsystem for control program generation (description preparation of the configurations of operation used in the automatic mode of operation). In addition, each of the systems must be implemented with a sufficiently high level of abstraction to ensure uniform operation when using different video equipment and control equipment. So when choosing an design pattern for software package, the following criteria were used: modularity; openness; configurability; separation of graphical user interface and functionality. The most convenient design pattern based on the listed before criteria is MVC (Model-View-Controller). MVC pattern with some modifications allows to take into account mentioned above criteria and requirements to the architecture of the SCS: the control system must be able to receive commands from several sources – an interface for receiving control commands is added for converting general view commands (meta-commands) into specific commands for such equipment; the control system must be able to receive a video stream from several sources – an interface for receiving video data is added, which converts various data formats to a single format an interface for receiving control commands is added for converting general view commands (meta-commands) into specific commands for such equipment, the virtual data model that stores a description of the state of the parameter control process is implemented. The main part of SCS is the image processing subsystem that implements both basic and special image processing algorithms The basic ones are : Contrast Correction; Gamma Correction; Invertion; Laplace Filtration; Mean Filtration; Median Filtration; Morphological Closing; Morphological Dilatation; Morphological Erosion; Morphological Opening; Threshold Binarization [2] – [4], [12] – [14]. The special ones include preprocessing, autofocus and analysis algorithms. The preprocessing algorithms are:

- algorithms of preliminary processing of images of layers of VLSI taking into account DTLs;
- algorithms for increasing the informativeness of images of layers of semiconductor chips, taking

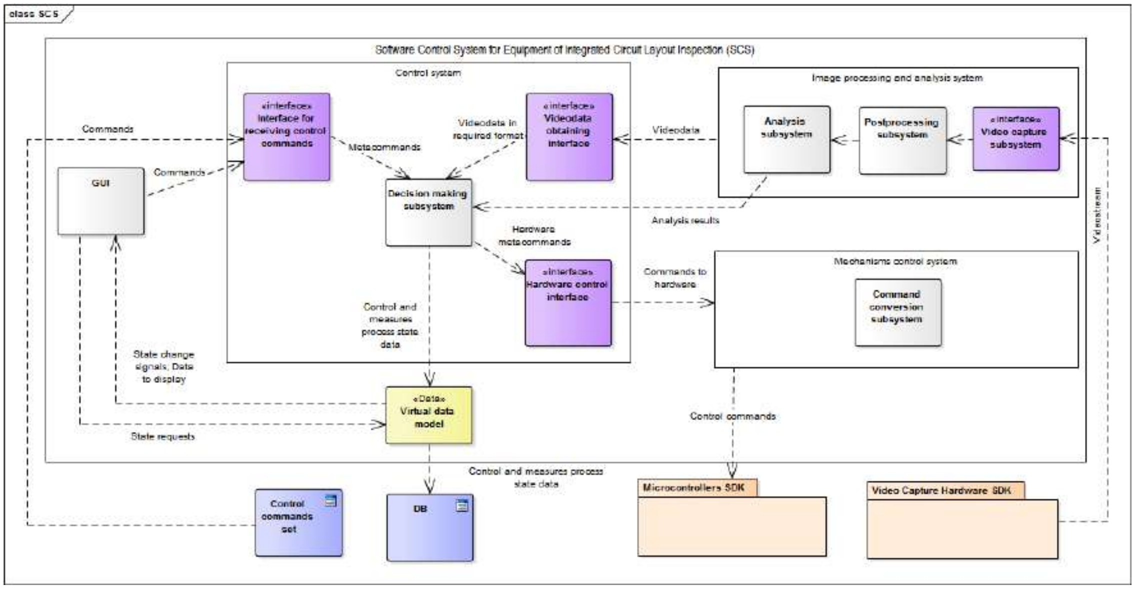


Figure 1. Architecture of the SCS

into account DTLs, based on combining objects of interest from several copies of the chip;
- algorithms of semantic filtering of images taking into account DTLs, based on the use of descriptors of the shape of the image segments: the total area of the segment, the geometric center of the segment, the length of the main diagonal of the segment, the dimensions of the approximant rectangle for the segment;
- an algorithm for preprocessing images taking into account DTLs based on soft morphology, conditional erosion and dilatation;
- algorithms of stitching images taking into account DTLs;
- algorithm for improving images taking into account DTLs, which allows you to align the heterogeneity of lighting of individual frames and brightness differences at their border.

The autofocus algorithms are:

- the algorithm for estimating the sharpness function in which several different criteria are quantied on specific series of images;
- the algorithm of continuous automatic determination of focal length for continuous determination of the correct focal length for the survey system from images of the VLSI topology;
- the lgorithm for determining the unevenness of illumination based on the analysis of the edge of the topology. According to this algorithm, it is supposed to set the direction of the scanning line on the image, along which the edge analysis is performed.

The analysis algorithms do:

- alignment and orientation – binding of the reference system and the coordinate system of the object to the coordinate system of the installation;
- control and measurement of dimensions – launch of algorithms for control and measurement of dimensions;
- automatic measurement – launch of algorithms for automatic measurement of dimensions;
- determination of the size of elements - the launch of algorithms for determining the size of the image;
- creation of a control and measurement program for automatic mode - the formation of a list of control effects with the appropriate parameters and their saving as a file or record in the database.

The functioning of the system is carried out according to the generalized algorithm, the steps of which are given below [15] - [19].

Step 1. Loading operation parameters from the command source ICommandSource.

Step 2. Loading dynamic libraries.

Step 3. Preparation of the algorithms' implementations required for processing.

Step 4 If the images in the IFrameSource are still available, then go to step 5, otherwise go to step 10.

Step 5. Loading image from source.

Step 6. Image processing with the selected algorithm / algorithms.

Step 7. Analysis of the processed image.

Step 8. Saving the analysis result to the IProcessingResult Storage result storage.

Step 9. Go to step 4.

Step 10. Completion of work. As noted earlier, the software module for image processing and analysis, almost all elements of the SCS should be a set of dynamically loaded libraries containing supported functions.

CONCLUSION

The general architecture of program complex for control of equipment for monitoring of critical size based on machine vision systems has been developed, which allows working with big input data and easily adapted to specific equipment. In this paper, the requirements and the structure of the SCS are described. technology allows identifying effectively defects that is especially important for software engineering for equipment of critical sizes inspection of VLSI manufacturing based on submicron technology. The developed software provides the following functions:

- image preprocessing taking into account design and technological restrictions;
- image processing and analysis with support of different video equipment;
- image analysis for inspections of manufacturing operations;
- storage and access to data with the ability to import and export data in various formats;
- synthesis of the program for automatic operation mode;
- control of different mechanisms;
- data visualization.

The architecture of the software system was developed, providing possibility of flexible adjustment of general algorithm of image processing and analysis. The user can independently compose chains of simple algorithms to obtain more complex ones. It is also possible to connect external routines. The operation parameters and automatic processing programs are stored in the database. The software is used in production of competitive precision equipment for VLSI manufacturing what determines its practical importance: for automatic photometry with precision laser focusing system; for automated microsize inspection system; for mask pattern coordinates measurement system, equipment for mask pattern generation and inspection. A significant advantage of equipment of the controlled of SCS and developed by JSC Planar for the production of VLSI over foreign counterparts is that it is designed on a single design and technological base, realizing full hardware, software and metrological compatibility of the entire set of installations operating in a single technological cycle for Embodiments in silicon of critical technologies of the microelectronic industry.

REFERENCES

[1] S. V. Ablameiko and D. M. Lagunovskii, Image Processing: Technology, Methods, and Applications (Inst. Tekh. Kibern. NAN Belarusi, Minsk, 2000) [in Russian].

[2] Methods of Computer Optics: Handbook, 2nd ed., Ed. By V. A. Soifer (Fizmatlit, Moscow, 2003) [in Russian].

[3] R. C. Gonzalez and R. E. Woods, Digital Image Processing (Prentice Hall, 2002; Tekhnosfera, Moscow, 2005).

[4] Ya. A. Furman et al., Introduction to Contour Analysis; Applications to Image and Signal Processing (Fizmatlit, Moscow, 2003) [in Russian].

[5] D. Forsyth and J. Ponce, Computer Vision: A Modern Approach (Prentice Hall, 2003; Izd. Dom Vil'yams, Moscow, 2004).

[6] V. V. Krasnoproshin and V. A. Obraztsov, "Problems of solvability and choice of algorithms for decision making by precedence," Pattern Recognit. Image Anal. 16 (2), 155–169 (2006).

[7] R. Kh. Sadykhov, P. M. Chegolin, and V. P. Shmerko, Signal Processing Methods in Discrete Bases (Nauka Tekh., Minsk, 1987) (in Russian).

[8] S. M. Avakov. Automatic Control of the Topology of Planar Structures (FUAinform, Minsk, 2007) [in Russian].

[9] A. A. Doudkin and R. Kh. Sadykhov, Image Processing in the Design and Production of Integrated Circuits (OIPI NAN Belarusi, Minsk, 2008) [in Russian].

[10] A. I. Galushkin, in Theory of Neural Networks, Ed. By A. I. Galushkin (IPRZhR, Moscow, 2000), Book 1 [in Russian].

[11] A. P. Dostanko, eds. Technological complexes of integrated processes for the production of electronic products. Minsk, Belorusskaya Navuka [Belarusian Navuka], 2016, p. 251.

[12] V. V. Starovoitov. Local geometric methods of digital image processing and analysis. Minsk, In-t techn. Cybernetics [In-t techn. Cybernetics], 1997, p. 282.(In Russian)

[13] V. A. Soifer. Theoretical foundations of digital image processing : Textbook. Samara: SSAU, 2000, p. 255

[14] A. A Dudkin, A. A. Voronov, E. E. Marushko. Algorithm for stitching frames of the VLSI topology layer by key points. Bulletin of the Brest State Technical University, Physics, mathematics, computer science, No. 5 (94), 2015, pp. 11-14. (In Russian)

[15] M. Fowler. Architecture of corporate software applications. Moscow: Publishing house "Williams", 2006.

[16] L. Craig. Application of UML 2.0 and design patterns. Ed. 3, Moscow, Izd. house "Williams", 2013.

[17] G. Erich. Techniques of object-oriented design. SPb. Ed. house "Peter", 2015, p. 368.

[18] E. Evans Domain-oriented design (DDD): structuring of complex software systems / Eric Evans. – Moscow: "Williams", 2011.

[19] D. Spinellis, G. Gusios. Ideal architecture. Leading experts on the beauty of software architectures. St. Petersburg, Symbol-Plus, 2010.

Программная платформа для проектирования систем оптического контроля изготовления прецизионного оборудования микроэлектроники

А.А. Дудкин, С.М. Аваков,
А.А. Воронов, В.В. Ганченко

При реализации программного комплекса была разработана архитектура, обеспечивающая возможность гибкой настройки общего алгоритма обработки и анализа изображений.

Отличительной особенностью архитектуры программной платформы и используемых алгоритмов является обеспечение измерения с автофокусировкой для повышения повторяемости и точности контроля и возможности гибкой настройки общего алгоритма обработки и анализа изображений, включая распараллеливание.

Программная платформа применяется при производстве конкурентоспособного прецизионного оборудования для изготовления высокоточных оригиналов топологий изделий электронной техники и обеспечивает определение неровности края на всем участке измерений и контроль критических размеров полупроводниковых пластин СБИС с поддержкой минимальных элементов размером 350 нм и повторяемостью не хуже 2 нм.

Received 16.07.2021

Scientific principles of coordinating the functioning of elements in a multi-level Intelligent Transportation Control System

Erofeev Aleksandr
Belarusian State University of Transport
Gomel, Belarus
Email: alerof@tut.by

***Abstract*—It has been established that the description of the coordination of functioning in the Intelligent Transportation Control System (ITCS) involves the use of three levels of hierarchy. The features of the stratified system are determined. It is proposed to use the principle of multilayer when forming control decisions, when each layer is responsible for solving coherent operational problems. The functions of the three main layers of the system are determined. Describes the multilevel organizational hierarchy of the ITCS. A formalized description of the coordination of the functioning of the elements of a two-level fragment of the ITCS**

***Keywords*—Intelligent Transportation Control System, coordination, stratified system, organizational level, abstraction description level, multi-level system, Multilayer hierarchy**

One of the most important intellectual property of systems should be considered the ability to work in coordination. Such system property provides:

- solving complex problems by collective efforts;
- formation of hybrid solvers for new problems that were not considered at the stage of creating systems;
- improving the quality of control solutions (CS) due to the subsequent integration of modules with new more accurate algorithms;
- detection of errors in the CS by comparing their incoming information from different sources (systems).

Coordination of functioning can be ensured through the use of harmonized standards (for example, specifications based on OSTIS technologies) and unified principles of building intelligent systems [1–4].

In this article the key principles of the coordinated functioning of elements in a multi-level system, based on the Intelligent Transportation Control System (ITCS) example, are proposed [5, 6].

The description of the rules for coordinating functioning in the ITCS involves the using of three hierarchy levels [7]:

a) the object environment or abstraction description level (stratum);
b) complexity of generated CS level (layer);
c) organizational level (echelon).

The ITCS as a stratified system (Figure 1) is described by a family of models, each of which reflects the behavior of the system at different levels of abstraction. Each level has characteristic features, parameters, rules of behavior, patterns.

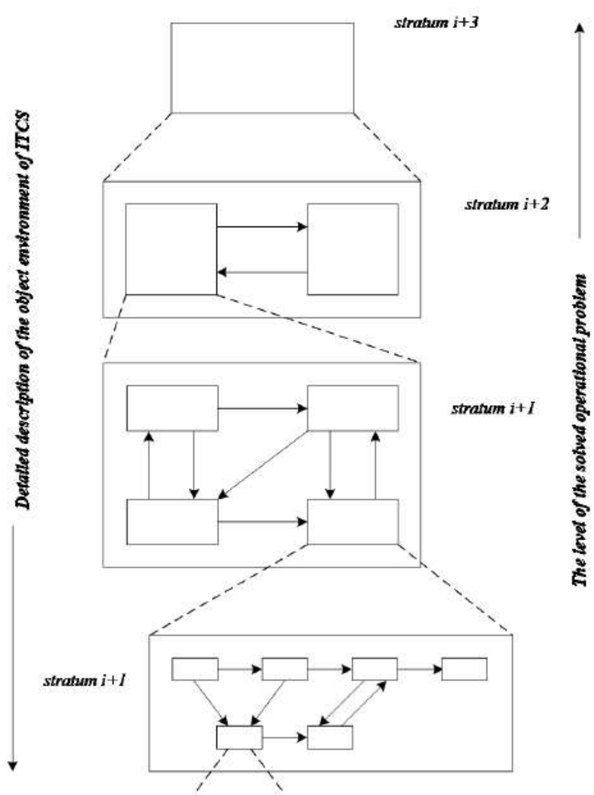


Figure 1. The relationship between the strata of the ITCS

The following features are inherent in a stratified system.

1. The choice of strata used in the ontological description of a subsystem depends not so much on the goals of the functioning of the ITCS, as on the goals of the functioning of a particular subsystem. For example, an object "marshalling yard"

in solving various operational problems will require descriptions from different points of view. For the development of a train formation plan (TFP), a station can be described only by enlarged quantitative parameters: the accumulation parameter C, the savings from running without processing t_{ec}, the limiting processing capacity N_{trans}.
When developing a train schedule (GDS), it is necessary to know the characteristics of the station's receiving and departure tracks and their layout. When managing plant operations, detailed characteristics of plant objects are required. In the above examples, the control object is considered from similar positions, which requires the consistency of descriptions in different subsystems. However, when solving the problems of "personnel work", "accounting", "repair management" approaches to the description of the "marshalling yard" will be fundamentally different.

2. Ontologies of subsystems on different strata are generally not interconnected. Therefore, the laws, principles of functioning, decision rules, etc., which are used to describe a subsystem on any stratum, in the general case, cannot be deduced from the principles used in another stratum. For example, the decision rules of the subsystem "planning of shunting work" cannot be used in the subsystem "control of the work of a shunting locomotive". In this regard, in ITCS for each subsystem, its own ontology can be used. The set of ontologies of subsystems form a single ontology of the ITCS. The purpose of a unified ontology is to ensure the consistency of descriptions of private ontologies among themselves.
3. There is an asymmetric relationship between the conditions for the functioning of the ITCS on different strata. The requirements for the conditions of functioning and the decision rules of the higher subsystems act as conditions or restrictions for the lower ones. For example, the choice of the speed of shunting movement in the subsystem "planning of shunting work" is a limitation in the formation of the CS in the subsystem "control of the operation of the shunting locomotive". As a result, information (feedback) about the course of the real process and its deviations from the planned values should be transmitted to a higher level.
4. Each stratum has its own set of terms, definitions, decision rules and concepts. An elementary object of a higher stratum may be an independent subsystem on a lower one. Subsequent strata describe the internal mechanisms of the object's behavior and the principles of its functioning. The higher strata are the principles of interaction between objects. For example, a station on higher strata (for example, when developing a plan for the formation of trains) should be described in aggregate, and the emphasis is on the principles of interaction between stations. In the lower-level stratum, the station is an independent subsystem with its own principles and peculiarities of functioning. As a consequence, the study of an object in a lower stratum does not always make it possible to more effectively solve the problems of higher strata. Thus, the ITCS ontology should provide for a hierarchy of semantic relations between any two successive subsystems of the hierarchy: "subsystem behavior rules - rules of interaction between subsystems".
5. The detailing of the principles of ITCS functioning increases with the successive transition from strata of higher levels to strata of lower levels. A different detail level of control objects in digital models of subsystems belonging to different strata makes it possible to simplify the general description of ITCS, but ensure its necessary completeness.

The system of CS formation in ITCS is based on the principle of multilayer - when each layer is responsible for solving coherent operational tasks (OT), but taking into account different "powers" and detailing the final decisions (Figure 2).

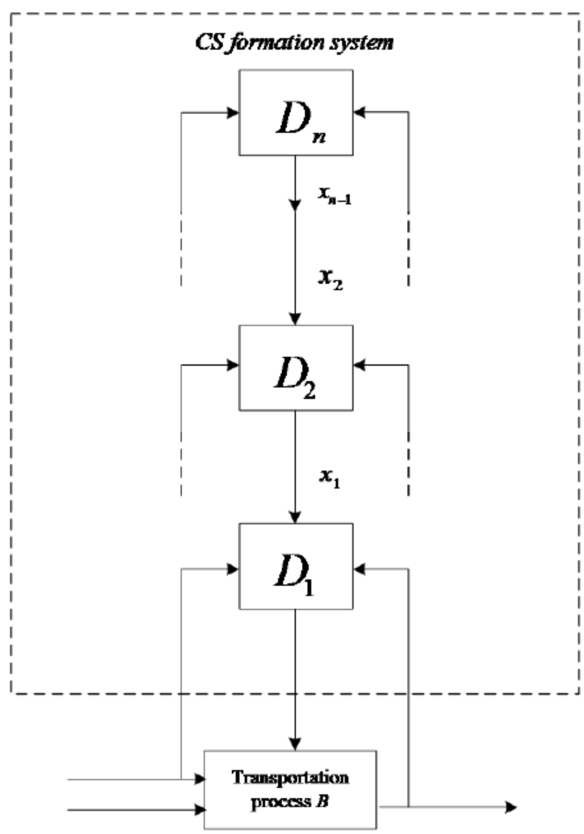


Figure 2. Multilayer hierarchy of the CS formation system in ITCS

For example, the upstream OT "TFP Development" defines a plan for the distribution of shunting work between the technical stations. OT "formation planning" at the station level, based on the established TFP, develops a shunting plan. The subordinate subsystem "control of shunting work" on the basis of a train formation plan forms shunting routes and transmits telecontrol commands to shunting locomotives.

CS in ITCS can be thought of as a family of consistently solved operational problems. Those. CS of any of the OT defines such control parameters that allow formulating the subsequent problem as fully defined and solvable. The CS of the original OT can be considered found if all associated OTs have been (or can be) resolved. For example, if within the framework of the decision OT "formation planning" it is established that the specified amount of shunting work cannot be performed (due to a shortage of track or shunting resources, etc.), the OT "development of TFP" is considered not solved and requires re-consideration.

Thus, each element in Figure 2 is a formative CS and can be assigned to a certain hierarchical layer. The output of an element (for example, D_2) x_1 is CS or a sequence of CS, depending on the input control x_2. At the same time, the input parameters x_2 are the CS of the parent element. In the future, in the ITCS, such a system of CS formation will be called the hierarchy of layers of CS formation, and the system itself will be called a multilayer decision-making system.

In ITCS, as a system providing CS formation in conditions of uncertainty, the hierarchy of CS formation layers is determined by the following stages:

1. Choice of the strategy that should be used in the process of CS adoption;
2. Elimination or minimization of uncertainty;
3. Search for the preferred or acceptable CS according to the given rules.

The ITCS functional hierarchy is shown in Figure 3.

1. Layer of choice: the purpose of the layer is the formation of CS m, which directly affects the subject of management (transportation process). The element forming the CS on this layer receives from the environment and higher subsystems the initial data for solving the OT and, using one or another algorithm, forms the CS. The algorithm can be defined directly as pre-formulated in the ontology of system T or indirectly using the search process.

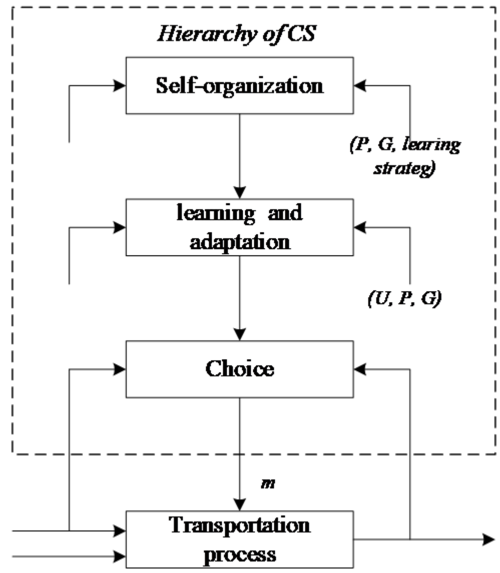


Figure 3. Functional multi-layer hierarchy CS

For this, the system specifies the output function P, the estimation function G, and the choice of the effective CS m is based on the application of the estimation function G to the function P. Using the set-theoretical approach of systems theory, the output function can be defined as a mapping P:

$$M \times U \rightarrow Y, \tag{1}$$

where M - is a set of alternative CS; Y - is the set of possible outputs; U - is a set of uncertainties that adequately reflect the lack of knowledge about the relationship between CS m and the output y.
Similarly, the estimate function G is a mapping G:

$$M \times Y \rightarrow V, \tag{2}$$

where V - is a set of parameters that can be related to the characteristics of the system's functioning quality.
If the set U consists of one element or is empty, i.e. there is no uncertainty at the output with respect to CS m, the choice can be based on optimization: find such m in M such that the value $\nu' = G(m', P(m'))$ is less than $\nu' = G(m, P(m))$ for any other CS $m \in M$.
If U is a richer set, the choice of an effective CS can be based on other principles, including those involving the introduction of additional, in addition to P and G, mappings.
Corollary 1. The search for an effective CS based on optimization approaches is a special case of search. When solving a significant number of OT, fundamentally different criteria for choosing effective CS can be used.

Consequence 2. The evaluation function G should be formed in the external, relative to the forming CS element, subsystem and be consistent with the functioning goals of higher subsystems. Otherwise, you can always find such a function G', which will allow you to define any CS as effective.

2. Layer of adaptation. The task of this layer is to concretize the set of uncertainties U, which are inherent in the layer of choice, and to narrow this set. If the system and environment are stationary, then the set U can be narrowed down to one element. However, U represents not actually existing, but assumed uncertainties. When solving new OT in ITCS, the U layer can be modified (extended), thereby assuming that certain underlying hypotheses are unfair and require adaptation to the predicted operational environment. For example, a station shunting plan is developed based on the number of locomotives available at the station (the number of locomotives is known, there is no uncertainty). However, in the process of performing the technical task, it became necessary to carry out unscheduled repairs with an indefinite time for their completion. Under such conditions, the number of shunting locomotives at the station is an undefined value.
3. The layer of self-organization. Designed to select the structure, functions and strategy for solving OT, used on the lower layers, taking into account the maximum approximation to the global goal of the system functioning. If the global goal is not achieved, the subsystem of this layer changes the functions P and G on the first layer or the adaptation (learning) strategy on the second layer.

In ITCS, as a multi-echelon system, the concept of hierarchy implies that:

1. The system consists of a family of clearly interacting subsystems;
2. Some of their subsystems are forming CS;
3. The forming CS elements are arranged hierarchically in the sense that some of them are controlled by other (superior) elements (Figure 4).

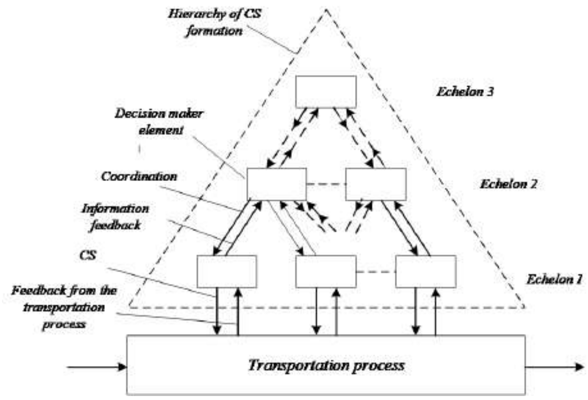


Figure 4. Multilevel organizational hierarchy of ITCS

The presence of many subsystems that have the right to form the CS implies the possibility of the functioning "conflicting" goals existence. In addition, the goals of functioning can be not only "conflicting", but also form "coalitions". Traditionally, such systems are considered as multi-agent, but this approach leads to an oversimplification of the system description (functional connections between elements of the same level, target parameters of functioning set by higher subsystems, etc. are neglected). In addition, conflicts between elements of the same echelon are not resolved within one hierarchical level, but through the intervention of an element of a higher level.

Considering ITCS as a multi-level multi-purpose system, it should be borne in mind that the upper level elements determine the purposeful activity of the lower level elements, but do not completely control it. Subordinate elements forming CS should have the freedom to form and choose CS. In some cases, the CS chosen by the lower level may differ from the choice of the higher level element. For example, when generating a predictive GDS, the route of receiving a train to a station is determined. However, in the subsystem "Managing the operation of a technical station" in view of the availability of more complete information (information awareness), a different reception path can be chosen (for example, due to the presence of a train on the initial track with which operations for non-decoupling car repair are performed).

At the same time, in the interest conflict event arising between elements of one or different echelons, the priority in the choice of SD remains with the element of a higher level. Let's consider the coordination of functioning on the example of a two-tier fragment of ITCS. The choice of such a fragment is explained by the following reasons:

1. This is the simplest type of system in which all significant properties of multilevel systems are manifested;

2. Any more complex multi-level system can be built from two-level by hierarchical unification.

Fragment ITCS contains upstream control elements C_0 and downstream C_1, C_2, ..., C_n of various control levels, the control object for which is the transportation process B_1, B_2, ..., B_n. (Figure 5).

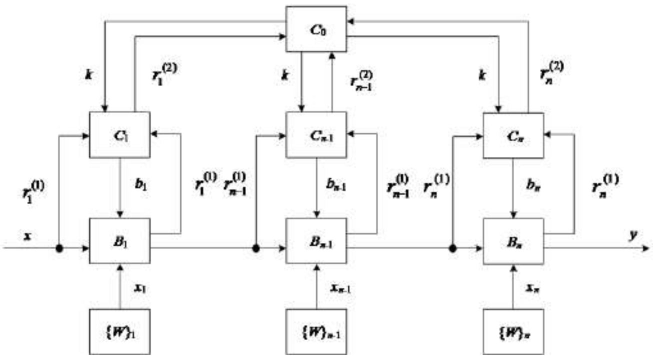


Figure 5. Diagram of management coordination in ITCS

Bodies of the control system C_1, C_2, ..., C_n act on the control objects B_n through the implementation of the SD b_k ($b \in B_k$ to the set of admissible SD) and receive information about the progress of the SD implementation on the control objects $r_n \in R_n^{(1)}$ (where $R_n^{(1)}$ – is the set of information messages of the 1st level).

In ITCS, the objective of good governance is formulated as follows:

– the subsystem of the higher level C_0 should form such CS $\{k_i\}$ so that at the lower level subsystems C_1, C_2, ..., C_n form operational tasks and optimal local SD $\{b_n\}$ in relation to the global quality function Y;
– subsystems C_1, C_2, ..., C_n solving problems of local optimization should provide an extremum of the quality function Y.

The model of ITCS functioning from the point of view of set theory describes the SP as

$$B_n\colon L_k \times X \to Y, \tag{3}$$

where X – is the set of perturbations of the external object environment affecting the controlled process.

The model of functioning of the control element C_i can be described by a functional of the form

$$C_i\colon K \times B_n \to L_k, \tag{4}$$

and the model of the coordinator K is implemented in the form

$$C_0\colon R_n^{(2)} \to K, \tag{5}$$

where $R_n^{(2)}$ – set of second level information signals.

The implementation of the control loop in ITCS includes information coming through feedback channels ($R_n^{(2)}$). Feedback information enters the control bodies and provides the formation of an array of data on the state of control objects. Feedback allows you to form dependencies between a set of control decisions L_k and changes in the parameters of control objects B_n, i.e.

$$x_i\colon L \times W \times Y_n \to R_n^{(2)}, \tag{6}$$

where W - is the set of all possible factors influencing the result of solving the control problem; Y_n - set of effective outputs of transportation processobjects in the considered subsystem.

The function coordinating control actions coming from the upper control level is defined as

$$F(L) \to x_0\colon K \times R_n^{(2)} \times L, \tag{7}$$

where K - is a set of coordinated actions.

At the lower control level, the tasks of coordinating subprocesses to be solved can be optimization. In this regard, it is expedient to consider any local problem as a pair (g_i;x_i), where g_i - is a given local objective function; x_i - a predetermined set of input actions X. In this case, we can assume that g_i it is determined by the output function of the process B_n and the local quality function $\mathbf{G}_i$:

$$g_i(x_i) = G_i(x_i, B_n(x_i)). \tag{8}$$

The solution to the local optimization control problem in the ITCS subsystem is the element $x_i \in X$, which determines the minimization condition

$$g_i(x_i) = \min g_i(x_i). \tag{9}$$

In ITCS, the implementation of the principle of coordinating the activities of subsystems is based on:

– for top-level subsystems - on "coordination of interaction";
– for subsystems of a lower level - on "unleashing interaction".

To implement coordination in ITCS, it is proposed to use a modified method for obtaining the quality function for lower-level elements by means of operators for evaluating the indirect effect of the implementation of CS.

In order to obtain the global optimum of the function Y in accordance with the method of coordinating "untied" interactions, the elements of the lower level must maximize their quality functions both in terms of local controls and in interactions. Subsystem C_0 should form coordinating CS k_w^j so as to balance interactions

$$\sum_{j=1}^{n} y_{jf}^i = u_{if}; \sum_{j=1}^{n} y_{jw}^i = u_{iw}, \tag{10}$$

where u_{if}, u_{iw} – optimal parameters of subsystem functioning adopted at the upper control level; y_{jf}^i, y_{jw}^i – optimal performance parameters selected by the corresponding subsystems at the lower control levels.

The use of the above principles made it possible to create and introduce into industrial operation a number of intelligent systems on the Belarusian Railway: “Shift-daily planning of cargo work” - the system allows you to develop detailed agreed work plans for three levels of management and more than 100 objects of management;

“Linking the formation with the train schedule” - the system allows to solve a new operational problem due to the coordinated interaction of the systems “train planning”, “station work planning” and “development of the train schedule”. Detailed coordinated plans are being developed for two management levels and about 20 objects of management.

Harmonized OSTIS standards and uniform principles for coordinating the functioning of intelligent systems can be used to build intelligent systems in other areas of knowledge. In the future, this will allow the creation of intelligent systems that have interdisciplinary knowledge and are able to form CS for various fields of activity.

References

[1] V. Golenkov and N. Guliakina, “Principles of building mass semantic technology component design of intelligent systems,” in *Open semantic technologies for intelligent systems*. Minsk : BSUIR, 2011, pp. 21–58.

[2] V. Golenkov and N. Gulyakina, “Semanticheskaya tekhnologiya komponentnogo proektirovaniya sistem, upravlyaemyh znaniyami [Semantic technology of component design of knowledge-driven systems],” in *Materialy V mezhdunarodnoj nauchno-tekhnicheskoj konferencii “Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nyh system”, 19-21 fevravya, Minsk [Proceedings of the V international scientific and technical conference of “Open Semantic Technologies for Intelligent Systems”, 19-21 February, Minsk*. Minsk : BSUIR, 2015, pp. 57–78, (in Russian).

[3] V. V. Golenkov, “Ontology-based design of intelligent systems,” in *Open semantic technologies for intelligent systems*, ser. Iss. 1. Minsk : BSUIR, 2017, pp. 37–56.

[4] D. Shunkevich, “Ontology-based design of knowledge processing machines,” in *Open semantic technologies for intelligent systems*, ser. Iss. 1. Minsk : BSUIR, 2017, pp. 73–94.

[5] A. Erofeev, “The intelligent control concept of the transportation process on the belarusian railway,” in *Open semantic technologies for intelligent systems*, ser. Iss. 4. Minsk : BSUIR, 2020, pp. 157–160.

[6] ——, “Semioticheskaya model’ perevozochnogo protsessa i ee ispol’zovanie pri proektirovanii intellektual’nykh system [Semiotic model of the transportation process and its use in the design of intelligent systems],” in *Intellektual’nye sistemy upravleniya na zheleznodorozhnom transporte. Komp’yuternoe i matematicheskoe modelirovanie [Intelligent railway managementsystems. Computer and mathematical modeling]*. Moscow, 2017, pp. 24–26.

[7] M. Mesarovich, D. Mako, and I. Takahara, *Teoriya ierarkhicheskikh mnogourovnevykh struktur [Theory of hierarchical multilevel structures]*. Moskow, Mir, 1973.

Научные принципы координации функционирования элементов в многоуровневой интеллектуальной системе управления перевозочным процессом

Ерофеев А.А.

Установлено, что описание координации функционирования в интеллектуальной системе управления перевозочным процессом предполагает использование трех уровней иерархии. Определены особенности стратифицированной системы. Предложено при формировании управляющих решений использовать принцип многослойности, когда каждый слой отвечает за решение связных эксплуатационных задач. Определены функции трех основных слоев системы. Описана многоуровневая организационная иерархия ИСУПП. Приведено формализованное описание координации функционирования элементов двухуровневого фрагмента ИСУПП.

Received 04.05.2021

Situation-semantic Modeling a Risk Monitoring System in Urban Transport

Kostiantyn Tkachenko, Olha Tkachenko
Department of Information technologies and Design
State University of Infrastructure and Technology
Kyiv, Ukraine
tkachenko.kostyantyn@gmail.com, oitkachen@gmail.com

Oleksandr Tkachenko
Department of Software Engineering
National Aviation University
Kyiv, Ukraine
aatokg@gmail.com

***Abstract*—The article deals with the actual problems of modeling of monitoring risks in urban transport. This monitoring involves determining the state of transport objects, as well as procedures for minimizing the consequences caused by the onset of risks.**

The article discusses the problems of effective creation of system for monitoring of risks in urban transport based on the use of appropriate models.

The article proposes modeling this system based on situational-semantic model. The approach proposed would contribute to recognition of risks and generation of management decisions to eliminate their consequences.

The proposed situational-semantic model allows: to predict the behavior of complex transport objects and transport infrastructure objects; take into account the possibility of emergence of new transport objects and processes for ensuring their functioning in conditions of minimizing possible risks; respond adequately to local and global factors of influence on transport objects; dynamically change the structure of the system; take into account new data to predict the development and improvement of relevant transport objects; predict the development of processes to ensure minimization of risks and the consequences of them.

***Keywords*—model, situational-semantic model, modeling, information system, risks in urban transport, monitoring of risks in urban transport.**

I. Introduction

Urban transport is always associated with certain risks. Studying these risks, determining the degree of their impact on the functioning of transport objects and transport infrastructure objects is a complex problem.

This is due to the fact that, firstly, there are several classes of risks in urban transport, in particular: technical (production, technological, innovative), economical (property, commercial, financial, credit and interest rate); social; environmental.

Each of the risks is associated with a large number of dynamically changing factors of influence. Therefore, the monitoring of risks, the development of appropriate information systems is an important and urgent problem.

The creation of such complex systems will be optimal if it is based on the use of models of objects and processes in urban transport.

Modeling of risk monitoring systems in urban transport would facilitate the timely recognition of risks and the generation of solutions to eliminate their consequences.

When building complex technical system, which is a system for monitoring of risks in the transport [1, 2], it is important to choose the method of presenting knowledge about the domain (urban transport).

Among the main types of models of knowledge representation should be noted [3 - 6]: production model; semantic network; frame model; ontological model [7, 8].

II. Modeling of Monitoring of Risks

The main goal of this work is to develop a model of the system for monitoring of risks in urban transport. As such a model, a situational-semantic model was proposed [9, 10].

The authors propose to use modeling based on a multilevel model of a special class of semantic networks (SM) - situational-semantic networks (SSM), in which situations determine not only the description of the model at any of its levels, but also the transition from one level to another.

The SM that is used is a system of knowledge of domain. This has certain content in the form of a coherent image of a specific network model of the system for monitoring of risks in the urban transport.

The nodes of this model correspond to transport objects or transport infrastructure objects (concepts of the subject area, system components, constituent elements of the process of ensuring the functioning of transport objects).

Arcs reflect the relationship between all objects of the considered domain.

When constructing SM, the number of elements and their connections is not limited, and the systematization of relations between transport objects and transport infrastructure objects of the network is necessary for the subsequent formalization of the processes of monitoring risks in urban transport.

Systematization of SM relations is a complex problem and depends not only on specific objects in urban transport, but also on the processes occurring in them and possible risks both as a result of their functioning and for their functioning.

When systematizing SM relations, an important role is played by the hierarchy of relations between objects, which can be divided into:

- generally valid (characteristic of almost all objects);
- significant (characteristic of many objects);
- specific (characteristic of individual objects).

Objects are understood as transport objects and/or transport infrastructure objects.

The authors propose SSM, which also takes into account the situational monitoring of the risks of objects in urban transport.

All situations, according to which modeling of the system for monitoring of risks in urban transport is carried out, can be divided into: regular, non-standard.

Standard situations can be divided into:

- generally significant;
- significant;
- specific.

The use of situations and their typification contributes to the multilevel structure of SSM of system for monitoring of risks in urban transport.

Formally, the SSM can be set as follows:

$$M_{ssm} = (G_{ssm}, H_{ssm}, U_{ssm}, S_{ssm}),$$

where

G_{ssm} – set of transport objects and transport infrastructure objects (nodes), $G_{ssm} \neq 0$;

H_{ssm}– set of connections between nodes (arcs),

$$H_{ssm} \subseteq (G_{ssm} \cup G_{ssm});$$

$$dom(H_{ssm}) \subseteq ran(H_{ssm}) = G_{ssm},$$

where

$dom(H_{ssm}) = (y \in G_{ssm} | \exists x \in G_{ssm}, (x, y) \in H_{ssm}),$

$$ran(H_{ssm}) = (y \in G_{ssm} | \exists x \in G_{ssm}, (x, y) \in H_{ssm}),$$

that is, any SSM node is incident to at least one SSM node;

U_{ssm} – set of loads on elements H_{ssm};

S_{ssm} – set of situations in which functioning occurs SSM.

$$G_{ssm} = (G^s_{ssmi}),$$

where G^s_{ssmi} - i-th node SSM;

$$H_{ssm} = (H^s_{ssmj}),$$

where H^s_{ssmj} - j-th arc SSM;

$$Ussm = U^s_{ssmj},$$

where U^s_{ssmj}– j-th load on the j-th arc SSM;

$$S_{ssm}. = (S^s_{ssmi}),$$

where

S^s_{ssmi}– situation that determines the semantics of the i-th node SSM.

Analysis of the functioning of a transport object based on the SSM provides quantitative and qualitative characteristics of its states.

If deficiencies are found in the SSM, then the model is modified several times until a model is obtained that is adequate to the transport object.

SSM components and their actions act as events. Examples of events can be, in particular: situational determination of the route on the SSM, according to which the values of the criteria that determine the situation are calculated.

The SSM of the system for monitoring of risks in urban transport should be:

- reliable;
- adequate;
- purposeful;
- simple and understandable for the user;
- complete;
- such that it assumes the possibility of modification.

To adequately reflect the connection between input and output in the SSM, the concepts "state" and "situation" are used.

The state $z(t_i)$ is a set of properties (states, situations) of the SSM, the knowledge of which at the moment of time $t > t_i$ allows us to determine its behavior at the moments of time $t > t_i$.

Modeling of system for monitoring of risks in urban transport, processes for ensuring the functioning of objects in urban transport in conditions of minimizing risks and/or eliminating their consequences should begin with:

- descriptions of all elements of the model of;
- determining the content of these components and areas of change.

For the full functioning of the system for monitoring of risks in urban transport, it is necessary to determine:

- the time interval in which the SSM;
- input and output impacts on possible risks and areas of their possible changes;
- the set of characteristics of the state of domain and the area of their possible changes.

Note that within the framework of one SSM, several variants of its submodels can be built (depending on situations, factors of influence, criteria for assessing risks, etc.)

The constructed model reflects expert knowledge about possible risks in transport, the reasons for their occurrence and and ways of their elimination or minimization.

There are two approaches to constructing an SSM.

1. "From above".

At the initial stage, the core of the SSM is built, which is further completed with the help of separate blocks of the model.

Individual components of the model can be dynamically completed to the core of the SSM.

2. "From below".

At the initial stage, the core of the SSM is built, which is further completed with the help of separate blocks of the model.

The model allows you to:

- clear cognitive analysis;
- dynamic modeling;
- forecasting trends in the development of the transport system and its individual subsystems (individual transport objects and transport infrastructure objects);
- forecasting the quantitative values of the criteria characterizing the risks in transport;
- analysis of model cycles, including life cycles;
- analysis of the stability of the process;
- analysis of structural resistance to disturbing and control influences;
- topological analysis of the structure of the model.

Topological analysis, calculation of system indicators of transport objects and transport infrastructure objects can also be carried out for the SSM.

The formation of control actions involves the implementation of the adjustment of the SSM.

A model correction is understood as:

- changing the structure of the model (adding or removing any objects, factors and relationships (connections) between them);
- change in values characterizing objects, factors and connections.

Users are prompted to make one of the following decisions:

- make adjustments to the initial SSM;
- to develop a new SSM.

Let's consider an example of risk assessment.

To build the SSM of the system for monitoring of risks in the transport, the factors necessary for assessing risks, the relationship between them, and their significance were identified.

At subsequent stages, when constructing an SSM, factors that characterize the individuality of transport objects and transport infrastructure objects can be used.

The values of the connections between the vertices are assigned based on expert knowledge (their opinions, judgments, forecasts, etc.).

The judgments were obtained by interviewing experts in the subject area under consideration - the urban transport.

Fig. 1 shows the risk assessment model using the principle "from below". Here:

v1 – the number of tasks;

v2 – the speed of execution of work to eliminate the consequences caused by the risks;

v3 – the number of expert assessments of risks and their consequences;

v4 – the onset of a situation caused by the emergence of a risk;

v5 – minimization of economic risks;

v6 – system reliability;

v7 – the ability of transport objects and objects of the transport system to function without the manifestation of negative environmental consequences;

v8 – external factors affecting transport objects and transport infrastructure objects;

v9 – the number of errors made by users of the system;

v10 – the time taken to create the system;

v11 - financial costs for creating the system;

v12 – the number of system users;

v13 – qualification of system users;

v14 – violation of normal operation;

v15 – emergency.

The relationships shown in Fig. 1 can be interpreted as follows:

- – for example, the relationship v3 $\rightarrow$ v6 with a weight of 0.9 means that if the value of the parameter of the vertex v3 increases (decreases) by 10 percent, then the value of the parameter of the vertex v6 increases (decreases) (sign "+") by 9 percent;
- the relationship v10 $\rightarrow$ v9 with a weight of – 0.7 means that if the value of the vertex parameter v10 decreases by 10 percent, then the value of the vertex parameter v9 will increase (sign "–") by 7 percent.

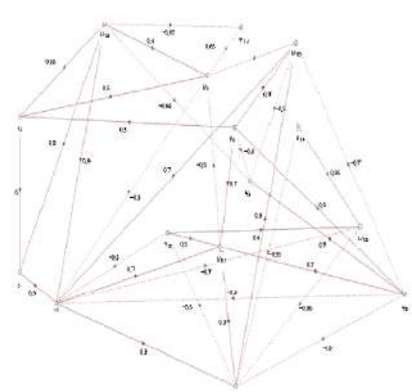


Figure 1. Risk Assessment Model.

III. Conclusions

The proposed situational-semantic model (SSM) of system for monitoring of risks in urban transport:

- allows predicting of the behavior of complex transport objects and transport infrastructure objects in the context of monitoring possible risks;
- takes into account (due to its dynamism) the possibility of the emergence of a new type of transport objects (and transport infrastructure objects) and processes for ensuring their functioning in conditions of minimizing possible risks;
- responds adequately to local and global factors of influence on transport objects and transport infrastructure objects;
- dynamically changes its structure;
- allows you to take into account new data for more accurate forecasting of the development and improvement of the relevant transport objects and transport infrastructure objects;
- allows predicting of the development and improvement of processes to ensure minimization of risks, their complete absence or elimination and minimization of consequences due to the onset of situations caused by the emergence of risks.

References

[1] Lipaev V.V. Problemy programmnoy inzhenerii: kachestvo, bezopasnost', risk, ekonomichnost' [The problems of software engineering: quality, safety, risk, economy]. Programmnaya inzheneriya [Software engineering], 2010, no 1, pp. 7-20.

[2] Info TS 08/17Compliance and Risk Monitoring of Transport Operations through Auditing. Available at: https://www.eiga.eu/publications/safety-informations-transport/info-ts-0817-compliance-and-risk-monitoring-of-transport-operations-through-auditing/ (accessed 2021, May)

[3] Nechaev V.V, Koshkarev MI. Arkhitektura intellektual'nykh sistem ob"yektno-oriyentirovannoy bazy znaniy [Architecture of object-oriented knowledge base intelligent systems]. Obrazovatel'nyye resursy i tekhnologii [Educational resources and technology], 2015, no 1, pp. 156-163.

[4] Adema J. New Models of Knowledge Production. Open Access Publishing and Experimental Research Practices (Part III). Available at: https:// openreflections.wordpress.com/2015/02/27/new-models-of-knowledge-production-open-access-publishing-and-experimental-research-practices-part-iii/ (accessed 2021, May)

[5] Types of Knowledge Representation. Available at: https://www.brainkart.com/article/Types-of-Knowledge-Representation-8892/ (accessed 2021, May)

[6] How Important Are Semantic Networks In Artificial Intelligence. Available at: https://analyticsindiamag.com/semantic-networks-ai/ (accessed 2021, May)

[7] Gorelova G.V. Kognitivnyy podkhod k modelirovaniyu slozhnykh system [A cognitive approach for modeling of complex systems]. Novosti YUFU. Tekhnicheskiye nauki [News of SFedU. Technical science], 2018, no 3, pp. 239-250.

[8] Chebanyuk O.V., Palahin O.V., Markov K.K. Domain engineering approach of software requirement analysis. CEUR Workshop Proceedings, 2020, no 2866, pp. 164-173.

[9] Tkachenko O.I., Tkachenko O.A. Nekotoryye aspekty situatsionno-semanticheskogo modelirovaniya slozhnykh ob"yektov, protsessov i sistem [Some aspects of situation-semantic modeling of complex objects, processes and systems]. Vodnyy transport [Water transport], 2017, no 1, pp. 129-133.

[10] Tkachenko O.I., Tkachenko O.A., Tkachenko K.O. Designing complex intelligent systems on the basis of ontological models. The Third International Workshop on Computer Modeling and Intelligent Systems, 2020, pp. 266-277.

Ситуационно-семантическое моделирование системы мониторинга рисков на городском транспорте

Ткаченко К.А., Ткаченко О.И., Ткаченко А.А.

В статье рассматриваются актуальные проблемы моделирования мониторинга рисков на городском транспорте. Этот мониторинг предполагает определение состояния транспортных объектов, а также процедуры минимизации последствий, вызванных наступлением рисков.

В статье рассматриваются проблемы эффективного создания системы мониторинга рисков на городском транспорте на основе использования соответствующих моделей.

В статье предлагается моделировать данную систему на основе ситуационно-семантической модели. Предлагаемый подход будет способствовать признанию рисков и выработке управленческих решений по устранению их последствий.

Предлагаемая ситуационно-семантическая модель позволяет: прогнозировать поведение сложных транспортных объектов и объектов транспортной инфраструктуры; учитывать возможность появления новых транспортных объектов и процессов для обеспечения их функционирования в условиях минимизации возможных рисков; адекватно реагировать на локальные и глобальные факторы воздействия на транспортные объекты; динамически изменять структуру системы; учитывать новые данные для прогнозирования развития и совершенствования соответствующих транспортных объектов; прогнозировать развитие процессов, обеспечивающих минимизацию рисков и их последствий.

Received 01.06.2021

Development an ontology based intelligence search system for goods for an online store on the CS-Cart platform

Gleb Guskov, Vladimir Zarayskiy, Aleksey Filippov, Anton Romanov
Information systems department
Ulyanovsk State Technical University
Ulyanovsk, Russia
guskovgleb@gmail.com, v-zar@list.ru, al.filippov@ulstu.ru, romanov73@gmail.com

***Abstract*—The article presents the implementation of a intelligence search system for an CS-Cart platform online store. The system is based on the formation of a domain products ontology. The system assumes the processing and expansion of the user request. This article describes the structure of an OWL ontology. The article describes the algorithm for processing custom queries, as well as the results of experiments.**

***Keywords*—knowledge base, ontology, smart search, e-commerce.**

I. Introduction

The term "information retrieval" was first introduced into scientific circulation by the American mathematician Calvin Moers in 1951. In state standard 7.73-96, information retrieval is actions, methods and procedures that provide the selection of certain information from a data array.

The problems of modern information retrieval are reflected in the work processes of the online store. It's that users can not quickly and accurately find the desired item and leave the online store.

Currently, the platform for the CS-Cart online store by default search by matches in the name or description of the product with the entered user request.

In this research, process of implementing a intelligence search system will be considered. Intellectual search system is part of an online store based on CS-Cart platform, which allows you to expand the capabilities of the standard functionality of the search engine.

Any Internet resource must have a search panel, as it is an integral part of the graphical user interface. Most systems today are based on the Boolean search model.

Boolean variables get the value "true" or "false" depending on the occurrence or not occurrence of the query terms in the corresponding document.

The search model is foundation for search technology in a particular system . The search model is a combination:

- the representations of documents form;
- an approach of forming representations of search queries;
- type of criterion of relevance of documents.

There are also systems that use external services to search through an array of data. Unfortunately, in this case, there is no way to control the behavior of the search engine, and the it's internal mechanisms are not available.

The growth of the World Wide Web has exposed the flaws classic search and indexing methods. Information retrieval can be based on a semantic structure. The Semantic Web allows connections to be created between terms, giving meaning to information and helping the machine integrate new hidden knowledge.Information retrieval can be based on a semantic structure, namely the relationship between information available on the network [1].This problem complicates the translation of user queries into an executable query for the information system, which leads to the loss of some of the relevant results when searching. Therefore, the identification of the semantics of search queries plays an important role in the development of search engines. This area of research is called semantic search. Semantic search aims to deliver contextually relevant results to the user by understanding the purpose and meaning of the query provided.

Most often, users know what they are looking for, but do not know how to formulate a search query. Users formulate a request based on the key characteristics or metadata of the sought object [2]. The online store provide to categorize products and set individual characteristics for each product. Presenting products, categories, characteristics and relationships between them as semantic network, and apply an inference engine to form the result of a search query, quality of the search engine will increase. The use of semantic search for an online store contains scientific interest.

On the basis of the ontological approach, there are platforms for creating information systems in which the advantages of semantic technologies are available [3], [4], [5].

Since with the help of ontology it is possible to formalize any subject area in the form of a set of statements and terms [6], that is, the possibility of using ontology for e-commerce. Ontological representation of knowledge will help customers can find the right products and make purchasing decisions. Ontologies can be applied in a search system to expand and adjust user queries.

II. ONTOLOGY BASED SEARCH SYSTEM

The ontological approach to knowledge representation has many successfully solved problems in various industries. For example, the task of classifying documents in an electronic archive can also be solved using the ontological approach [7] or the task of finding similar program products [8].

The structure of the ontology classes is shown in “Fig. 1”. The concepts of characteristics and categories are stored in the "features" and "categories" classes, respectively. The first concept contains the names of the characteristics, and the second contains the names of the categories in the form in which these terms are presented in the online store. Product concepts are stored in the products class. The filling of the ontology with concepts occurs in an automated mode.

The intelligence search algorithm contains 3 stages:

- Query string preprocessing;
- Formation of logical conditions and definition of classes;
- Synchronization of the inference engine and getting a list of products.

A. *Query string preprocessing*

The input query string is split into tokens and stop words are removed from the resulting array of terms. Bigrams are formed from the received tokens. Then, using an approximate string comparison [4], terms for autocomplete are compared with a bigrams list. The algorithm for the approximate comparison of strings is based on the calculation of the Levenshtein distance [5]. Comparing two strings with each other, the comparison function produces a number from 0 to 100, which indicates the degree of similarity. From the beginning, a partial comparison takes place, which looks for a match with a bigram in the entire set of autocomplete terms. The resulting set of matches is re-checked against the bigram using a word-by-word comparison. Thus, a list of concepts that exist in the ontology is distinguished.

B. *Forming logical conditions and determine classes*

At the second stage of the formation of logical conditions, it is checked to which class the element of the list of found concepts belongs. For each characteristic from the general set of conditions, a new generated condition of the "hasFeature value currentFeature" format is added. Where "hasFeature" is an entity of type "ObjectProperty", which allows you to create a relationship between the concept of the product and the concept of the characteristic "currentFeature". A set of conditions for characteristics and categories form equivalence conditions for four classes: two for the logical "AND" condition and two for the logical condition "OR". The sets of conditions are also combined using the logical operators "AND" and "OR", and form the value of the equivalence of four classes:

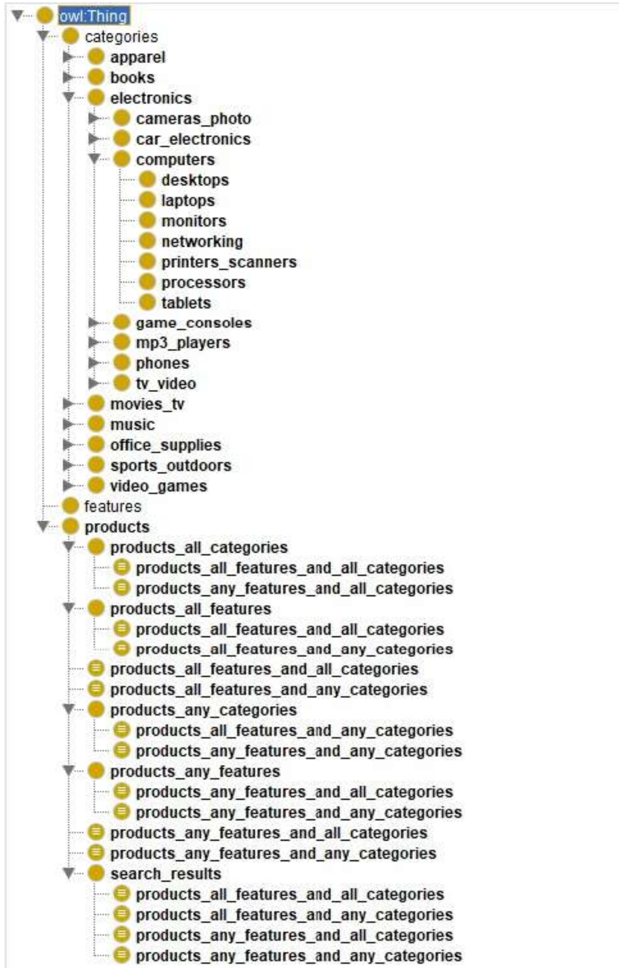


Figure 1. Ontology class structure.

- products_all_categories - products belonging to all categories from a set of conditions;
- products_all_features - products that have all the characteristics from a set of conditions;
- products_any_categories - products belonging to at least one category from a set of conditions;
- products_any_features - products that have at least one characteristic from a set of conditions.

These classes have subclasses that represent the intersections of the sets of found concepts:

- products_all_features_and_all_categories - products that have all characteristics and belong to all categories from a set of conditions;

- products_all_features_and_any_categories - products that have all characteristics and belong to at least one category from a set of conditions;
- products_any_features_and_all_categories - products that have at least one characteristic and belong to all categories from a set of conditions;
- products_any_features_and_any_categories - products that have at least one characteristic and belong to at least one category from the set of conditions.

C. Information system developed during the research

The sequence diagram shown in “Fig. 2” is the best way to describe how the system works. The diagram shows the process of searching for goods in the system being developed in general form. The diagram consists of the following elements: 1. User request; 2. Two repositories of information (database and knowledge base); 3. Two developed program modules (search controller and search service); 4. An auxiliary module required for preprocessing a request.

As part of the research, a web service in python was developed using the REST API of the flask framework and a module for an online store on the CS-Cart platform version "4.11.2 Ultimate" in php. The ontology is stored in an owl file on the web service server. Interaction with the ontology is carried out using the owlready2 [9] library for python. Autocomplete words are stored as a list in a json file.

III. Synchronizing the inference engine and forming list of products

The third stage starts the synchronization of the inference engine. Subclasses that allow you to identify the intersection of the sets of found concepts contain products that match the conditions. It is necessary to traverse all four classes in order from more precise to more complete list. More precise is a class that contains products that satisfy all the conditions, connected by the "AND" operator. And less accurate, but more complete, through the "OR" operator. The HermiT inference engine was used to synchronize the classes.

IV. Experimental results

Experiments were run on demo store data, which contains 83 categories, 300 products, and 35 types of characteristics. The time for filling the ontology with all concepts and relations between them ranged from 0.2 to 1 second, depending on the amount of data. To check the synchronization time of the inference engine on the same user request, larger data sets were formed based on the existing ones. The library for interaction with ontology allows to automatically launch two inference machines: HermiT and Pellet. The results of the execution time of queries are presented in ”Table. I”.

The results of the experiments, HermiT is faster at coping with inference almost twice as fast as Pellet.

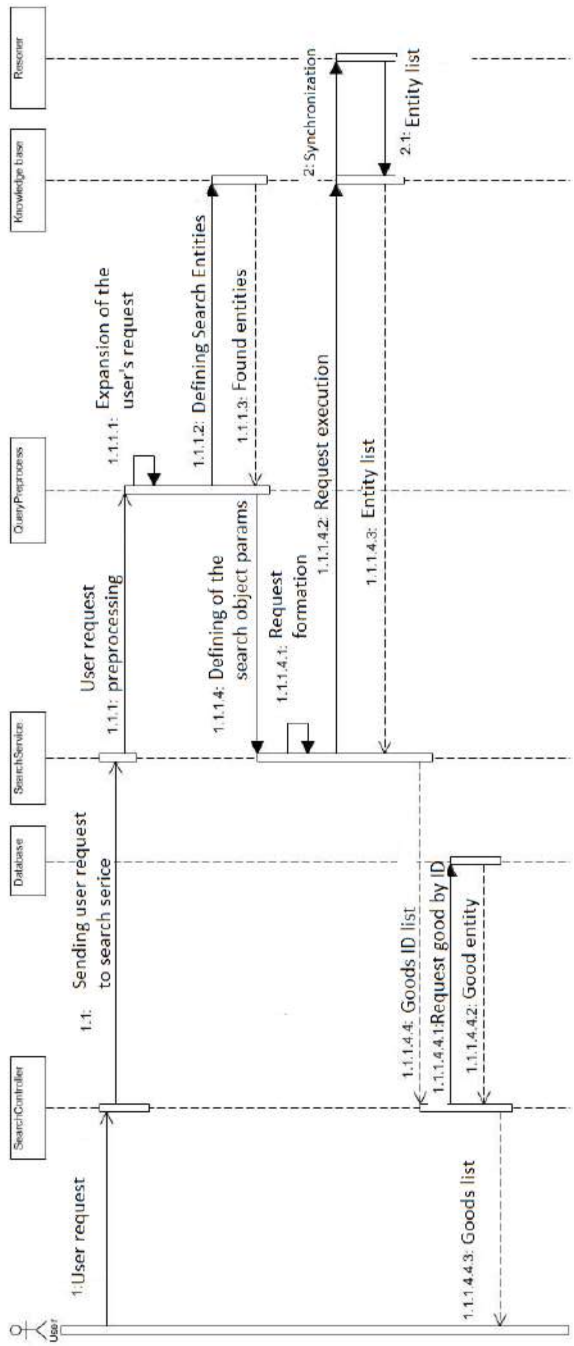


Figure 2. System sequence diagram.

Table I
THE RUNNING TIME OF THE ALGORITHM WITH DIFFERENT REASONERS

Total goods	***Founded goods***	***HermiT***	***Pellet***
300	10	4,5 sec	9 sec
600	20	15,8 sec	18,2 sec
1000	30	26,5 sec	46,3 sec
2000	60	72,1 sec	180 sec
5000	150	529 sec	1010 sec

The larger the total number of products included in the ontology, the longer the search takes. The results of executing queries by the intelligent search system(ISS) and the standard search algorithm(SSA) based on word coincidence are presented in ”Table. II’.

Table II
COMPARISON OF QUERY RESULTS

Query	Goods count		Relevant goods count	
	ISS	*SSA*	*ISS*	*SSA*
blue medium t-shirt	1	0	1	0
apple iphone 128gb	10	0	2	0
windows 7 premium	9	0	8	0
monitor ips fullhd	29	0	7	0
huawei notebook	18	0	16	0
huawei	3	3	3	3
games	1	11	1	5

The results of the experiments, the standard search algorithm cannot find any product based on user queries consisting of several words. When searching for the word "huawei", the standard search system found all relevant products, since this word was present in the name of the products. When searching for the phrase "huawei notebook", the standard search search system did not find the products, as it was in the previous query, since this phrase is missing in the name or description of the goods. The intelligent search system was able to find the same products for the query "huawei" as the standard search algorithm, since in this case, among the characteristics of the products found, there was a brand characteristic with the value "huawei". When performing the query "huawei notebook" among the products that have the brand "huawei" and the type of notebook "notebook", there was no combination of such characteristics. However, there are products that have either the first or the second characteristic. Therefore, the system strives to offer the user products that may be of interest to him, and not leave him with an empty search result. The average accuracy of the smart search system according to the results of the experiments from Table 2 was 70 percents.

V. CONCLUSION

As a result of the work, an intelligent search system was developed for an online store based on the CS-Cart platform using ontology. The experiments carried out show that the developed system gives out products with characteristics and categories by which the user searches, in contrast to the standard search. The standard search algorithm does not find products for the query, for which smart search returns products. The running time of the algorithm with the HermiT inference engine is less than 10 seconds for a small number of products, which can be used for online stores with a specific subject area. To increase the accuracy of the algorithm, you can add additional information about the product to the ontology and modernize the query preprocessing process. To reduce the query time, you need to add an inference engine written in python to the library.

ACKNOWLEDGMENT

This work is supported by the Russian Foundation for Basic Research (project No. 19-47-730006).

REFERENCES

[1] Saravanaguru K., Shekhar M. Semantic Web Search based on Ontology Modeling using Protégé Reasoner. School of Computing Science and Engineering, 2013, pp. 10

[2] Kravchenko Yu.A., Markov V.V., Novikov A.A. Cemanticheskii poisk v semantic web [Semantic search on the semantic web]. Izvestiya YuFU. Tekhnicheskie nauki [Izvestia Yufu. Technical science], 2016, 11p. (In Russ)

[3] Shunkevich D.V., Taberko V.V., Ivaniuk D.S., Pupena O.N. Ontological Approach for Standards Development Within. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], 2013, pp. 31-42.

[4] Yarushkina N.G., Filippov A.A., Moshkin V.S., Guskov G.Yu. Development of a System Based on a Fuzzy Knowledge Base Filled from Various Formats Documents. Recent Developments and the New Direction in Soft-Computing Foundations and Applications. Studies in Fuzziness and Soft Computing, vol 393. Springer, Cham. https://doi.org/10.1007/978-3-030-47124-8_13

[5] Filippov A.A., Yarushkina N.G., Moshkin V.S. Development of a Technological Platform for Knowledge Discovery. Computational Science and Its Applications – ICCSA 2019. ICCSA 2019. Lecture Notes in Computer Science, vol 11619. Springer, Cham. https://doi.org/10.1007/978-3-030-24289-3_39.

[6] Lukashevich N.V. Tezaurusy v zadachakh informatsionnogo poiska [Thesauri in information retrieval problems], Moskow Izd-vo Moskovskogo universiteta [Moscow Publishing house of Moscow University], 2011, 512 p. (In Russ)

[7] Koval A.R., Zarubin A.A., Moshkin V.S. Classification of Text Documents of an Electronic Archive Based on an Ontological Model. Intelligent Data Engineering and Analytics, 2021. https://doi.org/10.1007/978-981-15-5679-1_56.

[8] Dudarin P.V., Yarushkina N.G., Guskov G.Yu. Two Phase Approach to Detection of Software Projects with Similar Architecture Based on Clustering and Ontological Methods. Recent Developments and the New Direction in Soft-Computing Foundations and Applications. Studies in Fuzziness and Soft Computing, vol 393. Springer, Cham. https://doi.org/10.1007/978-3-030-47124-8_12.

[9] LAMY Jean-Baptiste Owlready2 Documentation Owlready2. at: https://owlready2.readthedocs.io//en//v0.31// (accessed 2021, May)

Разработка системы интеллектуального поиска товаров на основе онтологий для интернет-магазина на платформе CS-Cart

Гуськов Г. Ю., Зарайский В.И.
Филиппов А.А., Романов А.А.

В статье представлены осбенности реализации системы интеллектуального поиска для интернет-магазина на платформе CS-Cart. Предлагаемая система основана на формировании онтологии предметной области к которой относятся товары. В рамках исследования рассматриваются процессы системы по обработке и расширению запроса пользователя. В статье предствалена структура онтологии в формате OWL. В работе описан алгоритм обработки пользовательских запросов, а также представлены результаты экспериментов.

Received 31.05.2021

Semantic Approach to NLP Problem Solving

Aliaksandr Hardzei
Minsk State Linguistic University
Minsk, Belarus
Email: alieks2001@yahoo.com

***Abstract*—On the basis of the third edition of the second version of Theory for Automatic Generation of Knowledge Architecture (TAPAZ–2), a new approach to the semantic markup of an event and the syntax formalization of Chinese, English and Russian sentences is proposed [1].**

***Keywords*—combinatory semantics, semantic markup, semantic case, semantic classifier, knowledge graph, role list of individs, subject, object, action, macroprocess, specialized process, world model, TAPAZ-algebra, TAPAZ-unit**

I. Introduction

Almost simultaneously with the advent of the Internet, the idea of building algorithms for relevant information retrieval according to strictly specified semantic rules was born in the American scientific community. In 2001, the creator of the World Wide Web and the current head of the W3C Consortium Timothy Berners-Lee, together with James Hendler and Ora Lassila, published a keynote article in Scientific American "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities" [1]. The article was focused on the development of semantic technologies for searching and processing information on the Internet and through the Internet: "To date, the World Wide Web has developed most rapidly as a medium of documents for people rather than of information that can be manipulated automatically. By augmenting Web pages with data targeted at computers and by adding documents solely for computers, we will transform the Web into the Semantic Web. Computers will find the meaning of semantic data by following hyperlinks to definitions of key terms and rules for reasoning about them logically. The resulting infrastructure will spur the development of automated Web services such as highly functional agents. Ordinary users will compose Semantic Web pages and add new definitions and rules using off-the-shelf software that will assist with semantic markup" [1: 36].

In fact, the authors offered an alternative to the statistical methods of data processing that were gaining popularity in Artificial Intelligence: "The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users. Such an agent coming to the clinic's Web page will know not just that the page has keywords such as "treatment, medicine, physical, therapy" (as might be encoded today) but also that Dr. Hartman works at this clinic on Mondays, Wednesdays and Fridays and that the script takes a date range in yyyy-mm-dd format and returns appointment times. And it will "know" all this without needing artificial intelligence on the scale of 2001's Hal or Star Wars's C-3PO" [1: 37], and then: "For the semantic web to function, computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning. Artificial-intelligence researchers have studied such systems since long before the Web was developed. Knowledge representation, as this technology is often called, is currently in a state comparable to that of hypertext before the advent of the Web: it is clearly a good idea, and some very nice demonstrations exist, but it has not yet changed the world. It contains the seeds of important applications, but to realize its full potential it must be linked into a single global system" [1: 37].

II. TOWARDS COMBINATORY SEMANTICS

Since with the help of N. Chomsky's transformational grammar, on which context-dependent and context-free grammars were built, and that became the basis for higher-level object-oriented programming languages, such as Java, C++, C#, etc., it was possible only with varying success to formalize syntax, but not semantics, the views of American scientists turned to the semantic cases of Ch. Fillmore [2–5], Jackendoff's early work [6, 7] and Stowell's "theta-grids" [8]. It is curious that at the same time the formal grammar of R. Montague [9] with its PTQ (Proper Treatment of Quantification in Ordinary English) and lambda abstraction was pushed into the background, although it was this grammar in a number of cases that generalized in terms of mathematical logic the achievement of generative semantics by G. Lakoff [10] and the interpretive semantics of R. Jackendoff. It is also curious that the semantic syntax of L. Tesnière [11] with verb nodes of actants and syrconstants, in fact

[1] The article develops the scientific provisions formulated by the author in the following works: A. Hardzei, "Plagiarism Problem Solving Based on Combinatory Semantics". *Communications in Computer and Information Science (CCIS).* Switzerland: Springer Nature Switzerland AG, 2020, vol. 1282, pp. 176–197. Available: https://link.springer.com/book/10.1007%2F978-3-030-60447-9 and A. Hardzei, "Semantic Markup of the Event and its Display by Means of the Chinese and Russian Languages". *Foreign Languages in Tertiary Education*, 2021, no. 2(57), pp. 5–26.

– analogs of the semantic cases of Ch. Fillmore, was also taken out of the brackets – the works of L. Tesnière were published for 30 years earlier, and the monograph by V. V. Martynov "Cybernetics. Semiotics. Linguistics" with a prototype of the Universal Semantic Code (USC) and a description of the roles of signs in the nuclear semantic string *subject – action – object* (SAO) – for 2 years earlier than the case grammar of Ch. Fillmore [12]. Note that at present linguistics has only one synthetic (sequentially deductive and procedural) model of language – the Panini grammar, dating from the 5th century BC, in which 3959 short sutras (rules) totally described the generation, construction and transformation of all Sanskrit units, starting from the phonetic-phonological level and ending with the semantic-syntactic level [13–15]. It is still not clear what formalisms were used as the basis for such an accurate description of a natural language and how it was possible to achieve this in such ancient times, just as, for example, it is not unknown, what kind of mathematics were used to collect the hexagrams of the ancient Chinese "Book of Changes" (易經), the analysis of which led Leibniz to the idea of binary calculus, that became the basis of modern computing, one thing is clear – European linguistics, first of all, French, in a hidden form borrowed a number of postulates of Panini's grammar, in particular, that the case is not so much morphological as semantics-syntactic category – the founders of structural linguistics, of course, knew about Panini's grammar, the departments of Sanskritology were in many European universities. However, we emphasize that semantic cases were important, but not the only achievement of Panini. They were calculated by some algebra and organically fitted into the entire architecture of grammar. Without this algebra, it was possible, albeit with difficulty, to translate Panini's grammar from one language to others, but it was impossible to describe other languages, like Panini: the formalization of languages and translation are rather different tasks. Therefore, V. V. Martynov started looking for such an algebra.

The first version of USC was published at 1974, 1977 – the second, 1984 – the third, 1988 – the fourth, 1995 – the fifth, 2001 – the sixth [16–21]. From version to version the algebraic apparatus and the list of semantic primitives were improving. Thus, the list of tasks to equip computer with encyclopedic knowledge bases was narrowed, and finally the list consists of five components:

1. "To calculate semantic primitives, i.e. semantically irreducible kernel words and define rules of their combinatorics.
2. To define the necessary and sufficient set of formal characteristics constituting 'dictionary entry'.
3. **To define a set of semantic operations for calculating a subject domain of any kind.**
4. To propose heuristic teaching rules to work with the system.
5. To build a system of mutual references based on semantics" [21: 42].

In 1993 achievements in the approach allowed the researchers of the center "Semantics" of Minsk Linguistic State University, headed by V. V. Martynov, to begin an intensive research of ways to expand the basic semantic classifier to the encyclopedic knowledge base. In 1994, the first procedure of calculating the subject domains in the form of a directed graph of complex strings was proposed by A. Hardzei [22]. Use of the procedure has required the establishment of a one-to-one (vector) transition between actions in basic semantic classifier and has led to the creation of the automatic generation of knowledge architecture theory (TAPAZ) which was founded on: the formal theory; the semantic counterpart; the set of macroprocesses (actions) as semantic primitives; the algorithm defining roles of individs, and the knowledge graph for searching processes through macroprocesses (see Fig. 1) [23, 24].

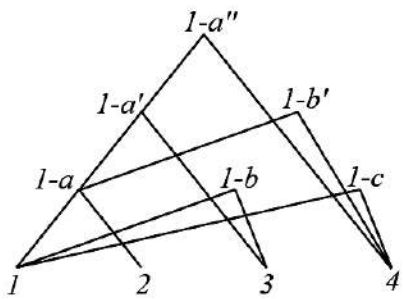


Figure 1. TAPAZ Knowledge Graph (the Semantic Classifier Graph). where: *1* – active macroprocess; *2, 3, 4* – clarifying macroprocesses; *1-a, 1-b, 1-c* – derivative processes with *1-a* as the active derivative process; *1-a'* and *1-b'* – derivative processes of the second level with *1-a'* as the active derivative process of the second level; *1-a''* – the active derivative process of the third level.

For example, the macroprocess '*restore*' may be considered as a set and the processes '*treat*', '*repair*', '*adjust*' as its subsets. Such subsets represent isomorphism of subject domains and create a knowledge structure where subsets of processes fill cells of the knowledge structure with a concrete content [1]. TAPAZ–2 as the new version of the Theory for Automatic Generation of Knowledge Architecture differs from the previous version in several ways: simplified algebraic apparatus, increased number of rules for interpretation of the standard superposition of individs, and minimized semantic calculus. The number of operations with the strings of semantic code are reduced to two and it is now the algebra type:

$$A = \langle M, *, - \rangle \tag{1}$$

[1]For a detailed description of the TAPAZ Knowledge Graph, see: A. Hardzei, A. Udovichenko, "Graph of TAPAZ–2 Semantic Classifier". In: V. V. Golenkov et al. (eds.) CONFERENCE 2019, *Open Semantic Technologies for Intelligent Systems (OSTIS)*. Minsk: Belarussian State University of Informatics and Radioelectronics Publ., 2019, iss. 3, pp. 281–284. Available: http://tapaz.by.

where: M is a set of elements, '*' is operation of superposition, '–' is operation of extension [1].

Examples of TAPAZ formulas:

$$((((((((((((((X \ast \overline{X}) \ast \overline{\overline{Y}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{Z}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{Z}}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{W}}; \tag{45}$$

$$((((((((((((((X \ast \overline{X}) \ast \overline{\overline{X}}) \ast \overline{Y}) \ast \overline{\overline{Y}}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{Z}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{W}}; \tag{46}$$

$$((((((((((((((X \ast \overline{X}) \ast \overline{\overline{\overline{Y}}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{\overline{Z}}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}; \tag{47}$$

$$(((((((((((((((X \ast \overline{X}) \ast \overline{\overline{Y}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{Y}}) \ast \overline{\overline{Z}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{W}}) \ast \overline{W}; \tag{48}$$

$$(((((((((((((((X \ast \overline{X}) \ast \overline{\overline{Y}}) \ast \overline{\overline{Y}}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{Z}}) \ast \overline{\overline{Z}}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{Z}}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{W}}) \ast \overline{W}; \tag{49}$$

$$(((((((((((((((X \ast \overline{X}) \ast \overline{\overline{X}}) \ast \overline{Y}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{Z}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{W}}) \ast \overline{W}; \tag{50}$$

$$(((((((((((((((X \ast \overline{X}) \ast \overline{\overline{Y}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{Z}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W; \tag{51}$$

$$((((((((((((((((X \ast \overline{X}) \ast \overline{\overline{\overline{Y}}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{\overline{Y}}}) \ast \overline{\overline{\overline{Z}}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W; \tag{52}$$

$$(((((((((((((((X \ast \overline{X}) \ast \overline{\overline{Y}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{Z}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{Z}}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{W}} \ast \overline{W}) \ast W; \tag{53}$$

$$((((((((((((((((X \ast \overline{X}) \ast \overline{\overline{\overline{X}}}) \ast \overline{\overline{Y}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{\overline{Z}}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W; \tag{54}$$

$$(((((((((((((((((X \ast \overline{X}) \ast \overline{\overline{\overline{Y}}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{\overline{Z}}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}; \tag{55}$$

$$((((((((((((((((((X \ast \overline{X}) \ast \overline{\overline{\overline{Y}}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{\overline{Y}}}) \ast \overline{\overline{\overline{Z}}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}; \tag{56}$$

$$((((((((((((((((((X \ast \overline{X}) \ast \overline{\overline{Y}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{Z}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{Z}}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{W}}) \ast \overline{W}) \ast W) \ast \overline{W}; \tag{57}$$

$$((((((((((((((((((X \ast \overline{X}) \ast \overline{\overline{\overline{X}}}) \ast \overline{\overline{Y}}) \ast \overline{Y}) \ast Y) \ast \overline{Y}) \ast \overline{\overline{\overline{Z}}}) \ast \overline{Z}) \ast Z) \ast \overline{Z}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}) \ast \overline{\overline{\overline{W}}}) \ast \overline{W}) \ast W) \ast \overline{W}. \tag{58}$$

TAPAZ–2 is a tool for generating a world model in a form suitable for Natural Language Processing in systems of Artificial Intelligence. The Intellectual Knowledge Base built in a computer combines the Semantic Classifier – a final ordered (vector) set of semantic primitives (actions and roles of individs) and the Semantic Ontology – an algorithm for generating new sense units based on the original set of primitives, presented in the form of TAPAZ Knowledge Graph.

An Intelligent (Expert) Search System based on the TAPAZ–2 Semantic Classifier may consist of an intelligent search engine that selects and reviews content on a given topic from the Internet, and a dialog user interface that allows the system to process user requests and transform them in the canonized text corresponding to the machine-readable the World Model, and the user will confirm whether this conversion was performed correctly, and if not, then offer his own decoding through the Semantic Classifier.

This Intelligent Search System can be used to solve various problems, including the task of automatic identification of semantically equivalent fragments of text documents, which will be discussed below.

The main components of this Intelligent Search System are: an online content monitoring module with adequate crawler and stapler; automatic lexical analysis module with a tagger on a semantic (Parts of Language), not on a morphological (Parts of Speech) basis; dynamic syntactic analysis module with a recursive reconstruction algorithm (parser) for sentence string elaborated in combinatory semantics technology; a module for direct and reverse conversion of syntactic expressions into TAPAZ–2 algebraic formulas, as well as the Intellectual Knowledge Base (IKB) consisting of TAPAZ-units (IKB-taxonomy) assembled in the order indicated by the Knowledge Graph of the Semantic Classifier (IKB-ontology). The knowledge base taxonomy also serves as a corresponding subject domain semantic dictionary during automatic lexical analysis.

III. ROLE LIST OF INDIVIDS

Combinatory semantics studies the linguistic mapping of the dynamics of individs' roles in an event. Its founder, as we consider, is Z. Harris, who put forward the nuclear semantic string *subject – action – object* as a starting point of formalizing sentences [25]. Research in this direction was continued at Minsk Semantic School under the guidance of V. V. Martynov and A. Hardzei. Combinatory semantics should not be confused with combinatorial semantics, which studies co-occurrence of signs using statistical methods, the founder of which is also Z. Harris [26].

The updated vector role list of individuals ordered by the TAPAZ-algebra (The TAPAZ Role List) [2] is the following set:

subject *(initiator → spreader → inspirer → creator) →* ***instrument*** *(activator → suppressor → enhancer → converter) →* ***mediator*** *(landmark → locus → carrier → adapter → acceptor → stock → separator → material → model → retainer → resource → stimulus → regulator → chronotope source → indicator) →* ***object*** *(coating → hull → interlayer → kernel) →* ***product*** *(billet → semi-product → prototype → end item),*

were: ***subject*** – the originator of the action, varieties of the subject: *initiator* – initiates the action, *spreader* – spreads the action, *inspirer* – involves into the action, *creator* – completes the action by making a product from the object; ***object*** – the recipient of the action, varieties of the object: *coating* – the outer insulation of the individ's shell, *hull* – the individ's shell, *interlayer* – the inner insulation of the individ's shell, *kernel* – the core of the individ; ***product*** – the result of the subject's impact (action) on the object (the individ adapted to a given role in a new action), varieties of the product: *billet* – the object turned into a raw material, *semi-product* – the product half-made from raw materials, *prototype* – the prototype product, *end item* – the finished product; ***instrument*** – the performer of the action, the closest individ to the subject, varieties of the instrument: *activator* – directly affects the mediator, *suppressor* – suppresses the resistance of the mediator, *enhancer* – increases the effect on the mediator, *converter* – converts the mediator into the instrument; ***mediator***, i.e. the mediator of the action – the closest individ to the object; varieties of the mediator: *landmark* – orientates the impact on the object, *locus* – the closest environs of the object partially or completely surrounding the object that localizes the object in space and thereby containing (enclosing) it, *carrier* – carries the object, *adapter* – adapts the instrument to affect the object,

[1]For a detailed description of the new version of TAPAZ-algebra, see: A. Hardzei, Theory for Automatic Generation of Knowledge Architecture: TAPAZ-2. Rev. English edn. Minsk: RIHE, 2017, 50 p. Available: http://tapaz.by.

[2]To date, each of the 32 TAPAZ-algebra role formulas have been deciphered. M. I. Svyatoshchik provided all possible assistance in the interpretation of some formulas of the TAPAZ Role List [27].

acceptor – catches the object, *stock* – the object collected for processing, *separator* – sorts the object, *material* – the object used as a raw material for making a product, *model* – the physical or informational original sample for making a product from the object, *retainer* – turns a variable locus of the object into a constant one, *resource* – feeds the instrument, *stimulus* – reveals the parameter of the object, *regulator* – serves as an instruction in making a product from the object, *chronotope* – localizes the object in time, *source* – provides instructions for the instrument, *indicator* – displays a parameter of impact on the object or a parameter of the product as the result of subject's impact on the object.

The algorithm for extracting specialized terminology from the Internet content of selected subject domain and constructing TAPAZ-units assumes answers to the key questions:

Who? With which tool? In relation to whom / what? In what place? Arriving on what? Adjusting by what? Accepting by what? Stocked up (on) what? Selecting by what? Making of what? Following what example? Fixing by what? Spending what? Stimulating with what? Guided by what? In what period? Knowing wherefrom? At what parameter? Affecting who / what? Produces whom / what?

To facilitate the work of experts in the construction of TAPAZ-units on the basis of the updated TAPAZ Role List for Chinese, English and Russian sectors of IT-Industry, a new ExpertTool version 1.0.0.0 was developed by the efforts of software engineer A. A. Matsko (see Fig. 2).

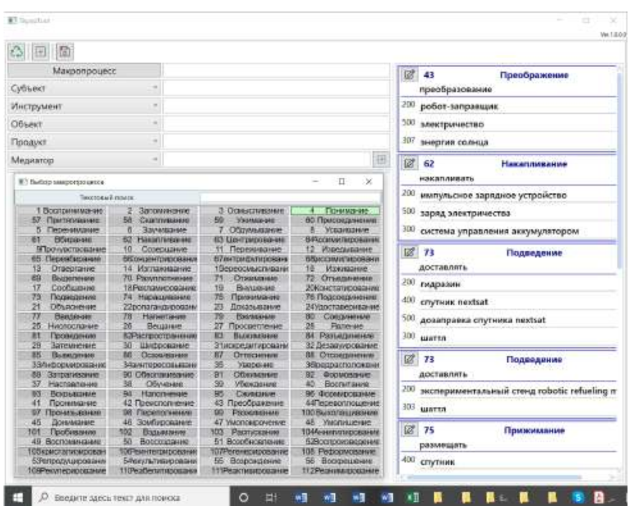


Figure 2. The working window of the software tool with an expanded tab.

There must be a one-to-one correspondence not only between noun phrases and thematic roles, as N. Chomsky mentioned [28], but a one-to-one correspondence between roles of individs in the event, parts of the sentence, mapping this event [1], and parts of the language [2], playing relevant roles in this sentence, otherwise, we will not be able to implement machine learning algorithms, such as, for example, an artificial neural network or a random forest and thereby provide automatic semantic markup of texts collected in the Knowledge Base. TAPAZ Semantic Dictionary consists of subject domains' TAPAZ-units assembled in the order indicated by TAPAZ Knowledge Graph. TAPAZ-units simultaneously form the taxonomy of the Intellectual Knowledge Base. Initially, TAPAZ-units are manually assembled by experts until the training data is sufficient to implement machine learning algorithms.

IV. EXAMPLES OF SEMANTIC MARKUP

Media report: *Yesterday at 10:30 am the Belarusian spacecraft was launched from the Baikonur cosmodrome.*

Text preprocessing by an expert using the TAPAZ software tool: *Specialists of the Federal State Institution "Roskosmos" on July 22, 2012 at 10:30 am from the Baikonur cosmodrome using the "Soyuz" carrier rocket into the near-earth orbit delivered the Belarusian spacecraft;* specified remote sensing process – *deliver,* TAPAZ macroprocess – *(73) approach,* subject – *the Federal State Institution "Roskosmos",* instrument – *the "Soyuz" carrier rocket,* object – *the Belarusian spacecraft,* locus – *the Baikonur cosmodrome*, landmark – *the near-earth orbit,* chronotope – *July 22, 2012; 10:30 am.*

Technical description: *The Belarusian spacecraft is similar to the Russian one.*

Stable expression *the Belarusian spacecraft* plays the role of a grammatical subject, but at the same time it is not mapped into a subject, because the Belarusian spacecraft does nothing with the Russian spacecraft. Correct semantic reconstruction of the role structure of the sentence: *What does someone do with information about the Belarusian spacecraft? Compares it with information about the Russian spacecraft and states the identity;* TAPAZ macroprocess – *(20) state,* subject – virtual *someone*, object – *information about the Belarusian spacecraft*, landmark – *information about the Russian spacecraft,* product – *coincidence of information.*

Let us show the solution to the problem of attributive 的 *de* when parsing of a Chinese sentence by means of TAPAZ technology: 皮球是红的 *Píqiú shì hóngde (The ball is red)*. The fact is that the Chinese sign 红 *hóng* (red) in the language system denotes a property, and in the sentence plays a specialized role of a grammatical attribute, therefore it does not require 的 *de*, which

[1] Syntactical rules for dynamic syntactic analysis module with a recursive reconstruction algorithm (parser) for sentence string is described in [29, 30].

[2] For definitions of Parts of Language, their paradigm, and semantic delineation procedures, see: [31, 32].

transforms nominal semantics into attributive, for example: 石头 *shítou (a stone)* → 石头的 *shítoude (stony)*. The presence of 的 *de* in the sentence 皮球是红的 indicates the omission of the grammatical direct object 东西 *dōngxi (thing)*, i.e. ***the ball is a red object, not the redness***. Indeed, from the classical logic point of view, identity should be established between homogeneous objects: an individ and another individ, or a feature of an individ and another feature of an individ. In this case, the Chinese language turns out to be more accurate in representing the world model than English or Russian. Correct parsing of the sentence: 皮球 *píqiú (the ball)* is a grammatical subject, 是 *shì (to be)* is a grammatical predicate, 红的 *hóngde (red)* is a grammatical attribute, 东西 *dōngxi (thing)* is a reconstructed grammatical direct object. However, the role structure of the event will be different:

What does someone do with the color information about the ball? Compares it with information about the red color and states their identity; TAPAZ macroprocess – *(20) state,* subject – virtual *someone,* object – *color information about the ball,* landmark – *information about the red color,* product – *coincidence of information.*

Due to the fact that semantic primitives lie at the core of the language system, one should achieve the minimum depth of recursive reconstruction, observing the strict requirements of **order, clarity and simplicity,** when the missing parts of a sentence are restored at the first or, at most, at the second step, because the deep recursion and the complex reconstruction create a high probability of error.

Another example: 苹果多少钱两斤？*Píngguǒ duōshao qián liǎng jīn? How much is a kilogram of apples?* → Recursive reconstruction: 苹果[是]多少钱，[少要买]两斤[苹果]。*Píngguǒ [shì] duōshao qián, [wǒ yào mǎi] liǎng jīn [píngguǒ].* Word for word translation: *Apples [are] how much money, [I need to buy] one kilogram [apples].* 苹果 *píngguǒ (apples)* is the 1st grammatical subject, 是 *shì (are)* is the reconstructed 1st grammatical predicate, 钱 *qián (money)* is the 1st grammatical direct object, 多少 *duōshao (how much)* is the 1st grammatical quantitative attribute of the 1st grammatical direct object; 少 *wǒ (I)* is the reconstructed 2nd grammatical subject, 要 *yào (need)* is the reconstructed modal component of the 2nd grammatical predicate, 买 *mǎi (buy)* is the reconstructed main component of the 2nd grammatical predicate, 两斤 *liǎng jīn (one kilogram)* is the 2nd quantitative attribute of the reconstructed 2nd grammatical direct object, which is 苹果 *píngguǒ (apples)*.

To determine the role structure of the event, it is necessary to transform the interrogative sentence into a narrative one: *The buyer asks for information about the cost of one kilogram of apples in order to compare it with information about the amount of money he has, if the information coincides, the buyer will buy apples;* 1st TAPAZ macroprocess – *(1) perceive,* object – *information about the cost of one kilogram of apples,* source – *product price of a seller;* 2nd TAPAZ macroprocess – *(24) certify,* object – *information about the cost of one kilogram of apples,* landmark – *the amount of money a buyer has;* 3d TAPAZ macroprocess – *(60) attain,* subject – *buyer,* object – *one kilogram of apples.*

We note that the power of the TAPAZ semantic markup, only in terms of the typical roles of individs, not even talking about the TAPAZ-algebra and generated by it Paradigm of Actions and the Knowledge Graph, almost 5 times exceeds the power of the closest analogue – the technology of Active Vocabulary [33], standardized and adopted by W3C Consortium in 2017 within the framework of Semantic Web project [34] and then Schema.org [35]. This technology is predominantly based on the theory of semantic cases of Fillmore and Jackendoff's early work, that we mentioned above, the inventory of which is:

"*Agent* – the initiator of some action, capable of acting with volition, and *actor* – supertype of agent which performs, effects, instigates, or controls the situation denoted by the predicate;

patient – the entity undergoing the effect of some action, often undergoing some change of state;

theme – the entity which is moved by an action, or whose location is described;

beneficiary – the entity for whose benefit the action was performed;

experiencer – the entity which is aware of the action or state described by the predicate but which is not in control of the action or state;

percept or *stimulus* – the entity which is perceived or experienced;

instrument – the means by which an action is performed or something comes about;

source – the entity from which something moves, either literally or metaphorically;

goal – the entity towards which something moves, either literally or metaphorically, and *recipient* – subtype of goal involved in actions describing changes of possession;

location – the place in which something is situated or takes place" [36].

It is not difficult to see that *experiencer, source, percept* or *stimulus, goal* and *recipient,* in fact, represent the same typical role of *landmark* in TAPAZ–2; *patient* and *theme* – the role of *object; agent* and *actor* – the roles of *subject* and *creator; beneficiary* – the role of *mediator; percept* or *stimulus* – the role of *source*; *location* – the role of *locus;* the role of *instrument* in both theories is almost the same, if one does not take into account the varieties of the instrument in TAPAZ–2. There is no any algebra in the substantiation of Fillmore's "case frames" or Stowell's "theta-grids", all these semantic categories were empirically distinguished, so it is impossible to establish their consistency, independence and completeness, thereby avoid the Russell's paradox, which inevitably arises from a mixture of theory and metatheory, language and metalanguage, semantics and metasemantics. It is for these reasons that the developers of the Semantic Web, despite titanic efforts to standardize technology, have so far failed to reduce various subject ontologies to a top-level ontology, which, as many commentators emphasize, is "critical to the whole concept" [37: 94]. This was partially acknowledged in 2006 by T. Berners-Lee himself in a joint article "Semantic Web Revisited" with N. Shedbolt and W. Hall: "The Semantic Web is a Web of actionable information – information derived from data through a semantic theory for interpreting the symbols. The semantic theory provides an account of "meaning" in which the logical connection of terms establishes interoperability between systems. This was not a new vision. Tim Berners-Lee articulated it at the very first World Wide Web Conference in 1994. This simple idea, however, remains largely unrealized" [38: 96].

TAPAZ Semantic Classifier is just such a top-level ontology. It includes the Ordered Set of Macroprocesses as Semantic Primitives (Paradigm of Actions), Role List of Individs and TAPAZ Knowledge Graph.

The Paradigm of Actions consists of informational and physical macroprocesses ordered by TAPAZ-algebra. The physical macroprocesses are shaded (see Fig. 3). Note separately that the construction of the TAPAZ Universal Problem Solver [39–41] is carried out using the TAPAZ-algebra and the TAPAZ Semantic Classifier, that is by combinatory methods, and not statistical, since all statistical methods, including artificial neural networks, only imitate the intellectual or inventive human activity, guessing the correct solutions with more or less degree of reliability, but in the fact that neural networks are able to effectively scale the solutions found by combinatory methods – we have no doubts. Moreover, it was precisely with the advent of deep learning algorithms for multilayer neural networks proposed by Geoffrey Hinton in 2007 [42, 43] that it became possible with the help of only one scientific laboratory to solve such large-scale tasks as compiling vast collections of texts of various subject domains and operating with big data, whereas before this required transnational scientific conglomerations and global interstate associations.

		I	II	III	IV
A	*a*	1 perceive	2 reflect	3 comprehend	4 understand
		attract 57	cumulate 58	constrict 59	attain 60
	b	5 adopt	6 memorize	7 contemplate	8 learn
		absorb 61	accumulate 62	center 63	assimilate 64
	c	9 feel	10 behold	11 feel profoundly	12 experience
		over absorb 65	concentrate 66	centrifuge 67	dissimilate 68
	d	13 reject	14 erase	15 rethink	16 overcome
		expel 69	decompress 70	force off 71	disassociate 72
B	*a*	17 notify	18 advertise	19 instill	20 state
		approach 73	joint 74	press down 75	connect 76
	b	21 explain	22 propagandize	23 prove	24 certify
		insert 77	pump 78	press in 79	link 80
	c	25 reveal	26 prophesize	27 enlighten	28 divine
		conduct 81	spread 82	squeeze out 83	disconnect 84
	d	29 darken	30 encode	31 discredit	32 disavow
		take out 85	pull up 86	push out 87	unlink 88
C	*a*	33 inform	34 interest	35 assure	36 predispose
		touch on 89	envelope 90	clamp 91	mold 92
	b	37 admonish	38 teach	39 convince	40 nurture
		rip up 93	fill up 94	press 95	form 96
	c	41 pierce	42 intend	43 transfigure	44 reincarnate
		penetrate 97	overflow 98	unclamp 99	eviscerate 100
	d	45 pester	46 mesmerize	47 lose conscious	48 go mad
		punch 101	uplift 102	disband 103	annihilate 104
D	*a*	49 recollect	50 recreate	51 restart	52 render
		recrystallize 105	reintegrate 106	regenerate 107	restore 108
	b	53 reproduce	54 reclaim	55 renew	56 revive
		recuperate 109	rehabilitate 110	reactivate 111	reanimate 112

Figure 3. Paradigm of Macroprocesses (Actions).
where: *A* – activation group, *B* – exploitation group, *C* – transformation group, *D* – normalization group; *a* – surroundings-shell subgroup, *b* – shell-core subgroup, *c* – core-shell subgroup, *d* – shell-surroundings subgroup; *I* – initiation raw, *II* – accumulation raw, *III* – amplification raw, *IV* – generation raw.

V. Conclusion

The TAPAZ technology offers a search by event fragments or technological cycles, which are described by special TAPAZ-units, which are macroprocesses [1] in the assembly, when specialized subject domain processes are algorithmically correspond to TAPAZ macroprocesses and the roles of all participants in the events are algorithmically calculated [2].

This approach provides maximum accuracy and speed of search, relevance of search results and simultaneously solves the problem of automatically identifying the semantic equivalence of text documents and borrowing scientific ideas in order to curb the spread of plagiarism and prevent clogging the information space under the conditions of its globalization. In addition, it allows you to find similar technological cycles in close (adjacent) and distant subject domains, thereby providing support to the user in analytical activities, which greatly expands the functionality of the search engine, shifting it towards inventive level.

Judging by the rapid development since 2011 of activity-based technology for the international public resource Schema.org by Google, Microsoft, Yahoo and Yandex, as well as since 2017 – the Activity Vocabulary by the W3C Consortium for the Semantic Web, in the next 10–15 years, the main efforts of international scientific and financial centers will be focused on the

[1] We emphasize that macroprocess is one of 112 extremely abstract processes that are isomorphic to any subject domain and are calculated and encoded by the TAPAZ-algebra.

[2] "There are such concepts as "culprit", "tool", "product of labor" ⟨. . .⟩ We are here in the field of various categories, apparently ontological, but essentially semantic" [44: 11].

creation of knowledge graphs for automatic extraction of semantically relevant information from search pages, in other words, on the stage-by-stage development of a language capable for representing and transforming information in a readable machine form. Such a language should describe both the data that exists in any branch of knowledge (subject domain), and the rules for reasoning about this data, as well as the rules for displaying data on the Internet and back. The transition to the seventh technological order depends on this.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities," *Scientific American*, vol. 284, pp. 34–43, 2001.

[2] C. J. Fillmore, *"The Case for Case". Universals in Linguistic Theory*. New York : Holt, Rinehart and Winston, 1968.

[3] ——, "Frame semantics and the nature of language," in *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, vol. 280, 1976, pp. 20–32.

[4] ——, "Frame semantics," in *Linguistics in the Morning Calm*. Seoul: Hanshin Publishing Co., 1982, pp. 111–137.

[5] C. J. Fillmore and S. Atkins, "Starting where the dictionaries stop: The challenge for computational lexicography," in *Computational Approaches to the Lexicon*. Oxford: Oxford University Press, 1994, pp. 349–393.

[6] R. S. Jackendoff, *Semantic Interpretation in Generative Grammar*. Cambridge, MA: MIT Press, 1972.

[7] ——, *X-bar Syntax: A Study of Phrase Structure*. Cambridge, Mass.: MIT Press, 1977.

[8] T. A. Stowell, "Origins of phrase structure," in *Thesis (Ph.D.)*, Massachusetts Institute of Technology. Dept. of Linguistics and Philosophy., 1981, p. 496, Available: http://www.ai.mit.edu/projects/dm/theses/stowell81.pdf.

[9] R. Montague, in *Formal Philosophy: Selected Papers of Richard Montague*. New Haven, London: Yale University Press, 1974, pp. 222–247.

[10] G. Lakoff, "Instrumental Adverbs and the Concept of Deep Structure," *Foundations of Language*, vol. 4, no. 1, pp. 4–29, 1968.

[11] L. Tesnière, *Éléments De Syntaxe Structurale*. Paris: Librairie C. Klincksieck, 1959, english translation: L. Tesnière, Elements of structural syntax. Amsterdam / Philadelphia: John Benjamins Publishing Company, 2015, 698 p.

[12] V. Martynov, *Semiologicheskie osnovy informatiki [Semiological foundations of computer science]*. Minsk: Nauka i tekhnika [Science and technics], 1966.

[13] Ashtadhyayi or Sutrapath of Panini [Electronic resource]. Mode of access: https://sanskritdocuments.org/doc_z_misc_major_works/aShTAdhyAyI.pdf. — Date of access: 12.06.2021.

[14] V. N. Toporov, "On some analogies to the problems and methods of modern linguistics in the works of ancient indian grammarians," *Brief communications of the Institute of Asian Peoples*, pp. 123–133, 1961.

[15] O. A. Voloshina, "On the structure and linguistic terminology of panini grammar and its influence on european linguistics," *Bulletin of the Russian State University for the Humanities, Series: History. Philology. Culturology. Oriental studies*, vol. 9, no. 52, pp. 161–172, 2010.

[16] V. Martynov, *Semiologicheskie osnovy informatiki [Semiology foundations of informatics]*. Minsk: Nauka i tekhnika [Science and technics], 1974.

[17] ——, *Universal'nyj Semanticheskij Kod. Grammatika. Slovar', Teksty [Universal Semantic Code. Grammar. Dictionary, Texts]*. Minsk: Nauka i Tekhnika [Science and Technics], 1977.

[18] ——, *Universal'nyj Semanticheskij Kod. USK-3 [Universal Semantic Code: USC–3]*. Minsk: Nauka i Tekhnika [Science and Technics], 1984.

[19] ——, *Universal'nyj Semanticheskij Kod. USK-4 [Universal Semantic Code: USC–4]*. Institute of Linguistics of Academy of Science of BSSR, Minsk: Nauka i Tekhnika [Science and Technics], 1988, no. 2, Preprint.

[20] ——, *Universal'nyj Semanticheskij Kod. USK-5 [Universal Semantic Code: USC–5]*. Minsk State Linguistic University, Minsk: Izdatel'stvo Minskogo Gosudarstvennogo Lingvisticheskogo Universiteta [Minsk State Linguistic University Publ.], 1995, no. 4, Preprint.

[21] ——, *Osnovy Semanticheskogo Kodirovaniya. Opyt Predstavleniya i Preobrazovaniya Znanij [Foundations of Semantic Coding. Experience of Knowledge Representation and Conversion]*. Minsk: Izdatel'stvo Evropejskogo gumanitarnogo universiteta [European Humanitarian University Publ.], 2001.

[22] A. Hardzei, "Procedural Semantics and Calculus of Subject Domains," in *CONFERENCE 1994, Language: Semantics, Syntactics, Pragmatics*, M. Leschenko, Ed. Minsk: Minsk State Linguistic University Publ., 1995, pp. 16–17, pt. 1.

[23] ——, *The Deductive Theory of Language*. Minsk: Belarus Science, 1998.

[24] ——, *The Principles of Calculating the Semantics of Subject Domains*. Minsk: Belarusian State University Publ., 1998.

[25] Z. Harris, *String Analysis of Sentence Structure*. The Hague: Mouton, 1962.

[26] ——, “Co-occurrence and transformation in linguistic structure,” *Language*, vol. 33, no. 3, pp. 283–340, 1957.

[27] M. I. Svyatoshchik, “Translating Multi-Object Semantics into Single-Object Semantics as a Method of Minimizing Semantic Calculation,” in *Uchonyye zapiski VGU im. P. M. Masherova: sb. nauch. trudov [Scientific notes of Vitebsk State University named after P. M. Masherov: collection of scientific papers]*. Vitebsk: Vitebsk State University Publ., 2020, vol. 31, pp. 180–184.

[28] N. Chomsky, *“Lectures on government and binding: The Pisa lectures”. Studies in generative grammar*. New Jersey : Foris Publications, Dordrecht and Cinnaminson, 1981, no. 9.

[29] A. Hardzei, “Virtual string as a syntactic code of a sentence (on the example of the Chinese language),” in *CONFERENCE 2007, Language, society and problems of intercultural communication*, L. M. Sereda, S. V. Grinev-Grinevich, and I. F. Uhvanova-Shmygova, Eds. Grodno: Grodno State University Publ., 2007, pp. 349–358, pt. 2.

[30] ——, “Dynamic Syntax: A Semantic View,” *Foreign Languages in Tertiary Education*, vol. 4, no. 43, pp. 26–34, 2017.

[31] ——, “The Paradigm of Parts of Language,” in *CONFERENCE 2003, Word Formation and Nominative Derivation in Slavic Languages*, S. Emelyanova, L. Rychkova, and A. Nikitevich, Eds. Grodno: Grodno State University Publ., 2003, pp. 173–179.

[32] ——, “Parts of Language and the Procedures of Its Delineation,” in *The Paths of the Middle Kingdom*, A. Hardzei and W. Hongbin, Eds. Minsk: Belarusian State University Publ., 2006, ch. 1, pp. 69–75, iss. 1.

[33] Activity Vocabulary. W3C Recommendation [Electronic resource]. Mode of access: https://www.w3.org/TR/activitystreams-vocabulary/#dfn-activity. — Date of access: 12.06.2021.

[34] Semantic Web [Electronic resource]. Mode of access: https://www.w3.org/standards/semanticweb. — Date of access: 12.06.2021.

[35] Schema.org Vocabulary, version 8.0 [Electronic resource]. Mode of access: https://schema.org. — Date of access: 12.06.2021.

[36] B. Aarts, in *English Syntax and Argumentation*, 2nd ed. Houndmills, Basingstoke, Hampshire, New York: Palgrave (Macmillan), 2001, pp. 94–95.

[37] S. P. Rippa and O. M. Lyashenko, “Semantic platforms of knowledge bases software in informatics,” in *Pratsi Odes’koho politekhnichnoho universytetu [Proceedings of Odessa Polytechnic University]*, vol. 1, no. 40. Odessa: Odessa Polytechnic University Publ., 2013, pp. 91–96.

[38] T. Berners-Lee, N. Shedbolt, and W. Hall, “Semantic Web Revisited,” *IEEE Intelligent Systems*, vol. 21, pp. 96–101, 2006, iss. 3.

[39] A. Hardzei, *Theory for Automatic Generation of Knowledge Architecture: TAPAZ-2*, Rev. English ed. Minsk: The Republican Institute of Higher School Publ., 2017, Available: http://tapaz.by.

[40] ——, “Plagiarism Problem Solving Based on Combinatory Semantics,” *Communications in Computer and Information Science (CCIS)*, vol. 1282, pp. 176–197, 2020, Available: https://link.springer.com/book/10.1007%2F978-3-030-60447-9.

[41] ——, “Semantic Markup of the Event and its Display by Means of the Chinese and Russian Languages,” *Foreign Languages in Tertiary Education*, vol. 2, no. 57, pp. 5–26, 2021.

[42] G. E. Hinton. “Boltzmann machine”. Scholarpedia [Electronic resource]. Mode of access: http://www.scholarpedia.org/article/Boltzmann_machine. — Date of access: 12.06.2021.

[43] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[44] T. Kotarbinski, *Praxiology: An Introduction to the Sciences of Efficient Action*. Oxford: Pergamon Press, 1965.

Семантический подход к решению проблемы обработки данных на естественном языке

А. Н. Гордей

На основе третьей редакции второй версии Теории автоматического порождения архитектуры знаний (ТАПАЗ-2) предложен новый подход к семантической разметке события и формализации синтаксиса китайских, английских и русских предложений.

Ключевые слова: комбина́торная семантика, сематическая разметка, семантический падеж, семантический классификатор, граф знаний, ролевой лист индивидов, субъект, объект, акция, макропроцесс, специализированный процесс, модель мира, ТАПАЗ-алгебра, ТАПАЗ-юнит.

Received 15.06.2021

The system for automatic suggestions generation for the purpose of autocompletion of Russian-language user search queries

Julia Haliak
Belarusian State University
IHS Markit Ltd.
Minsk, Belarus
julia.haliak@ihsmarkit.com

***Abstract*—The paper presents a solution to the problem of predictive input of user queries for information systems working with full-text databases. In contrast to the traditional approach, it is based on the preliminary automatic construction of a set of suggestions that are being recognized in the search space itself. The main advantages of the obtained solution, implemented in well-known information system, are presented.**

***Keywords*—query type-ahead, predictive query input, search query, query autocompletion, linguistic processor**

I. Introduction

The majority of modern information systems with an interactive natural language user interface have the functionality of automatic completion of a user query. The most common solutions are based on the generated history of the previous search, use query logging and/or some information about the user [1]–[3]. The solution proposed in our case is focused on information systems working with full-text databases (FTDBs) and is based on the possibility of preliminary (before the system exploitation) formation of a "history"of not yet carried out, but intended search, in the form of a set of autosuggestions that are automatically recognized in advance in the full-text database, since the user queries are going to be addressed to FTDB.

II. The proposed solution

The general formulation of the problem of autocompletion of user queries in our case states as follows. Let the query entered by the user at a certain point in time be represented as the chain:

$$W_1\ W_2\ \dots\ W_n\ , \tag{1}$$

where each W_i, $i = 1, \dots, n - 1$, $n > 1$, is a natural language word, and W_n, $n \geq 1$, is a word or a word prefix. The task of completing a query of the form (1), which to a certain extent already reflects the user's information need, is to automatically generate a list of such, ideally, grammatically and semantically correct natural language word sequences that include members of the chain (1). That means we are dealing not only with sequential completion of the string of the entered query, considered as its prefix, but with a more complex procedure, which, in the general case, involves immersing the part of the query already typed by the user into the context, namely, its addition at the beginning, at the end, or even inside. Moreover, the proposed sequences may not contain all words from (1), but in any case, they should be ranked according to their relevance to the original chain. In addition, the autocompletion procedure can be applied to the entered query repeatedly (fragmentarily). Proceeding from the fact that the majority of users follow the already established practice of formulating search queries and that they are traditionally more or less focused on keyword search, the following steps for solving the problem of generating a set of suggestions to autocomplete user queries can be proposed in described case:

1) expert analysis based on open source available datasets of the most frequent search queries and classification of their types;
2) classification of syntactic structures of queries for each of their types;
3) automatic linguistic analysis of the FTDB in order to recognize in its text documents the syntactic structures obtained at the previous stage and the selection of all sorts of corresponding lexical content from the FTDB, which constitutes the set of autosuggestions P.

There is the classification of the main types of search queries (the autosuggestions) obtained on the basis of the expert analysis, and the classification of syntactic structures for each of their types given in [4]. The following types of autosuggestions are defined as the most relevant:

1) "simple noun phrase"(повреждения металла (metal damage); защитный слой (protective layer)),
2) "extended noun phrase with prepositional-nominal construction"(установка оборудования в цехах (equipment installation in workshops); обработка поверхностей в условиях высокой температуры (surface treatment in high temperature conditions)),
3) "extended noun phrase with participle

phrase"(покрытие, создающее защитный слой (coating creating protective layer); повреждения, вызванные коррозией (damage caused by corrosion)),

4) "verb phrase"(создавать защитный слой (create protective layer); предотвратить повреждения металла (prevent metal damage)),
5) "sentence subject and predicate"(покрытие предотвращает (coating prevents); коррозия вызывает (corrosion causes)),
6) "lexicon"(антикоррозийный (anticorrosive); соосно (coaxially)).

There are also requirements formulated for the linguistic processor that provides automatic recognition of the above-mentioned syntactic structures in texts from FTDB. Such recognition was made possible due to the use of the basic linguistic processor (BLP) [5] of the information system IHS Goldfire [6], which carries out automatic linguistic analysis of the input text by stage-by-stage processing:

- text formatting and normalization: here the text is converted into a certain unified format that preserves the stylistic and structural markup of documents as much as possible; in addition, the text is divided into paragraphs; headings, subheadings and sections are recognized;
- lexical text analysis: here the words and sentences boundaries are defined, the problems of recognition of proper names, abbreviations, e-mail addresses, digital and other sign complexes are solved partially or completely;
- lexical-grammatical text analysis: here the system identifies the lexical-grammatical category of each word, taking into account its morphology and context in accordance with a given classifier;
- syntactic and semantic text analysis: here the syntactic relations are recognized in each sentence and presented, as a rule, in the form of a functional or syntactic tree, in which the words of the sentence obtain the identification of their grammatical function and the type of syntactic connection between them is determined; on this stage of analysis the system also recognizes the relationships between concepts expressed by noun phrases, within the so-called SAO-structure [7]: Subject - Action (Predicate) - Object, and each element in this structure has its own attributes [8], [9].

Consider the following sentence as an example:

Антикоррозийное покрытие аэрозольного нанесения, создающее защитный слой, предотвращает повреждения металла, вызываемые коррозией. (Anti-corrosion spray coating that creates a protective layer prevents metal damage caused by corrosion.)

Having a linguistic analysis of this sentence performed, BLP recognizes the following three SAO-relations presented in Table 1, where HW stands for Headword (words that serve as a link to a parent relation – usually Predicate or Noun Phrase, but links to other fields are possible), C – Conjunction that relates to the whole relation, S –Syntactic Subject, P – Predicate (verb infinitive), NP – Nominal Predicate, O – Direct Object, D – Object in Dative, I – Object in Instrumental, Pr – Preposition that introduces syntactic Indirect Object, IO – Indirect Object, A – Adverbial modifier, In – Introduction phrase, AO – Action Original, original form of a predicate in text.

Таблица I
Table 1. SAO Fields

Fields	**SAO 1**	**SAO 2**	**SAO 3**
HW	-	повреждения металла металла	анти-коррозийное покрытие аэрозольного нанесения
C	-	-	-
S	анти-коррозийное покрытие аэрозольного нанесения	коррозия	анти-коррозийное покрытие аэрозольного нанесения
P	предотвращать	вызывать	создавать
NP	-	-	-
O	повреждения металла	повреждения металла	защитный слой
D	-	-	-
I	-	-	-
Pr	-	-	-
IO	-	-	-
Adv	-	-	-
In	-	-	-
AO	предотвращает	вызываемые	создающее

The SAO-structure is he source material on the basis of which, taking into account a certain expansion of the BLP functionality, algorithms for the automatic construction of autosuggestions of all the above-mentioned types are being built. These algorithms are based on the synthesis of the required autosuggestions from the content of the fields corresponding to each of their types. In particular, the source for generating autosuggestions of «simple noun phrase» type are the following fields: Headword, Subject, Nominal Predicate, Direct Object, Object in Dative, Object in Instrumental, Indirect Object; for the «extended noun phrase with prepositional-nominal construction» type – Headword, Preposition and Indirect Object fields; for the «extended noun phrase with participle phrase» type – Headword, Action Original, Direct Object, Object in Dative, Object in Instrumental, Preposition and Indirect Object fields; for the «verb phrase» type – Predicate, Action Original, Direct Object, Object in Dative, Object in Instrumental, Preposition and Indirect Object fields; for the «subject and predicate» type – Subject and Action Original fields; and all the SAO fields – for the «lexicon» type.

From the autosuggestions obtained in this way derivative suggestions are additionally synthesized by truncating noun phrases with coordinated attributes (антикоррозийное покрытие -> покрытие (anti-corrosion coating -> coating)), and considering only the main words of noun phrases (коррозия металла -> коррозия (metal corrosion -> corrosion)). In addition, these algorithms take into account the following important circumstance. One of the main criteria for the relevance of a suggestion for the automatic completion of a query is its informativeness, as a separate query or its part, in a given subject domain [10]. For example, noun phrases obtained from introduction constructions, such as «в том числе» (amongst other things), «в большинстве случаев» (in majority of cases) – «то число» (e.g. other things), «большинство случаев» (majority of cases), – firstly, cannot be a separate query, and secondly, due to their obviously poor informativeness, are also a bad part of a query. This fact is confirmed by the absence of such bigrams and trigrams in the lists of user requests available in open sources [11]–[14]. In addition, certain filtering is required, for example, for such noun phrases as « т.н. желтая ржавчина» (so-called yellow rust) because elements like certain abbreviations, which are more typical for formal style, are low-frequent in user queries [11]–[14]. The studies carried out in this aspect allowed, as a result, to develop a set of rules for identifying such cases in order to exclude certain SAO-relations or their fields from the list of candidates for generating autosuggesions, as well as the necessary further filtering of the candidates remaining in the list and their possible transformation.

Assuming the example sentence is included in the FTDB, the following list of autosuggestions will be obtained from it:

1) покрытие аэрозольного нанесения (aerosol application coating); аэрозольное нанесение (aerosol application);
2) нанесение (application);
3) повреждения металла (metal damages);
4) повреждения (damages);
5) металл (metal);
6) коррозия (corrosion);
7) защитный слой (protective layer);
8) слой (layer);
9) антикоррозийное покрытие аэрозольного нанесения, создающее защитный слой (anticorrosive spray coating creating a protective layer);
10) повреждение металла, вызываемые коррозией (metal damage causing by corrosion);
11) антикоррозийное покрытие аэрозольного нанесения, создающее (anticorrosive spray coating creating);
12) повреждения металла, вызываемые (metal damage causing); покрытие, создающее защитный слой (coating creating protective layer);
13) вызываемые повреждения металла (causing metal damage); покрытие, создающее (coating creating);
14) повреждения, вызываемые (damage causing);
15) предотвращать повреждения металла; вызывать повреждения металла (prevent metal damage);
16) создавать защитный слой (create protective layer);
17) предотвращает повреждения металла (prevents metal damage);
18) предотвращать повреждения (prevent damage);
19) предотвращает повреждения (prevents damage);
20) вызывать повреждения (cause damage);
21) создавать слой (create layer);
22) предотвращать повреждения металла, вызываемые коррозией (prevent metal damage causing by corrosion);
23) предотвращает повреждения металла, вызываемые коррозией (prevents metal damage causing by corrosion);
24) предотвращать повреждения, вызываемые коррозией (prevent damage causing by corrosion);
25) предотвращает повреждения, вызываемые коррозией (prevents damage causing by corrosion);
26) антикоррозийное покрытие аэрозольного нанесения предотвращает (anti-corrosion spray coating prevents);
27) предотвращает антикоррозийное покрытие аэрозольного нанесения (anti-corrosion spray coating prevents – inversed);
28) покрытие аэрозольного нанесения предотвращает (spray coating prevents);
29) покрытие предотвращает (coating prevents);
30) предотвращает покрытие аэрозольного нанесения (spray coating prevents – inversed);
31) предотвращает покрытие (coating prevents – inversed);
32) антикоррозийное (anti-corrosion);
33) аэрозольное (aerosol);
34) вызываемые (caused);
35) защитный (protective);
36) создающее (crating);
37) предотвращать (prevent);
38) вызывать (cause);
39) создавать (create).

III. Conclusions

The proposed method for solving the user search query autocompletion problem has a number of significant advantages.

- It is focused on the most general formulation of the problem; and moreover, immersing a query into context using a database of autosuggestions at the alphabet level allows to simultaneously and effectively solve the problem of user query autocorrection.
- It is universal in relation to the query language due to the universality of the SAO-relation itself; it also provides a solution to the problem focused on the ideology of semantic search.
- It gives a solution to the problem in the absence or inability to use the history of queries, relying on the texts to be indexed for the later search when forming a database of potential autosuggestions, which provides not only a higher probability of matching the autosuggestion to the user's expectations, but also a guaranteed relevant response of the search engine.
- It enables the user to more accurately formulate his informational need.

As for the problem of actually immersing the query prefix in the context provided by autosuggestions, it is being effectively solved using the well-known string algorithms [15]. The presented results were incorporated into the IHS Goldfire information system and showed their relevance and effectiveness.

References

[1] Question-answering system and method based on semantic labeling of text documents and user questions: US Patent 8,666,730. James Todhunter, Igor Sovpel, Dzianis Pastanohau. (2009/2010)

[2] Search entry system with query log autocomplete: US Patent 20080065617 A1. Eric Paul Burke, Duke Fan, Alan Wada, Jawahar Malhotra, Brian Coe. (2005/2008)

[3] Search query suggestions based on personal information US Patent 9317585 B2. Maureen Heymans, Ashutosh Shukla, Harish Rajamani, Matthew E. Kulick, Bryan C. Horling, Jennifer E. Fernquist, Weniger". (2013/2014)

[4] Haliak J.D. Printsipial'naya skhema resheniya zadachi avtodopolneniya pol'zovatel'skikh poiskovykh zaprosov na russkom yazyke i ee analiz [Conceptual scheme of the solution for the problem of user search query autocompletion in Russian language and its analysis]. Uchenye zapiski VGU imeni P. M. Masherova: sb. nauch. Trudov [Scientific notes of Vitebsk State University named after P.M. Masherov: collection of scientific papers]. Vitebsk, VSU P. M. Masherova, 2020, Vol. 31, pp. 142-147.

[5] System and method for automatic semantic labeling of natural language texts: Patent EP2406731A4. J. Todhunter, I. Sovpel, D. Pastanohau. (2010).

[6] IHS Goldfire. Available at: https://ihsmarkit.com/products/enterprise-knowledge.html (accessed 2021, May).

[7] Cheusov, A.V. Razrabotka algoritmov i tekhnologii postroeniya mnogoyazychnogo bazovogo lingvisticheskogo protsessora. Diss. k-ta tekhn. nauk [Development of algorithms and technologies for building a multilingual basic linguistic processor. PhD thesis]. Minsk, 2013. 116 p.

[8] Haliak, J.D., Sovpel, I.V. Avtomaticheskoe dopolnenie russkoyazychnykh pol'zovatel'skikh zaprosov, formuliruemykh v vide imennykh grupp [Autocompletion of russian language user queries formulated as noun phrases]. Vestnik MGLU. Filologiya [MSLU Bulletin. Philology], Vol 1, 2021, No 3 (112), pp. 101-110.

[9] Haliak, J.D., Sovpel, I.V. Avtomaticheskoe dopolnenie russkoyazychnykh pol'zovatel'skikh zaprosov na osnove podskazok tipa «glagol'naya gruppa», «grammaticheskaya osnova predlozheniya» i «leksikon» [Autocompletion of Russian language user queries based on suggestions of «verb phrase», «sentence subject and predicate» and «lexicon» types]. Vestnik MGLU. Filologiya [MSLU Bulletin. Philology], Vol 1, 2021, No 4 (113) (in press)

[10] Voronkov, N.V. Metody, algoritmy i modeli sistem avtomaticheskogo referirovaniya tekstovykh dokumentov. Diss. k-ta tekhn. nauk [Methods, algorithms and models of systems for automatic summarization of text documents. PhD thesis]. Minsk, 2007, 165 p.

[11] Kpapamih: search-queries. Available at: https://www.kaggle.com/kpapamih/search-queries (accessed 2021, May).

[12] Wordstat.Yandex. Available at: https://wordstat.yandex.ru (accessed 2021, May).

[13] Reidsma, M. (2016). Summon Topic Explorer Results by Search Query [Data set]. Zenodo. http://doi.org/10.5281/zenodo.47723 [Electronic resource]. Available at: https://zenodo.org/record/47723 (accessed 2020, Apr).

[14] Rishiraj. (2018). Supplementary Material for Nested Segmentation of Web Search Queries [Data set]. Zenodo. http://doi.org/10.5281/zenodo.1137746 [Electronic resource]. Available at: https://zenodo.org/record/1137746 (accessed 2020, Apr).

[15] Irzhavskii, P. A. eds. Teoriya algoritmov: ucheb. posobie [Algorithm theory: schoolbook]. Minsk, BSU, 2013, 159 p.

Система автоматического построения подсказок с целью автодополнения русскоязычных пользовательских запросов

Ю. Д. Голяк

В работе представлено решение задачи предиктивного ввода пользовательских запросов для информационных систем, работающих с полнотекстовыми базами данных. Оно, в отличие от традиционного подхода, основано на предварительном автоматическом построении множества подсказок, распознаваемых в самом поисковом пространстве. Приводятся основные достоинства полученного решения, внедренного в состав известной информационной системы.

Received 21.07.2021

Linking Russian Words to Semantic Frames of FrameNet*

Dana Zlochevskaya
Computational Mathematics Faculty
Lomonosov Moscow State University
Moscow, Russia
dana_zl@mail.ru

Natalia Loukachevitch
Research Computing Center
Lomonosov Moscow State University
Moscow, Russia
louk_nat@mail.ru

Olga Nevzorova
ICMIT Institute
Kazan Federal University
Kazan, Russia
onevzoro@gmail.com

***Abstract*—This work is devoted to the transfer of the FrameNet semantic frames created for English into Russian. The transfer is based on semantic frames identified in English sentences and uses a parallel Russian-English corpus. The resulting set of semantic frames in Russian is utilized to train and evaluate the models for identifying semantic frames in Russian. The results of the work include: 1) a lexicon of Russian words linked to the FrameNet frames, 2) an annotated Russian dataset - sentences with labeled semantic frames, 3) the implementation, testing and analysis of models for identifying semantic frames in Russian.LaTeX.**

***Keywords*—semantic analysis, frames, language transfer, parallel corpus, neural networks**

I. Introduction

Semantic text analysis remains one of the most difficult tasks of natural language processing. Such an analysis requires a theory and a model of semantics representation. One of the most known semantic models is Charles Fillmore's Frame Semantics [1], which was explicated in the FrameNet lexicon of frames [2].

The idea of the Frame Semantics approach is as follows [1]: the senses of words can be represented using situations, their relations and roles. For example, the situation of “cooking” in most cases can be described with the help of the following participants and relations between them as: “the subject who cooks” (the cook), “the object that is cooked” (food), “the source of heat for cooking” (heat source) and "cooking container" (utensils). In this case, it is possible to say that the cooking frame has been introduced, in which “cook”, “food”, “heat source” and “containers” are elements of the frame. Frame elements, which are "markers" of the frame, that is, they often signal the location of this frame in the text (for example, "fry", "Cook", "stew", etc.) are called lexical units. Frames can be of different complexity, can have different numbers of elements and lexical units. The main task in constructing semantic frames is to show how the elements of the frame are related to each other.

The project is supported by the Russian Science Foundation, grant # 19-71-10056.

FrameNet is a basis of numerous works on the first step of semantic analysis called semantic role labeling [3]–[6]. However, such data are absent for most languages, and development of FrameNet-like resources from scratch is a very laborous process, requiring a lot of resources and time [7], [8]. Therefore various approaches for automatizing of framenet creation for other languages are discussed [9]–[11].

In this paper we consider the transfer of the FrameNet semantic frames into the Russian language. The method is based on the transfer of semantic frames identified in English sentences using a parallel Russian-English corpus. The resulting set of semantic frames in Russian is used to train and evaluate the models for identifying semantic frames in Russian. The results of this work include: 1) a lexicon of Russian words linked to the FrameNet frames, 2) an annotated Russian dataset - sentences with labeled semantic frames, 3) the implementation, testing and analysis of models for the identification of semantic frames in Russian.

II. Related Work

There is an interest of researchers in many countries to have a semantic resource similar to FrameNet for their own languages. However, to create a FrameNet-like resource from scratch is a very difficult, expensive, and time-consuming procedure [7], [8]. Therefore automated methods for generating a framenet for a specific language are used. Such methods can be subdivided into two groups: cross-lingual transfer, and generating frames in unsupervised manner from a large text collection [9], [12], [13]. Cross-lingual transfer can be made via linking FrameNet with WordNet [14], [15] or via parallel corpora [16].

Cross-lingual transfer of frames based on a parallel corpus requires a preliminary extraction of frames in English texts, which usually includes the following main steps [17], [18]:

- recognition of lexical units, which can express frames, in a text. This stage can be reduced to the task of binary classification for each word of the text;

- recognition of semantic frames. For each lexical unit, it is necessary to determine what frames are expressed by this unit. This task can be considered as a problem of multiclass classification (with several labels) for each of the selected lexical units at the previous stage. The classes in this case are the semantic frames themselves, and several frames can correspond to one lexical unit. The main difficulties at this stage are the possible ambiguity of a lexical unit and/or its absence in the training set.
- recognition of the remaining elements of the frame such as roles. Most often, this problem is solved using named entity extraction methods with the condition that the set of roles strongly depends on the frame being processed. In the current work, this stage is not studied.

The solutions for these three tasks can be very different. In the work of the winners of the SemEval 2007 competition [18], to solve the problem of identifying lexical units, morphological and syntactic rules were used. To define semantic frames, classifiers were trained for each frame separately using support vector machine (SVM) method, the features for which were both morphological and syntactic characteristics of a lexical unit, and semantic information about it from WordNet [19].

The widely used SEMAFOR algorithm [20] has become an improvement of this algorithm. To recognize lexical units, an improved version of linguistic rules is used. To identify a frame, a probabilistic model is trained on a similar feature set. In [21], an approach based on recurrent neural networks is proposed for both tasks: recognizing lexical units and recognizing semantic frames. In recent years, works have also appeared that solve this problem using transformers, for example, BERT [22], [23].

III. FrameNet

FrameNet [2] consists of several components:

- The base of semantic frames, containing a description of each frame: its structure, roles and relations between frames;
- Examples of sentences annotated with semantic, morphological and syntactic information. These sentences are examples of the use of semantic frames in natural language;
- Lexical units linked to frames as well as links to examples of annotated sentences.

At the moment, the FrameNet database contains more than 13000 lexical units, of which 7000 are fully annotated with more than 1000 hierarchically related semantic frames. The number of sentences annotated with semantic information is about 200,000. The base exists in several versions and is in the public domain for research purposes. Basic concepts from the terminology of the FrameNet project are as follows:

- Semantic frame is a semantic representation of an event, situation or relationship, consisting of several elements, each of which has its own semantic role in this frame.
- Frame element is a type of participants (roles) in a given frame with certain types of semantic links.
- Lexical unit - a word with a fixed meaning that expressed a given frame or a given frame element.

For example, In the sentence “Hoover Dam played a major role in preventing Las Vegas from drying up." The word "played" is a lexical unit that conveys the presence in the text of the semantic frame "PERFORMERS_AND_ROLES" with the frame elements "PERFORMER", "ROLE" and "PERFORMANCE" (who created, that created, what role the creator assumed in the creation). This example also shows that in one sentence there can be several frames with varying degrees of abstractness.

It is important to note that for the Russian language there is a FrameNet-oriented resource FrameBank [24]. This is a publicly available dataset that combines a lexicon of lexical constructions of the Russian language and a marked-up corpus of their implementations in the texts of the national corpus of the Russian language. The main part of FrameBank consists of 2200 frequent Russian verbs, for which the semantic constructions in which they are used are described, and examples of their implementations in the text are collected. Each construction is presented as a template, in which the morphological characteristics of the participants in their role and semantic restrictions are fixed.

Despite the fact that the semantic constructs of FrameBank are similar to frames from FrameNet, they are methodologically different. FrameNet frames are built around generic events with specific participants and relations between them. FrameBank constructs are built around the senses of specific words. It is supposed that the senses of each lexeme (mainly verbs) in FrameBank form a separate frame. Due to this methodological difference and the orientation of the FrameBank resource towards verbs as a center of constructions, the full use of this data set to solve the problem is impossible.

IV. Methods of Linking Russian Words to FrameNet

The method of linking Russian words to FrameNet is divided into several parts, each of which solves a specific subtask:

1) Training the model of recognizing lexical units and frames in English. At this stage, based on the FrameNet knowledge base, a model is created that can extract lexical units and semantic frames from any sentence in English. The quality of the models is evaluated on the test part of the FrameNet dataset and compared with the existing results.

2) Extracting lexical units and frames from the English part of the parallel corpus using a trained model.
3) Transferring the obtained semantic information into Russian using word matching through a pre-trained embedding model of the Russian language. At this stage, for each lexical unit from an English sentence, its analogue is searched for in a parallel Russian sentence. Transfer quality is evaluated on a test sample, annotated manually.
4) Extracting lexical units for each semantic frame. A set of annotated sentences in Russian is also formed for further recognition of semantic frames in Russian texts.

A. *Training models for identification of lexical items and frames in English*

As in most studies, two sequential models are trained. The first model predicts potential lexical units that can express a semantic frame, and the second one, based on the predictions of the first model, determines which frames should be generated from the selected lexical units. Both models are trained and tested on the manually annotated FrameNet corpus of sentences.

For training and testing, the annotated corpus FrameNet version 1.5 was used. The entire corpus of annotated sentences contains 158399 example sentences with labeled lexical units and frames. They are presented in 77 documents, 55 texts of which were randomly selected for model training and 23 texts for testing. A modified CONLL09 format is used as a universal format for presenting annotations, in which the following set of tags is attached to each word of the sentence:

- ID - word number in the sentence,
- FORM - the form of a word in a sentence,
- LEMMA - word lemma,
- POS - part of speech for the given word,
- FEAT - list of morphological features,
- SENTID - sentence number,
- LU - lexical unit, if the word is it,
- FRAME - a semantic frame generated by a word if it is a lexical unit.

The task of predicting lexical units in a text is reduced to the task of binary classification for each word in a sentence. In some cases, a lexical unit may not be one word, but a phrase, but in this work, the most common variant with one word as a lexical unit is investigated. To solve this problem, a bidirectional recurrent neural network (BiLSTM) was used. Such models were actively used in previous studies for identification of lexical units and frames [21].

A sentence is sent to the input of the neural network, each word in which is represented by a vector representation of a word from a pretrained embedding model. In addition to this classical representation of words, new features responsible for semantic and morphological information were studied such as: part of speech and the initial form of a word. These features are represented as a one-hot vectors, which is fed to the input of fully connected layers of the neural network. During training, the outputs of these fully connected layers can be considered as vector representations of these features. These features can help the model to memorize morphological and semantic schemes for constructing frames from the FrameNet annotation. All obtained feature vectors for a word are joined by concatenation into the final vector representation, which is fed to the input of the neural network. In the learning process, only vectors of tokens from the pretrained embedding model are fixed, the rest of the vector representations are formed during training.

A sentence is sent to the input of the neural network, each word of which is represented as a vector according to the algorithm described above. The network consists of several BiLSTM layers, to the output of which a fully connected layer with the sigmoid activation function is applied for each word. As a result, at the output of the network, each word of the sentence is matched with the probability of a given word to be a lexical unit in a given sentence.

To optimize the parameters of the neural network, the logistic loss function was used. The adaptive stochastic gradient descent Adam [25] was chosen as an optimizer. To prevent overfitting of the neural network, the Dropout technique [26] was used, based on random switching off of neurons from the layers of the neural network for greater generalization of the trained models.

B. *Semantic frame identification*

The purpose of the semantic frame identification model is to recognize frames that correspond to lexical items in a sentence. Formally, this is a multi-class classification problem with multiple labels, since the same lexical unit can correspond to several semantic frames in a sentence.

To solve this problem, a model was used that is similar to the model of the selection of lexical units, but with several changes. The main difference is adding information about whether a word is a lexical unit using one-hot coding. If the word is not a lexical unit in a given sentence, the lexical unit representation vector will consist entirely of zeros and will not affect the prediction. It is important to note that the predictions of the model are taken into account only for words that are lexical units in a given sentence. The remaining components of the vector representation of a word are similar to the model for extracting lexical units - a vector of tokens from a pretrained embedding model and one-hot coding that encodes a part of speech.

C. *Implementation and results*

The following hyperparameters were chosen for training the models:

Model for lexical units	P	R	F1
$SEMAFOR$	74.92	66.79	70.62
$BILSTM$	74.13	66.11	69.89
$BILSTM_{token}$	76.01	67.15	71.3
$BILSTM_{token,lemma}$	76.12	67.98	71.8
$BILSTM_{token,lemma,partofspeech}$	79.47	68.31	73.46

Table I
Results for lexical units recognition in English

Model	P	R	F1
SVM	79.54	73.43	76.36
$SEMAFOR$	86.29	84.67	85.47
$BILSTM$	83.78	79.39	81.52
$BILSTM_{lu}$	88.19	81.54	84.73
$BILSTM_{lu,token}$	88.83	88.12	85.34
$BILSTM_{lu,token,partofspeech}$	89.87	83.91	86.78

Table II
Results of model of semantic frame recognition for English

- The Glove model trained on the English-language Wikipedia[1] was chosen as an embedding model. The vector dimension is 100;
- The size of the vectors encoding the lemma, lexical unit and part of speech is 100, 100, and 20, respectively;
- Number of BiLSTM layers is 3, output dimension is 100;
- The number of training epochs is 40;
- The Dropout coefficient is 0.01.

The results of lexical unit identification obtained on the test sample are presented in Table 1. For comparison, the SEMAFOR algorithm was applied, which determines the lexical units on the basis of linguistic rules. For this, an available author's implementation [2] was used, trained on the same data as the tested models. It can be seen from the results that adding information about the word lemma does not give a significant improvement, however, information about the part of speech allows obtaining quality that is superior to the classical SEMAFOR model.

To evaluate the results of the semantic frame model, the SEMAFOR algorithm was also used, which identifies a frame based on the probabilistic model. In addition, for comparison, the model of the SemEval 2007 [16] was recreated. The obtained results are presented in Table 2. It can be seen that the trained model has a quality comparable to the performance of the SEMAFOR model.

In this way, models for identification of lexical units and semantic frames in English were trained. At this stage, for any sentence in English, a set of frames and corresponding lexical units are identified. To transfer this information to the Russian part of the parallel corpus, it is necessary for each lexical unit from the English sentence to find its analogue in Russian. Since in this study only single-word lexical units are considered, the task is reduced to the comparison of words between sentences in a parallel corpus.

[1]https://nlp.stanford.edu/projects/glove/
[2]https://github.com/Noahs-ARK/semafor

D. Translation of annotations from English into Russian in a parallel corpus

For the study, we used the English-Russian parallel corpus [3], gathered by Yandex. It consists of 1 million pairs of sentences in Russian and English, aligned by lines. The sentences were selected at random from parallel text data collected in 2011-2013.

Like most parallel corpora, this resource is sentence-aligned, not word-aligned, so word-level matching requires further refinement. A simple dictionary translation of an English word into Russian does not provide the desired effect due to the ambiguity of words and differences in structure of languages. Therefore, in addition to direct translation of words, a matching algorithm was implemented based on the similariy of words in an embedding model of the Russian language.

The FastText model in the Skipgram version [27] trained on the Russian National Corpus [4] was chosen for word matching. Thus, the algorithm of matching between languages consists of the following steps:

1) Translation of a lexical unit in English, for which we are looking for an analogue in a parallel sentence in Russian. In this step, all possible translations into Russian are collected using the Google Translate API [5].
2) Further, between each obtained translation and each word of a parallel Russian sentence, the cosine similarity according to the embedding model is calculated.
3) A word in the Russian sentence is considered as an analogue of the original word in English if between the translation of an initial word and the Russian word, the cosine similarity is higher than 0.9.

To evaluate the quality of word matching, the transfer of lexical units in 100 parallel sentences was manually assessed. Both precision (in how many sentences the word was translated correctly) and recall (in how many sentences an analogue of the word in English was found in general) were considered. A simple search for a translation of a word in a parallel sentence was taken as the basic algorithm; the comparison results can be seen in Table 3. It can be seen that the use of the embedding model increases the recall of word matching with a slight decrease in precision.

Word translation search method	P	R
Direct translation matching	91 %	72 %
Distributive word matching	90 %	87 %

Table III
Results of matching words between sentences in a parallel corpus

Thus, applying this algorithm to each pair of sentences in a parallel corpus, it is possible to transfer the selected

[3]https://translate.yandex.ru/corpus
[4]https://rusvectores.org/ru/models/
[5]https://cloud.google.com/translate/docs/

lexical units and the corresponding semantic frames from English into Russian.

E. Characteristics and evaluation of the resulting corpus

The sentences in Russian obtained after the transfer with annotated lexical units and semantic frames form a dataset similar to a of the FrameNet knowledge base.

In it, for each of the 755 frames, lexical units with frequency of use in the context of the frame are presented. This frequency can be interpreted as a certain "reliability" of the lexical unit belonging to the frame. A total of 2.8 million lexical units were analysed out of 1 million sentences. They belong to 755 semantic frames from the FrameNet project. In total, 18150 lexical units have been identified, including 6894 unique words.

Table 5 shows an example of the obtained semantic frames and lexical units assigned to it. Each column refers to a frame, the first line contains its name from the original FrameNet knowledge base, and the second contains lexical units assigned to it in Russian with the frequency of use.

Fear	Labeling	Reason
страх : 1042	термин : 1045	причина : 3287
бояться : 552	понятие : 139	основа : 755
боязнь : 42	этикетка : 119	основание : 533
опасаться : 40	ярлык : 78	мотивация : 247
ужас : 28	терминология : 24	повод : 118
страшиться : 3	марка : 11	мотив : 110
страшно : 3	бренд : 9	поэтому : 2
испуг : 1	клеймо : 2	именно : 1

Table IV
Examples of linking Russian lexical units to the FrameNet frames

The resulting dataset can be used as a separate semantic resource for various natural language processing tasks. In addition, annotated Russian sentences are also valuable. They make it possible to conduct experiments on the selection of lexical units and semantic frames using supervised machine learning methods.

V. Training and testing model for identification of lexical units and frames in Russian

The resulting set of annotated sentences in Russian was used to train models for the selection of lexical units and semantic frames in Russian, similar to the already trained models in English. Out of 970 thousand sentences, 90% were used for training models, the rest were used for testing.

The architecture and method of constructing the vector representation of words are similar to the models for the English language, with the exception of the embedding model - instead of Glove, the FastText model of the Skipgram architecture was used, trained on the National corpus of the Russian language [6]. The vector dimension is 300. The results of the obtained models are presented in Tables 5 and 6.

[6]https://rusvectores.org/ru/models/

Lexical Identification Model	P	R	F1
$BILSTM$	71.67	59.14	64.80
$BILSTM_{token}$	76.09	61.04	67.73
$BILSTM_{token,lemma}$	77.65	61.33	68.53
$BILSTM_{token,lemma,partofspeech}$	78.44	61.72	69.08

Table V
Results of the Russian lexical unit identification model

Semantic Frame Identification Model	P	R	F1
SVM	80.28	72.43	76.15
$BILSTM$	84.10	76.99	80.38
$BILSTM_{lu}$	86.54	78.04	82.07
$BILSTM_{lu,token}$	87.01	78.20	82.40
$BILSTM_{lu,token,partofspeech}$	90.83	83.91	84.66

Table VI
Results of the model for identifying semantic frames for the Russian language

The obtained results for Russian are lower than the results of similar models in English. This can be explained by the fact that during the automatic transfer, there is a loss of data and the introduction of noise at each stage - both when using the models for extracting semantic information in English, and when directly transferring the resulting annotation. In addition, some frames and lexical units can be rarely represented in a parallel corpus, which leads to a low quality of their prediction.

Thus, the results show that the obtained dataset for the Russian language can be used to develop methods for extracting semantic frames and use it for other natural language processing tasks.

VI. Conclusion

In this work, methods of automatic identification of lexical units and semantic frames in Russian have been investigated. The following results were obtained:

- The existing approaches to identification of semantic frames with the use of expert FrameNet annotations have been investigated. Methods for transferring semantic information between languages have been studied, in particular, methods using parallel text corpora.
- Models of lexical units and semantic frames identification have been trained and tested for English. A neural network based on BiLSTM layers, trained on annotated sentences of the FrameNet 1.5 project, was used. Additional morphological information were also used as an input of a neural network. As a result of the experiments, it was possible to achieve the quality F1 73.46 for the selection of lexical units and F1-micro 86.78 for the model identifying semantic frames.
- An algorithm for transferring annotations from English to Russian in a parallel corpus was implemented. The use of the pretrained embedding model allowed increasing the recall of the transfer by 15%, leaving the precision at the same level.

- A set of semantic frames and lexical units linked to them in Russian has been created. A corpus of 970 thousand annotated sentences in Russian with identified lexical units and semantic frames was received. This resource can be used both for further research in the field of automatic extraction of semantic frames, and in other tasks of natural language processing.
- On the obtained set of annotated sentences, we trained models for identification of lexical units and frames in Russian. For the model of identifying lexical units, quality F1 69.08 was obtained, for the model for determining semantic frames F1-micro 84.88.

References

[1] C. J. Fillmore *et al.*, "Frame semantics," *Cognitive linguistics: Basic readings*, vol. 34, pp. 373–400, 2006.

[2] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, 1998, pp. 86–90.

[3] M. Palmer, D. Gildea, and N. Xue, "Semantic role labeling," *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–103, 2010.

[4] X. Carreras and L. Màrquez, "Introduction to the conll-2005 shared task: Semantic role labeling," in *Proceedings of the ninth conference on computational natural language learning (CoNLL-2005)*, 2005, pp. 152–164.

[5] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, "Deep semantic role labeling: What works and what's next," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 473–483.

[6] J. Cai, S. He, Z. Li, and H. Zhao, "A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware?" in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2753–2765.

[7] J. Park, S. Nam, Y. Kim, Y. Hahm, D. Hwang, and K.-S. Choi, "Frame-semantic web: a case study for korean." in *International Semantic Web Conference (Posters & Demos)*, 2014, pp. 257–260.

[8] K. H. Ohara, S. Fujii, H. Saito, S. Ishizaki, T. Ohori, and R. Suzuki, "The japanese framenet project: A preliminary report," in *Proceedings of pacific association for computational linguistics*. Citeseer, 2003, pp. 249–254.

[9] M. Pennacchiotti, D. De Cao, R. Basili, D. Croce, and M. Roth, "Automatic induction of framenet lexical units," in *Proceedings of the 2008 conference on empirical methods in natural language processing*, 2008, pp. 457–465.

[10] J. C. K. Cheung, H. Poon, and L. Vanderwende, "Probabilistic frame induction," *arXiv preprint arXiv:1302.4813*, 2013.

[11] S. Tonelli, D. Pighin, C. Giuliano, and E. Pianta, "Semi-automatic development of framenet for italian," in *Proceedings of the FrameNet Workshop and Masterclass, Milano, Italy*, 2009.

[12] D. Ustalov, A. Panchenko, A. Kutuzov, C. Biemann, and S. P. Ponzetto, "Unsupervised semantic frame induction using triclustering," *arXiv preprint arXiv:1805.04715*, 2018.

[13] A. Modi, I. Titov, and A. Klementiev, "Unsupervised induction of frame-semantic representations," in *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, 2012, pp. 1–7.

[14] M. L. De Lacalle, E. Laparra, and G. Rigau, "Predicate matrix: extending semlink through wordnet mappings." in *LREC*, 2014, pp. 903–909.

[15] M. L. De Lacalle, E. Laparra, I. Aldabe, and G. Rigau, "Predicate matrix: automatically extending the semantic interoperability between predicate resources," *Language Resources and Evaluation*, vol. 50, no. 2, pp. 263–289, 2016.

[16] T.-H. Yang, H.-H. Huang, A.-Z. Yen, and H.-H. Chen, "Transfer of frames from english framenet to construct chinese framenet: a bilingual corpus-based approach," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[17] C. F. Baker, M. Ellsworth, and K. Erk, "Semeval-2007 task 19: Frame semantic structure extraction," in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 2007, pp. 99–104.

[18] R. Johansson and P. Nugues, "Lth: semantic structure extraction using nonprojective dependency trees," in *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, 2007, pp. 227–230.

[19] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[20] D. Das, D. Chen, A. F. Martins, N. Schneider, and N. A. Smith, "Frame-semantic parsing," *Computational linguistics*, vol. 40, no. 1, pp. 9–56, 2014.

[21] S. Swayamdipta, S. Thomson, C. Dyer, and N. A. Smith, "Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold," *arXiv preprint arXiv:1706.09528*, 2017.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[23] W. Quan, J. Zhang, and X. T. Hu, "End-to-end joint opinion role labeling with bert," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2438–2446.

[24] O. Lyashevskaya and E. Kashkin, "Framebank: a database of russian lexical constructions," in *International Conference on Analysis of Images, Social Networks and Texts*. Springer, 2015, pp. 350–360.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

Связывание русскоязычной лексики с семантическими фреймами лексикона FrameNet

Злочевская Д.И., Лукашевич Н.В., Невзорова О.А.

Данная статья посвящена исследованию методов выделения семантических фреймов из текстов на русском языке. Рассматривается теория семантических фреймов и ее практическое использования при решении задач обработки естественного языка. Также производится ряд экспериментов по переносу крупнейшего корпуса семантических фреймов проекта FrameNet на русский язык с оценкой качества каждого из подходов. Основой метода является перенос результатов выделения сематических фреймов на английском языке с помощью параллельного русско-английского корпуса. Полученный набор семантических фреймов используется для обучения и оценки моделей выделения семантических фреймов на русском языке.

Результатом данной работы является размеченный набор данных – предложения с выделенными семантическими фреймами, а также реализация, тестирование и анализ характеристик моделей по выделению семантических фреймов на русском языке.

Received 30.05.2021

Deep Learning Approach in the Context of Information Retrieval for Solving both Automatic Natural Language Generation and Automatic Text Generation Problems

Yury Krapivin
Brest State Technical University
Brest, Belarus, 224000
Email: ybox@list.ru

***Abstract*—The article presents the solution of practical application of the deep learning approach in the context of information retrieval based on the usage of LSTM and GPT-2 language models for solving both automatic natural language generation and automatic text generation problems.**

***Keywords*—natural language, information retrieval, automatic natural language synthesis, automatic text synthesis, machine learning**

I. Introduction

The rapid development of information technology with each passing year increases the importance of obtaining prompt and high-quality information both for solving simple daily tasks (for example, related to the choice of a product or service, receiving region and world news) and for business analysis and making management decisions. The use for this purpose a global data network - the Internet, as well as a variety of related services and applying artificial intelligence technology applications has become an integral part of everyday life. Services and applications are turning from consumers to data generators insensibly and increasingly.

II. Natural language generation and text generation technologies

“Imperva Bad Bot Report 2021” – a research of “Imperva”, cybersecurity leader, whose mission is to protect data and all paths to it, suggests the growing scale and widespread impact of bots in daily life. It divide such applications in two groups:

1) Bad bots – interact with applications in the same way a legitimate user would, making them harder to detect and prevent. They enable high-speed abuse, misuse, and attacks on websites, mobile apps, and APIs. They allow bot operators, attackers, unsavory competitors, and fraudsters to perform a wide array of malicious activities. Such activities include web scraping, competitive data mining, personal and financial data harvesting, brute-force login, digital ad fraud, spam, transaction fraud, and more. [1]
2) Good bots – support the various business and operational goals of their owners—from personal users to large multinationals and can be categorized by the following four groups (“Fig. 1”):
   - Feed fetcher – Bots that ferry website content to mobile and web applications, which they then display to users.
   - Search engine bots – Bots that collect information for search engine algorithms, which is then used to make ranking decisions.
   - Commercial crawlers – Spiders used for authorized data extractions, usually on behalf of digital marketing tools.
   - Monitoring bots – Bots that monitor website availability and the proper functioning of various online features.

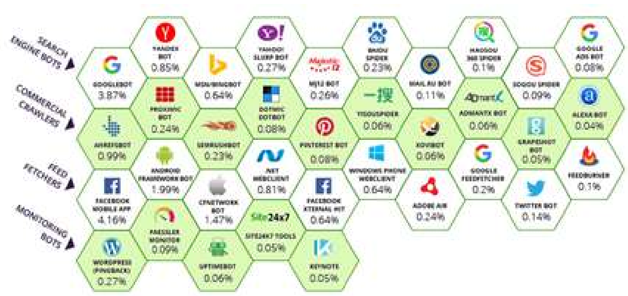


Figure 1. The four groups of good bots.

Imperva Research Labs saw the highest percentage of bad bot traffic (25.6%) since the inception of the report in 2014, while traffic from humans fell by 5.7%. More than 40% of all web traffic requests originated from a bot last year.

It is worth to note that activity of the "smart" applications also reveals in the creation of natural language data: vivid examples are dialogue systems and services that provide interaction with the user, as well as systems for automatic content creation.

Robot journalism is an actively developing area of research refers to the generation of news stories by algorithms based on data without human-journalistic intervention and these news stories are then published automatically on news websites. It relies on natural language generation (NLG) technology, which is a sub-field of artificial intelligence. The main objective of NLG technology is to design text generation systems that create readable explanatory stories based on data. Defining the grammatical and syntax rules of a language within an NLG system allows the automatic creation of various documents, reports, explanations, and summaries by the algorithms on the basis of the input data [2]. Currently, NLG systems help news media organizations to generate news stories and, in this context, the emergence of robot journalism is a result of the convergence of NLG technology and the news media sector. Narrative Science, Automated Insights, Yseop, and Arria are some of the large technology firms that are progressing NLG technology [3] and providing robot journalism services to news media organizations such as Associated Press, Forbes, and Yahoo. [4]

Another natural language processing technology related to tasks where the target is a single or several sentences or even the set of words – text generation. It also finds its applications in developing of question answering, dialog, machine translation and information retrieval systems. In the context of the last one, the results of solving these problems used for search-engine optimization to increase the rate of a web-resource in the search-engine output, as well as speedup and simplify the process of web-resource developing.

In today's digitalized world, search engines have become one of the most powerful tools on the Internet and an essential part of our daily live. By consolidating and organizing the wealth of information available online, search engines like Google [5], Yahoo [6] or Bing [7] help billions of online users find the content they need at a rapid pace. In 2019, almost 30 percent of global web traffic was generated via online search usage, showing the vital role these platforms play in directing and navigating user flows to different websites. For example, according to [8] the global marketing share percentage, in terms of the use of search engines heavily favours Google, with over 92% ("Fig. 2").

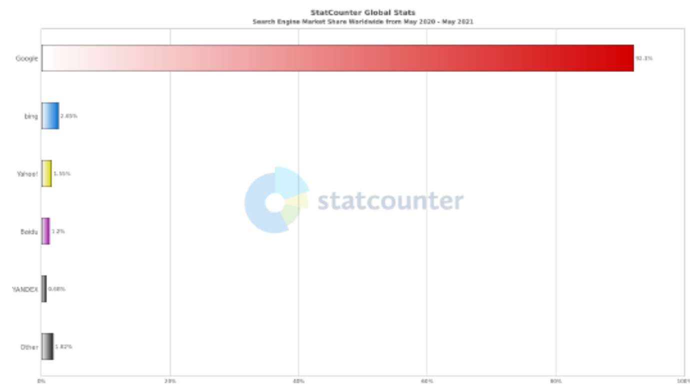


Figure 2. The global marketing share of search engines.

It is interesting to note that Google's large market share is still on the increase. In April 2017 the market share for Google was 77.43%.[9]

The analysis of search queries has revealed a stable downward trend of the number of keywords usage. One of the examples is the average number of typed search terms during online search in the United States as of January 2016. During that month, 20.14 percent of all U.S. online search queries contained tree keywords ("Fig. 3") and the situation changes heavily. As of January 2020, 40 percent of all U.S. online search queries contained two keywords. Three word search terms accounted for 22.74 percent of searches. Queries up to three words accounted for over 80 percent of online searches in the United States. [10] As search engines rank search results in order of relevancy, meaning that the most valuable links for users' queries are displayed prominently on the results page, high rankings have become one of the top digital priorities for companies worldwide. [11] That is why tools like UberSuggest's [12] or SemRush [13] are quite popular among web-masters and usually used for analysis of the web traffic, user queries statistics etc. Some of those routine but important functions: candidate words and

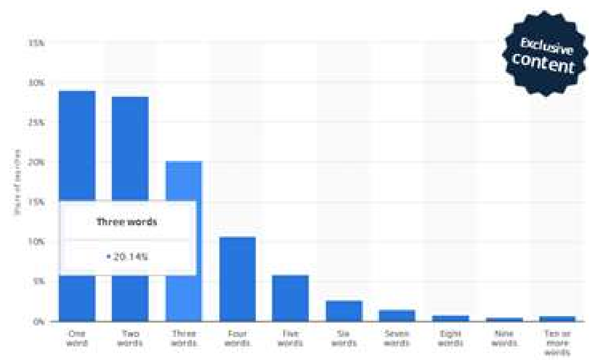


Figure 3. Average number of search terms for online search queries in the United States as of January 2016.

word sequences for semantic core generation[1], as well as the scratch of the future web-pages content suggestion can be automated in the context of the automatic text and automatic language generation tasks.

III. Practical application of the deep learning approach in the context of information retrieval

In regard to methods suitable to solve those tasks, along with summarization-based approaches such as extractive summarization [15], fractal summarization that is used to generate a brief skeleton of summary at the first stage, and the details of the summary on different levels of the document are generated on demands of users [16], linguistic and semantic analysis [17], the analysis revealed the machine learning methods can be used. It is worth to mention that methods of supervised learning are not effective enough due to the next limitations: they need large amount of annotated data for learning a proper task, which is often not easy available, and they fail to generalize for tasks other than they have been trained for. With the development of neural networks and deep learning approaches, models of vector representations of words [18], recurrent neural networks, Long-Short-Term-Memory (LSTM) or Gated Recurrent Unit (GRU) architectures and their variations and combinations [19-21] began to be used most often.

Taking into account typical length of the search query the choice of LSTM as the model for generation passages up to fourteen words length (that is a bit less than an average sentence length, for example for the well-known LOB Corpus [22] it is about nineteen words) implemented with the help of Tensorflow source-platform [23] and Keras framework [24] (main parameters are: ReLu, SoftMax activation functions, Adam optimizer) seems obvious. The dataset of about 200 thousand words length was created on the basis of a well-known MEDIQA-QA dataset for answer-ranking [25], encouraging research in medical question answering systems, and consisting of consumer health questions and passages selected from reliable online sources, as well as 120 thousand words length text corpus referred to IT-domain was collected from Wikipedia. Standard procedures of preprocessing related to cleaning, tokenization, and splitting into training and test data were performed.

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. The appearance of Generative Pre-trained Transformer (GPT) models by OpenAI gave ability to overcome limitations. The first one – GPT has the architecture which facilitated transfer learning and can perform various natural language processing tasks with very little fine-tuning. It also showed the power of generative-pre-training and opened up avenues for other models, which could unleash this potential better with lager datasets and more parameters. The second one – GPT-2 is a large transformer-based language model with 1.5 billion parameters, trained on a dataset of 8 million web pages. GPT-2 is trained with a simple objective: predict the next word, given all of the previous words within some text. The diversity of the dataset causes this simple goal to contain naturally occurring demonstrations of many tasks across diverse domains. GPT-2 is a direct scale-up of GPT, with more than 10X the parameters and trained on more than 10X the amount of data. [26] GPT-2 124M was fine tuned and used to generate the text passages up to the same length as the previous models as well as much more longer to create the web-page content template suitable for manual or semi-automatic postprocessing according to the objectives of the web-resource.

The examples of results are represented below ("Fig. 4"). The achieved text sequences can be further processed with the set of filters or stages that can be organized in a pipeline to identify named entities, syntagmata, taxanomic categories with the help of functionality provided by available standalone natural language and statistic processing libraries or linguistic processors. For example, at the first stage – the extraction of named entities – the names of programming languages were achieved: “Haskell”, “C”, “C++”, “Python” from a text fragment “languages like Haskell, C and C++, with a Python”, as well as, the names of the operating systems “Windows”, “Linux”, “macOS” – achieved from the input sequences “include Microsoft Windows, macOS” and “desktop operating system for Unix-like systems like Windows and Linux”. In particular, for identifying synonymy relations the multilingual lexical database MModWN can be used, due to the ability to take in consideration lexical units membership of the most informative lexico-grammatical, syntactical and semantic classes and achieve proper synonymic sets in different languages [9] or hybrid knowledge bases of ostis-systems and models for representing various types of knowledge within the framework of such a knowledge base. [27] The last one option might be useful if it is important to ensure possibility to use within the ostis-systems of various types of knowledge.

In regard to the web-page content template of the web-page, its structure depends on the type of the web-resource (a homepage, a magazine website, an e-commerce website, a landing page etc.) and can be

[1]The semantic core is a list of all words on the project’s topic, which serves as a basis for creating a common structure and individual pages of the site. An appropriate compilation of the semantic core will help the site to get into the top search engine results. [14]

Model	Input text	Output passage
LSTM	asthma attack requires	the first round of therapy
	Rasagiline is used	to treat anxiety disorders and other maternal illnesses to diagnose a heart
	asthma attack requires more	medicine to control your symptoms
	Pulmonary hypertension is high	blood pressure high blood cholesterol is lipid and a narrow
	Optical discs are	made of polycarbonate and can be useful to retain the data
GPT-2	the first standalone LCDs appeared	in the early 1970s, and follow such systems
	examples of operating systems	include Microsoft Windows, macOS
	high-level programming	languages like Haskell, C and C++, with a Python
	optical discs are used to	store data that can be transferred to, or read from
	macOS is the	desktop operating system for Unix-like systems like Windows and Linux

Figure 4. The examples of text passages achieved via LSTM and GPT-2 language models.

organized in unit-structure style, where the content for every item generates separately taking into account the sub-topic and related dataset, which is used to fine tune or prepare the language model discussed above.

IV. Conclusion

The analysis of the machine learning methods to solve the problem of automatic natural language generation as well as automatic text generation problems have been surveyed. A practical application of the deep learning approach in the context of information retrieval based on the usage of LSTM and GPT-2 language models has been performed. Proposed solution is language independent and can be reinforced with functionality of the natural language analysis provided by linguistic processor or standalone natural language and statistic processing libraries together with multilingual lexical databases or hybrid knowledge bases of ostis-systems.

References

[1] Bad Bot Report - Imperva. Available at: https://www.imperva.com/blog/bad-bot-report-2020-bad-bots-strike-back/. (accessed 2021, Jun).

[2] Reiter, E., Dale, R. Building applied natural language generation systemsn, *Natural Language Engineering*, 1997, vol. 3, no. 1, pp. 57–59.

[3] Van der Kaa, H. A. J. Krahmer, E. J. Journalist versus news consumer: The perceived credibility of machine written news. *Proc. of the Computation + Journalism Symposium New York, NY, USA.*, ACL, 2014.

[4] Firat F., Robot journalism. *The International Encyclopedia of Journalism Studies, 2019*, DOI: 10.1002/9781118841570.iejs0243.

[5] Google. Available at: https://www.google.com/. (accessed 2021, Jun).

[6] Yahoo. Available at: https://www.yahoo.com. (accessed 2021, Jun).

[7] Bing. Available at: https://www.bing.com. (accessed 2021, Jun).

[8] Search Engine Market Share Worldwide - StatCounter Global Stats. Available at: https://gs.statcounter.com/search-engine-market-share#monthly-202005-202105-bar. (accessed 2021, Jun).

[9] Krapivin, Y. Information Retrieval and Machine Translation in Solving the Task of Automatic Recognition of Adopted Fragments of Text Documents, *Proceedings of Open Semantic Technology on Intelligent Systems, International Conference, 21-23 Febr. 2019*: Belarusian State University of Informatics and Radioelectonics; ed. : V. Golenkov. – Minsk, 2019. – P. 289–292.

[10] U.S. online search query size 2020 – Statista. Available at: https://www.statista.com/statistics/269740/number-of-search-terms-in-internet-research-in-the-us/#statisticContainer. (accessed 2021, Jun).

[11] Online search usage - Statistics & Facts – Statista. Available at: https://www.statista.com/topics/1710/search-engine-usage/?#topicHeader_wrapper. (accessed 2021, Jun).

[12] Ubersuggest's Free Keyword Tool, Generate More Suggestions. Available at: https://neilpatel.com/ubersuggest. (accessed 2021, Jun).

[13] Serpstat — Growth hacking tool for SEO, PPC and content marketing. Available at: https://serpstat.com/. (accessed 2021, Jun).

[14] Six ways to collect a semantic kernel for your site: tools overview. Available at: https://serpstat.com/blog/how-to-collect-a-semantic-core-for-a-site/. (accessed 2021, Jun).

[15] Voronkov N.V. Metody, algoritmy i modeli sistem avtomaticheskogo referirovanija tekstovyh dokumentov.dis. kand. teh. nauk, Minsk, 2007.165 p.

[16] Yang, C., Wang, F. Fractal Summarization for Mobile Device to Access Large Documents on the Web // WWW '03: *Proceedings of the 12th international conference on World Wide Web*, 20 May. – 2003. – P. 215–224.

[17] Lipnitsky S.F., Stepura L.V., J. Algoritmy sozdaniya giperteksta na osnove situativno-sintagmaticheskoi seti [Algorithms of hypertext creation based on situational-syntagmatic network]. Izvestiya Natsionalnoi akademii nauk Belarusi. Seriya fiziko-tehnicheskih nauk, 2010, no. 3, pp.90-95.

[18] Kim D. [et al.] Lexical feature embedding for classifying dialogue acts on Korean conversations. *Proc. of 42th Winter Conference on Korean Institute of Information Scientists and Engineers*, 2015, pp. 575–577.

[19] Ravuri S., Stolcke A., Recurrent neural network and LSTM models for lexical utterance classification. *Proc. of 16th Annual Conference of the International Speech Communication Association*, 2015, pp. 135–139.

[20] Dey R., Salemt F., Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks. *IEEE 60th International Midwest Symposium on Circuits and Systems*, IEEE, 2017, pp. 1597–1600.

[21] Hochreiter, S., Schmidhuber, J. Long Short-term Memory. *Neural computation*, 1997, vol. 9, no. 8, pp. 1735–1741.

[22] Johansson, S. Manual of information to accompany the Lancaster-Oslo-Bergen Corpus of British English, for use with digital computers. Available at: http://clu.uni.no/icame/manuals/LOB/INDEX.HTM/. (accessed 2021, Jun).

[23] TensorFlow. Available at : https://www.tensorflow.org/. (accessed 2021, Jun).

[24] Keras. Available at: https://keras.io/. (accessed 2021, Jun).

[25] Question-Driven Summarization of Answers to Consumer Health Questions. Available at: https://arxiv.org/abs/2005.09067. (accessed 2021, Jun).

[26] Better Language Models and Their Implications. Available at: https://openai.com/blog/better-language-models/. (accessed 2021, Jun).

[27] Davydenko, I. "Semantic models, method and tools of knowledge bases coordinated development based on reusable components," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2018, pp.99–118.

Подход на основе глубокого обучения в контексте информационного поиска для решения задач как автоматического синтеза естественного языка так и автоматического синтеза текста

Крапивин Ю.Б.

В статье представлено решение практического применения подхода глубокого обучения в контексте поиска информации на основе использования языковых моделей LSTM и GPT-2 для решения задач как автоматического синтеза естественного языка, так и автоматического синтеза текста.

Ключевые слова: естественный язык, информационный поиск, автоматический синтез естественного языка, автоматический синтез текста, машинное обучение.

Received 04.07.2021

Extractive Russian Text Summarization as Greedy Sentence Sequence Continuation Search with Probabilities from Pretrained Language Models

Irina Polyakova
Department of Algorithmic Languages
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, Russia
polyak@cs.msu.ru

Semyon Pogoreltsev
Department of Algorithmic Languages
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, Russia
pogorelcevsa@gmail.com

***Abstract*—Pretrained language models based on Transformer have defined new state-of-the-art result on a wide range of tasks being finetuned or used as embedders. Models with Multi-head Self-Attention mechanism have defined a new standart of quality on text summarization task in English, while Transformer based models potential for extractive Russian text summarization has been barely explored. We propose a new method for extractive Russian text summarization, reducing the task to the selection of the most probable sequence of sentences. The new method beats ROUGE-1 and ROUGE-L scores of other models such as SummaRuNNer, and mBART evaluated on Gazeta dataset and is more preferred in human evaluation poll.**

***Keywords*—Russian Text Summarization, Pretrained Language Models, BERT, Sentence-BERT, Next Sentence Prediction**

I. Introduction

Text summarization is the task of creating a shorter version of a document that captures essential information. Automatic summarization approaches can be extractive or abstractive.

Extractive methods form a summary as a combination of the original text's chunks. Extraction is usually reduced to classifying sentences of the initial document. The resulting summary is grammatically correct, especially in the case of sentence copying. But extractive methods can't produce generalized and paraphrased text, which is essential for high-quality compression. In addition, these algorithms usually don't structurize summaries.

Abstractive approaches generate a new text - a generalization of the original one's ideas. These models can generate new words that are not from the original text, which leads to better generalization abilities. It allows them to compress text in a better way via sentence fusion and paraphrasing, but overall complexity, errors in generated texts' grammar, unexpected results in some cases and weak interpretability still limit the use of these methods in various projects.

We introduce a new extractive summarization approach for Russian texts, which leverages pretrained BERT(Bidirectional Encoder Representations from Transformers) [1] and Sentence-BERT [2] models. The proposed method forms a summary that has human-natural storytelling order of sentences. The algorithm can easily be adapted for other languages or be multilingual provided that there are such BERT models.

II. Related works

A. Pretrained language models

Pretrained language models such as BERT [1] are now a key technology in NLP industry being extensively used with finetuning, few-shot learning, or as embedders. State-of-the-art approaches in most NLP tasks are based on neural networks with Transformer [3] architecture. The main building block of the Transformer model is the Multi-head Self-Attention mechanism

$$\mathrm{MHA}(Q, K, V) = \mathrm{Concat}(head_1, ..., head_n)W^o \quad (1)$$

$$\mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\,(\frac{Q * K^T}{\sqrt{d_k}})V \quad (3)$$

where V - values, K - keys, Q - queries, $d_k = dimK$ In the case of self-attention Q, K, V are from the same source, so relations between tokens of the sequence are estimated.

"Bidirectional Encoder Representations from Transformers(BERT)" [1] is a new language model which is trained with a masked language modeling and a "next sentence prediction” tasks on large unstructured single language or multilingual text corpora. BERT is a stack of Transformer encoders.

Sentence-BERT [2] is a modification of the pre-trained BERT model that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings. These semantic vectors are meant for comparison using cosine-similarity.

“Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language” [4] proved that Transformer-based models are useful for Russian. The authors of the paper also prepared pre-trained BERT models’ weights for Russian.

B. Extractive Summarization

SummaRunNNer [5] is one of the first approaches, using neural networks for extractive text summarization. It leverages an RNN-based encoder on sentence and document levels for semantic embeddings which are then used for binary sentence classification.

Refresh [6] model from “Ranking Sentences for Extractive Summarization with Reinforcement Learning” [6] uses CNN and RNN layers for sentence and document embedding. Refresh is trained with the new algorithm proposed in the paper, which globally optimizes ROUGE [7] evaluation score through a reinforcement learning objective.

“Single Document Summarization as Tree Induction” [8] leverages structured attention to induce a multi-root dependency tree representation of the document while predicting the output summary.

This idea was further studied in “HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization” [9].

Methods, proposed in the “Text Summarization with Pretrained Encoders” [10] made a big leap in ROUGE [7] metrics on the CNN/Daily Mail dataset. The extractive model used pre-trained BERT model, finetuned with inter sentence Transformers for sentence classification.

The current state-of-the-art extractive model on the CNN/Daily Mail was introduced in ‘Extractive Summarization as Text Matching” [11] work. Sentence extraction task is reduced to semantic text matching with Sentence-BERT embeddings.

III. Sentence-BERT for Text Centrality Ranking

The algorithm uses cased Sentence-BERT, which was fine-tuned on SNLI [12] Google-translated to Russian and on the Russian part of XNLI [13] dev set.

Sentence embeddings from Sentence RuBERT are being ranked by cosine-similarity

$$\mathrm{sim} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} \tag{4}$$

where A, B are the semantic representation vectors that are compared. Cosine-similarity is considered a default method f or semantic similarity comparison used for similar tasks, such as NLI(Natural Language Inference).

This metric is human-interpretable.

$$\mathrm{sim} \in [0, 1] \tag{5}$$

Higher cosine-similarity means that sentences are closer.

Central text embedding text-central is a selected sentence semantic vector or mean pooled representation from all sentences of the document. The experiments have show that for news texts it’s better to use the first sentence or mean pooling of all sentences and the last option is better in terms of ROUGE [7] scores.

$$\text{text-central} = \frac{1}{n}\sum_{i=0}^{n} \text{sent-emb}_i \tag{6}$$

where n is the number of sentences in text, sent-emb_i is the Sentence-BERT embedding of i-th sentence in the text

Then for each i-th sentence r-score_i is computed

$$\text{r-score}_i = \mathrm{sim}(\text{s-emb}_i, \text{text-center}) \tag{7}$$

where $i = 1, 2, ..., n$ These scores are min-max normalized

$$\overline{\text{r-score}_i} = \frac{\text{r-score}_i - \min * \text{r-score}_i}{\max * \text{r-score}_i - \min * \text{r-score}_i} \tag{8}$$

$\text{r-score}'_i$ is a normalized centrality measure, which is used in a latter formula.

IV. Sequential Next Sentence Prediction

A. Next Sententence Prediction

BERT models are pre-trained on the task of Next Sentence Prediction(NSP) as well as Masked Language Modelling. In the BERT training process, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. During training, 50% of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while the other part consists of random sentences from the corpus. The assumption is that the random sentence will be disconnected from the first sentence.

The following steps are needed to estimate a probability that the sentence B is a continuation of sentence A:

1) A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.
2) A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2.

3) A positional embedding is added to each token to indicate its position in the sequence
4) The entire input sequence goes through the Transformer model.
5) The output of the [CLS] token is transformed into a 2×1 shaped vector, using a simple classification layer.
6) Calculating the probability with softmax.

Extractive summarization can be reduced to greedy sequential sentence selection maximizing the probability of each chosen sentence starting from the one that is the best generalization of the text. With this approach sentences are also ordered in a natural for human reading way, forming a story.

This heatmap(1) is an example of matrix on 5 sentence sample from the Gazeta [14] dataset. Each element is computed with formula (9)

$$\overline{P}(s_i|s_j) = \frac{P(s_i|s_j) - \min *P(s_i|s_j)}{\max *P(s_i|s_j) - \min *P(s_i|s_j)} \quad (9)$$

$$P(s_i|s_j) : i, j \in [0, n), i, j \in \mathbb{N}, i \neq j \quad (10)$$

Figure 1. Example of a min-max normalized NSP matrix heatmap

B. Contextualized Next Sentence Prediction

In original NSP it's supposed that comparing a pair of sentences is made without context. For extractive summarization sentences which were already selected for the summary can be used. Self-Attention mechanisms benefit from additional context, so probabilities estimation is more precise in terms of summary, not just the last selected sentence.

$$P(s_i|s_1, s_l, ..., s_k) = \mathrm{NSP}(C + s_1 + \sum_{j=l}^{k}(s_k) + S + s_i + S) \quad (11)$$

s_i is the i-th sentence, C is a [CLS] token, S is a [SEP] token, $l \in [2, k-1], l \in \mathbb{N}$ 1 is a skipping numerator, $k \in [0, m-1], k \in \mathbb{N}$ is a number of sentences in the summary that were already chosen, except the first one, m is a number of sentences that is expected to be in a summary.

l sentences between the start and the last selected are skipped if the sequence of tokens exceeds the input size of BERT. The first and the last ones are always in the context for probability estimation, because the first one is the most generalizing and the last is the one that should be naturally continued.

The probabilites are also min max normalized

$$\overline{P}(s_i|s_1, s_l, ..., s_k) = \frac{P(s_i|..., s_k) - \min *P(s_i|..., s_k)}{\max *P(s_i|..., s_k) - \min *P(s_i|..., s_k)} \quad (12)$$

where min, max are selected from the probabilites of one step. It's generally better to normalize on the whole matrix, but to reduce computational complexity and memory usage only the list of next possible sentences is used.

C. Extractive Summarization of News Texts

The summary is formed iteratively with the sentences that locally maximise the $\mathrm{SC}(s_1, ..., s_k \rightarrow s_i)$ score

$$\mathrm{SC}(s_1, ..., s_k \rightarrow s_i) = \overline{P}(s_i|s_1, s_l, ..., s_k) + \alpha * \overline{\text{r-score}_i} \quad (13)$$

where α is a coefficient of central sorting importance, by default $\alpha = 0.05$,

The starting element is the first sentence of the text, because in news texts, like the ones in the Gazeta [14] dataset, it's the most important, generalizing the whole text.

Using $\text{r-score}_i'$ in the $\mathrm{SC}(s_1, ..., s_k \rightarrow s_i)$ formula with a quite small α helps to control the semantic deviation from the central topics of the text. It's experimentally proven that choises based on pure $P(s_i|s_1, s_l, ..., s_k)'$, especially the variant without left context $P(s_i|s_k)'$, are less central to the main text topics in later iterations. It happens because heads of self-attention mechanisms are less useful with smaller context given as an input.

V. Results

A. Automatic Evaluation

In "Dataset for Automatic Summarization of Russian News" [14] some popular and recent summarization methods were evaluated on the test part of the dataset:

- TextRank [15]
- LexRank [16]
- LSA(Latent Semantic Analysis) [17]
- SumaRuNNer [5]
- Pointer Generator [18]
- CopyNet [19]

There were also published results of mBART [20] fine-tuned on the Gazeta dataset in the paper. mBART based method is a current state-of-the-art in abstractive English text summarization on the CNN/Daily Mail dataset.

The mean sentence number of human summaries in the dataset is 3, so our method compressed the texts into 3 sentences.

Here are the results on test part of the dataset compared with other methods that were evaluation in Gazeta's work [14]:

Approach	ROUGE-1	ROUGE-2	ROUGE-L
Lead-1	27.6	12.9	20.2
Lead-2	30.6	13.7	25.6
Lead-3	31.0	13.4	26.3
Greedy Oracle	44.3	22.7	39.4
TextRank	21.4	6.3	16.4
LexRank	23.7	7.8	19.9
LSA	19.3	5.0	15.0
SumaRuNNer	31.6	13.7	27.1
Proposed method	35.6	14.2	32.4
CopyNet	21.4	6.3	16.4
PG small	23.7	7.8	19.9
PG words	19.3	5.0	15.0
PG big	29.6	12.8	24.6
PG small+coverage	30.2	12.9	26.0
Finetuned mBART	32.1	14.2	27.9

All these values are F-scores of ROUGE [7] metric.

Lead-N as in the "Dataset for Automatic Summarization of Russian News" [14] paper is just N first sentences taken.

Greedy Oracle is a method that uses human-written summaries, choosing sentences which are maximizing ROUGE-2 [7] score. It's useful for training classifiers sentence classfiers and as a target value for extractive summarization algorithms, defining a max reachable score.

Our algorithm is better than other ones at ROUGE-1 and ROUGE-L scores. [7] In ROUGE-2 [7] it's equal to SummaRuNNer and fine-tuned mBART methods. So, our method is the new state of the art on Gazeta, as there are no other results published on this dataset

B. Human Evaluation

Current summarization automatic evaluation methods do not score in any way the improvements in sentence order. We organized an anonymous opinion poll among students from other departments, most of whose are not related to natural language processing, machine learning and didn't have advanced algorithms courses.

We had involved 63 students to participate in the poll.

Each pole got an example from the test part of the used dataset. In both forms were given a headline, text, human-made summary, and the ones of our algorithm marked as "Summary 1" and "Summary 2".

Students answered the question "Which summary do you like more?" with the following possible options:

- Summary 1
- Summary 2
- Both great
- Both poor

Summary 1 and Summary 2 options are randomly mixed options Algorithm and Human.

The results are shown in the following table and are visualized(2).

Preferred	Count	Percentage
Algorithm	21	33.3%
Human	17	26.98%
Both great	16	25.4%
Both poor	9	14.29%

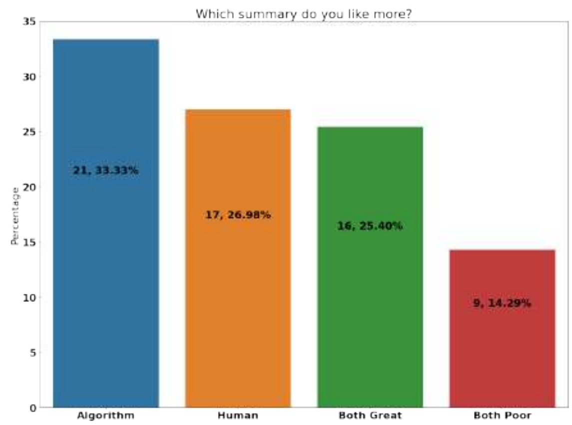


Figure 2. Student Poll Results

VI. Examples

A. Gazeta Dataset

Title: Почему мы остываем: температура человека снизилась за 200 лет

Original text: Температура человеческого тела снизилась за последние 200 лет, выяснили исследователи из Стэнфордского университета. О своем открытии они рассказали в статье в журнале eLife. Температура тела здорового человека колеблется в течение суток в пределах 35,5-37,2°C. Такая температура считается оптимальной для поддержания нормальной работы внутренних органов и протекания биохимических реакций, а также позволяет сдерживать грибковые инфекции. Хотя средней с XIX века считалась температура в 37°C, сегодня она считается повышенной и многим при таком показателе нездоровится. Раньше это списывалось на неточность измерений температуры в прошлом, но теперь оказалось, что люди с XIX века действительно «остыли». Чтобы выяснить, что на самом деле произошло, профессор Джули Парсоннет и ее команда объединили три набора данных. Первый из них охватывал почти 24 тыс. ветеранов Армии Союза времен Гражданской войны в США, температура которых измерялась в период между 1860 и 1940 годами. «Мне потребовалось много времени, чтобы найти данные XIX века, где была бы информация о температуре тела», — отмечает Парсоннет. Остальные наборы данных охватывали период с 1971 по 1975 год и с 2007 по 2017 год. В общей сложности команда проанализировала измерения температуры тела 677,5 тыс. человек. В среднем, температура тела людей снижалась на 0,03°C за десятилетие. У мужчин, родившихся в начале XIX века, температура тела была на 0,59°C выше, чем у мужчин сегодня. Данные по женщинам стали собираться несколько позже. Как выяснилось, температура их тела снизилась на 0,32°C с 1890-х годов. Средняя температура тела сегодня составляет около 36,6°C. Парсоннет предлагает два доказательства того, что дело именно в снижении температуры тела, а не в ненадежности старых термометров. Во-первых, тенденция к снижению температуры тела прослеживается и в поздних исследованиях, где использовались уже более точные термометры. «Изменения, которые происходили в 1860-1960-х годах, мы наблюдаем также в период с 1960-х годов по сегодняшний день, — говорит Парсоннет. — Я не думаю, что есть большая разница в термометрах между 1960-х годов и современными». Во-вторых, у пожилых людей температура тела была выше, чем измеренная в том же году у более молодых людей, причем разница была примерно одинаковой независимо от года. Также, сравнив несколько групп по возрастам, исследователи установили, что температура тела снижалась и у молодых, и у пожилых людей в одинаковой степени. Если бы проблема была в термометрах, то выявить такие точные различия было бы сложно, считает Парсоннет. В 1800-е годы люди страдали от малярии, туберкулеза, дизентерии, болезней полости рта и многих других продолжительных или хронических заболеваний, отмечает она. Сейчас с большинством болезней удалось справиться, что и могло повлиять на снижение температуры тела: организму не нужно больше постоянно бороться с инфекцией. «На мой взгляд, дело в том, что, с микробиологической точки зрения, мы очень отличаемся от людей прошлого, — говорит Парсоннет. — У современных людей меньше инфекций, благодаря вакцинам и антибиотикам, поэтому наша иммунная система не так активна, а ткани организма менее воспалены. Если это так, то температура тела должна была бы понизиться и в других странах, где здоровье людей улучшилось». Вряд ли в ближайшее время температура тела человека заметно понизится, считает Парсоннет. «До нуля мы не остынем, — шутит она. — Есть какой-то предел, просто я пока не знаю, где он». Как изменилась температура тела человека в большинстве других стран, еще предстоит узнать. Однако, например, результаты британского исследования 2017 года повторяют выводы Парсоннет. Измерив температуру тела 35,5 тыс. здоровых британцев, исследователи выяснили, что в среднем она составляет 36,6°C. Однако пожилые британцы, в отличие от американцев, были «прохладнее» молодых. Также температура зависела и от расовой принадлежности: темнокожие респонденты оказались «горячее».

Human summary: Температура человеческого тела снизилась за последние 200 лет, выяснили американские исследователи. В 1800-е годы она составляла 37°C, причиной чему, вероятно, были продолжительные и хронические болезни. Сегодня люди более здоровы — и более «прохладны».

Machine summary: Температура человеческого тела снизилась за последние 200 лет, выяснили исследователи из Стэнфордского университета. У мужчин, родившихся в начале XIX века, температура тела была на 0,59°C выше, чем у мужчин сегодня. Раньше это списывалось на неточность измерений температуры в прошлом, но теперь оказалось, что люди с XIX века действительно «остыли».

B. Artice Introduction

This is a summary of this article's introduction section with multilingual BERT and Sentence-BERT models:

Text summarization is the task of creating a shorter version of a document that captures essential information. Extractive methods form a summary as a combination of the original text's chunks. These approaches are usually reduced to classifying sentences of the initial document. We introduce a new extractive summarization approach for Russian texts, which leverages pre-trained BERT(Bidirectional Encoder Representations from Transformers) [1] and Sentence-BERT [2] models. The proposed method forms a summary that has human-natural storytelling order of sentences.

VII. Conclusion

Most modern extractive text summarization techniques use pre-trained language models as a decent linguistic feature aware foundation, that can be fine-tuned with additional layers or used for embedding and next sentence prediction probability estimation.

There are many works about English and other popular languages that leverage modern methods which are significantly better than older approaches, but, as for the Russian there are barely any works that use attention-based models, which have been proved to work better with longer sequences than RNNs, whereas text summarization is all about sequences processing.

In this paper, we introduced a new approach to extractive text summarization, which uses pre-trained language models and can easily be used for other languages. The new algorithm orders sentences in a human-natural way, making summaries easier to read. The proposed summarization method achieved higher ROUGE-1 and ROUGE-L [7] scores on the Gazeta dataset and was preferred more in human evaluation.

As for the future, we plan to go on with exploring pre-trained language models' possibilities in semantically driven tasks, especially abstractive summarization.

Acknowledgment

The work was carried out within the framework of the research "Mathematical and software support of advanced systems for processing symbolic information with elements of artificial intelligence" which is led by the department of algorithmic languages on the faculty of computational mathematics and cybernetics of the Lomonosov Moscow State University.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019, arXiv preprint arXiv:1810.04805. Available at https://arxiv.org/pdf/1810.04805.pdf (accessed 2020, Dec)

[2] Nils Reimers, Iryna Gurevych Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 2019, arXiv preprint arXiv:1908.10084. Available at https://arxiv.org/pdf/1908.10084.pdf (accessed 2020, Dec)

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in Neural Information Processing Systems, 2017, pp. 6000–6010.

[4] Yuri Kuratov, Mikhail Arkhipov Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. 2019, arXiv preprint arXiv:1905.07213. Available at https://arxiv.org/pdf/1905.07213.pdf (accessed 2021, Jan)

[5] Ramesh Nallapati, Feifei Zhai, Bowen Zhou SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. 2016, arXiv preprint arXiv:1611.04230. Available at https://arxiv.org/pdf/1611.04230.pdf (accessed 2020, Nov)

[6] Shashi Narayan, Shay B. Cohen, Mirella Lapata Ranking Sentences for Extractive Summarization with Reinforcement Learning. 2018, arXiv preprint arXiv:1802.08636. Available at https://arxiv.org/pdf/1802.08636.pdf (accessed 2021, Mar)

[7] Chin-Yew Lin ROUGE: a Package for Automatic Evaluation of Summaries. Association for Computational Linguistics, 2004, vol. Text Summarization Branches Out, pp. 74–81

[8] Yang Liu, Ivan Titov, Mirella Lapata Single Document Summarization as Tree Induction. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, vol. 1 (Long and Short Papers), pp. 1745-1755

[9] Xingxing Zhang, Furu Wei, Ming Zhou HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization. 2019, arXiv preprint arXiv:1905.06566. Available at https://arxiv.org/pdf/1905.06566.pdf (accessed 2021, Dec)

[10] Yang Liu, Mirella Lapata Text Summarization with Pretrained Encoders. 2019, arXiv preprint arXiv:1908.08345. Available at https://arxiv.org/pdf/1908.08345.pdf (accessed 2021, Apr)

[11] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, Xuanjing Huang Extractive Summarization as Text Matching. 2020, arXiv preprint arXiv:2004.08795. Available at https://arxiv.org/pdf/2004.08795.pdf (accessed 2021, Apr)

[12] Samuel R. Bowman, Gabor Angeli, Christopher Potts, Christopher D. Manning A large annotated corpus for learning natural language inference. 2015, arXiv preprint arXiv:1508.05326. Available at https://arxiv.org/pdf/1508.05326.pdf (accessed 2021, Mar)

[13] Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, Veselin Stoyanov XNLI: Evaluating Cross-lingual Sentence Representations. 2018, arXiv preprint arXiv:1809.05053. Available at https://arxiv.org/pdf/1809.05053.pdf (accessed 2021, Mar)

[14] Ilya Gusev Dataset for Automatic Summarization of Russian News. 2020, arXiv preprint arXiv:2006.11063. Available at https://arxiv.org/pdf/2006.11063.pdf (accessed 2021, Feb)

[15] Rada Mihalcea, Paul Tarau TextRank: Bringing Order into Text. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004, pp. 404–411

[16] Gunes Erkan, Dragomir R. Radev LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research, 2004, vol. 22, pp. 457-479

[17] P. W. Foltz Latent semantic analysis for text-based research. Behavior Research Methods, Instruments, & Computers, 1996, vol. 28, pp. 197–202

[18] Abigail See, Peter J. Liu, Christopher D. Manning Get To The Point: Summarization with Pointer-Generator Networks. 2017, arXiv preprint arXiv:1704.04368. Available at https://arxiv.org/pdf/1704.04368.pdf (accessed 2021, Feb)

[19] Jiatao Gu, Zhengdong Lu, Hang Li, Victor O.K. Li Incorporating Copying Mechanism in Sequence-to-Sequence Learning. 2016, arXiv preprint arXiv:1603.06393. Available at https://arxiv.org/pdf/1603.06393.pdf (accessed 2021, Feb)

[20] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, Luke Zettlemoyer Multilingual Denoising Pre-training for Neural Machine Translation. 2020, arXiv preprint arXiv:2001.08210. Available at https://arxiv.org/pdf/2001.08210.pdf (accessed 2021, May)

Извлекающее автореферирование русскоязычных текстов с применением предобученных языковых моделей

Полякова И.Н., Погорельцев С.А.

Предобучееные языковые модели на основе архитектуры Transformer установили новые рекорды на многих задачах обработки естественных языков при их дообучении или использовании для получения контекстуализированных семантических векторов. Модели с механизмом многоголового внутреннего внимания стали лучшими на задаче автореферирования англоязычных текстов, однако возможности применения подобных подходов для русского языка слабо изучены. Мы представляем новое решение задачи автореферирования текстов на русском языке, которое достигает лучших результатов по нескольким метрикам относительно других моделей, таких как SummaRuNNer и дообученный mBART для генерирующего автореферирования на датасете Gazeta. Результаты работы предлагаемого алгоритма является более предпочтительным вариантом в опросе среди студентов.

Received 31.05.2021

Dynamic features selection in authorship identification problem

Anton Paramonov, Ilya Trukhanovich, Uladzimir Kuntsevich
Belarusian State University Informatics and Radioelectronics
Minsk, Republic of Belarus
anton_paramonov@tut.by, ilya.trukhanovich@gmail.com, vkyntsevich@gmail.com

***Abstract*—The work is devoted to the problems of feature selection in the method of authorship identification in the context of authorial invariant defining. The problem of text author identification and existing approaches are described. In this article, models, methods, and experiment results for implementation of the dynamic method of defining features by genetic algorithm.**

***Keywords*—text author identification, texts classification, authorial invariant, genetic algorithm, support vector machine**

I. Introduction

The task of attribution of unknown text is an important information problem.

This is mainly due to the widespread use of messaging programs on the Internet, the increasing importance of e-mail for corporate correspondence, the popularity of forums and blogs. Without registering, users can send messages and specify their own information and registration is frequently simply symbolic. It is the same for e-mails and messengers. It means that the registration data cannot identify the contact person unambiguously, the address of the sender can be easily changed. Increasingly, the anonymity of internet messages is attracting cyberspace criminals [1].

In other areas such methods can also be applied. These techniques can be used in linguistic research to study authorship phenomena. The difference in this or that writer's style is of interest. Features that make his speech as individual or common characteristics of any characteristics easily visible. Authorship is unknown in a number of unassigned literary texts. It is evident that the existence of precise quantitative methods for the identification and evaluation of the author can resolve most controversial historical and literary criticism issues [2].

Education is another field of application. Students tend to do their own tasks less and prefer to spend less time and use prepared results. A more objective assessment approach will in this case be possible by using attribution methods [3].

The area uses machine learning, information retrieval and natural language processing approaches in a relatively interdisciplinary way. The authorial invariant is usually used as a "handwriting". This is a quantitative feature of literary texts or a parameter that uniquely characterizes one author's work or a small number of "closer authors" by their behavior and values in different groups of authors [4].

II. Problem Statement

The problem is formulated as follows when identifying the author of a text with a small number of alternatives. Assume that we have texts set $T = \{t_1, ..., t_n\}$ and authors set $A = \{a_1, ..., a_k\}$. We know authors for some subset $T_1 = \{t_1, ..., t_l\} \subseteq T$ so we have pairs like $\{a_i, t_j\}$ $(i = 1..k, j = 1..l)$ as training set L. The true authors of the remaining texts subset T_2 must be established.

In this context the problem of authorship can be seen as a problem in several classes. In this case, L is training set, A is a set of predefined classes, and T_2 is objects for classification. The aim is to create a classifier that solves this problem i.e. find a function that gives its true author an arbitrary text of set T.

A sequence of the following actions is part of the general technique for identifying the author of an unknown text:

- Selecting a text model in the form of feature sets.
- Select a group of characteristics for checking and forming an invariant of the author.
- The selection and the parameters for the classifier.
- Formation of an author's style model allowing two or more authors to be separated by a trained and invariant author.
- The authorship of the unknown text will be determined directly.
- The final decision by the classifier on the author of the text should be adopted if several groups of informative text features could be found.

You can use the word bag model and the n-gram one to represent texts in the information system. The word model bag is a collection of all words (or word attributes) which compose the text unordered. Text is defined as a sequence of n-element strings in the N-gram model [5].

The following sequence of measures is proposed to determine the differences in the authors' styles:

- Division into two groups of the existing set of texts. The first is for training the model classifier. Secondly, the identification author's accuracy is checked by using a trained model.
- Formation in accordance with the selected model of text representation in the form of a set of characteristics of a text vector characteristics from the invariant of the obtained author.
- Run attribute values into a single range through standardization and scaling operations.
- Correction by training the classifier in the normalized vectors of the features of the training text groups to ensure a high degrees of separation capability of the authors and verifying the accuracy of the trained classifier in the feature vectors of the test texts group.
- Change the list of the characteristics and/or property groups that constitute the group if it is not possible to achieve acceptable results by changing the parameters of the classification system.

III. Existing approaches

At the moment, a certain number of different solutions have been worked out.

K-means, support vector machine, neural network and other approaches were used as classifier basis.

Relevance of SVM as classifier basis was proved significantly by Thorsten Joachims [6]. Later it was tested by Rong Zheng and his colleagues [7].

Also neural network approach becomes more popular, especially in recent years, because of scientists from Stanford University [8] [9]. But it still requires significant improvements because of low accuracy in different cases and a lot of resources to select architecture and train.

The question of the set of features that make up the author's invariant is under discussion. Usually approaches are based on stylistics, syntactic, lexical and other features.

But nowadays a lot of approaches use low-level features like punctuation frequency, average length and so on and they works rather well with above 80% accuracy, It was demonstrated by scientists of Pace University and University of Sheffield [10] [11]. Besides we should admit that feature set for literary books is significantly determined. The fact is widely used in papers like work of University of Ottawa [12]. But for common cases it is still full of uncertainty.

IV. Supposed approach

Let's define key parameters of supposed approach. Text can be represented as feature set and success of classification depends on feature combinations quality.

Lexical, syntactic, structural, content-specific, style and other features can be used as author's invariant. Despite the use of completely different levels of abstraction, completely different ones can give acceptable results within the scope of the classification.

Using features from higher levels of hierarchy, the analyzes of the structure of the text are made more complicated and difficult to automate with each new level. Due to the level of noise in the analysed text, the language characteristics, etc, and others, for example, inaccuracies may arise at each stage, leading to serious errors at higher analysis levels. It was therefore decided in this study to focus on the characteristics of the levels of chars and words.

Along with all this, as mentioned above, at present the question of the conventional set of features that make up the author's invariant remains open. Due to the mentioned above reasons this is caused by a wide range of applied problems, and as a consequence by a wide variety of texts domains. Features of a certain level of abstraction can work well within the scope of differences in one type of text (style features during classifying literary books), but rather poorly when trying to classify analytical reviews in blogs.

Thus, it seems theoretically justified to test an approach that, within the scope of the task and the proposed texts, allows you to select the optimal situational features. In this work, it is proposed to use a genetic algorithm.

The algorithm consist of the next steps:

- Create initial set of supposed features.
- Enter the loop
- Add new feature to the set
- Run genetic algorithm and select appropriate features
- If selected features have appropriate accuracy exit the loop
- Use selected features

V. Experiment

A. Initial parameters

We can take arbitrary Reuters news feed articles pack for for research purposes. We will take 2, 5 and 10 authors from the dataset and 50 texts for each one. We will include articles of known authors in test set.

We can use Support Vector Machine (SVM) as classificator basement. The SVM is a supervised learning machine algorithm that can be used for both classification and regression challenges. But it is mostly used in problems with classification. The value of each feature is a value in a given coordination, so that each data element is drawn as a point in N-dimensional space in the SVM algorithm, where n is the number of features. Then we classify by finding the hyperplane that very well distinguishes classes.

Initial features can be selected based on those that have already proven themselves well (table I).

Table I
INITIAL FEATURES

Feature name	Feature description
DICTIONARY-M	M most frequent words from language dictionary
WORDS-M	M most frequent words from sample dictionary
UNIGRAM-M	M most frequent unigrams
BIGRAM-M	M most frequent bigrams
TRIGRAM-M	M most frequent trigrams
TETRAGRAM-M	M most frequent tetragrams
PENTAGRAM-M	M most frequent pentagrams
POS	Parts of speech frequency
AVERAGE-WORD	Average word length in characters
AVERAGE-SENTENCE	Average sentence length in words

Fig. 1 displays dependence of accuracy for 5 authors on dimension for some features (DICTIONARY (blue), WORDS (green), TRIGRAM (orange)).

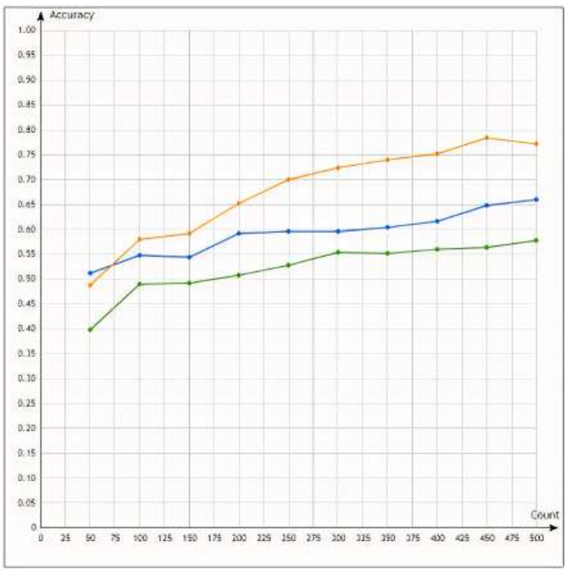


Figure 1. Dependence of accuracy for 5 authors (one feature)

Combining of features is associated with various effects. Let's take a look at this (Fig. 2).

Blue points are accuracy values for trigrams, gray ones are values for tetragrams. Orange points are values for combinations of trigrams and tetragrams. We can see significant increase of accuracy for some orange points but at the same time some ones show lower accuracy. That's why it is so important to choose feature combination carefully.

B. Results

According to experimental results we have 98% average accuracy for 2 authors, 84% accuracy for 5 authors and 65% accuracy for 10 authors. Best results uses DICTIONARY-M, WORDS-M and N-GRAM features (table II).

The results can be explained by the specificity of supposed texts.

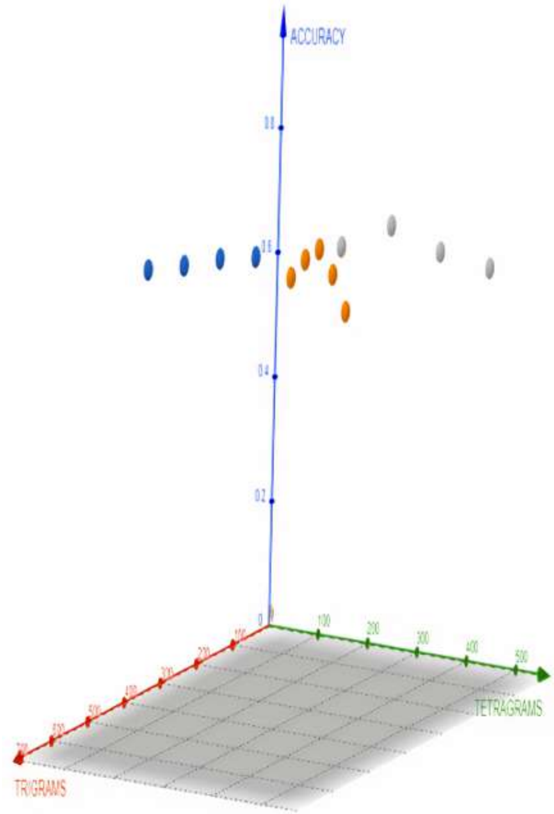


Figure 2. Dependence of accuracy for 5 authors (several features)

Table II
RESULTS

Number of authors	Average accuracy (test set)
2	98%
5	84%
10	65%

One of the most interesting features are expressive means but we didn't include them in initial feature set because they are not very relevant for news articles.

Words are the basis for authorial invariant so it is not surprising these features were selected. They underline text nature especially in our cases. At the same time N-GRAM feature is usually highly recommended and we can see that in our experiments they are relevant.

Other features are not so effective because, for example, parts of speech frequency does not shows personality for proposed news texts well because of their topics.

VI. CONCLUSION

Introduced approach has great potential because of its flexibility. Despite the fact that some aspects of text authorship are investigated a lot of patterns are not noticeable. Moreover, unexpected pattern combinations can lead to efficient increasing of accuracy. After feature selecting we can investigate their influence after the fact.

And this approach still have growth points:

- Selecting classifier basis along with features
- Integrating as a part in hybrid classifier system
- Using several selecting rounds with different abstraction levels of features
- Analysing threshold values for determining unknown authors

References

[1] R. Zheng, Y. Qin, Z. Huang Authorship Analysis in Cybercrime Investigation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2003, vol. 2665, pp. 59-73.

[2] R. Griffin Anonymity and Authorship. *New Literary History*, 1999, vol. 30, no 4, pp. 877-895.

[3] Types of plagiarism. Available at: https://www.bradford.ac.uk/library/find-out-about/plagiarism/types-of-plagiarism/ (accessed 2021, Mar)

[4] Writer invariant definition. Available at: https://www.encyclo.co.uk/meaning-of-Writer_invariant (accessed 2021, Mar)

[5] A. Leonova, I. Leonova Opredelenie avtorstva tekstov na osnove podkhoda n-gramm [Determination of the n-gram-based text authorship] Available at: https://science-engineering.ru/ru/article/view?id=1205 (accessed 2021, Mar)

[6] T. Joachims Text Categorization With Support Vector Machines: Learning With Many Relevant Features, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 1998, vol. 1398, pp. 137-142.

[7] R. Zheng, J. Li, Z. Huang A framework for authorship analysis of online messages: Writing-style features and techniques, *Journal of the American Society for Information Science and Technology*, 2006, vol. 57, no 3, pp. 378-393.

[8] Deep Learning based Authorship Identification. Available at: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174 /reports/2760185.pdf (accessed 2021, Apr)

[9] Wallace: Author Detection via Recurrent Neural Networks. Available at: https://cs224d.stanford.edu/reports/YaoLeon.pdf (accessed 2021, Apr)

[10] Stylometry for E-mail Author Identification and Authentication. Available at: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.547.438 &rep=rep1&type=pdf (accessed 2021, May)

[11] Topic or Style? Exploring the Most Useful Features for Authorship Attribution. Available at: https://www.aclweb.org/anthology/C18-1029/ (accessed 2021, May)

[12] Authorship Identification for Literary Book Recommendations. Available at: https://www.aclweb.org/anthology/C18-1033.pdf (accessed 2021, May)

Динамический выбор признаков в задаче идентификации автора

Парамонов А.И., Труханович И.А., Кунцевич В.С.

Работа посвящена проблемам выделения признаков в методе идентификации авторства в контексте определения авторского инварианта. Описана проблема идентификации автора текста и существующие подходы. В этой статье представлены модели, методы и результаты экспериментов по реализации динамического метода определения признаков с помощью генетического алгоритма.

Received 31.05.2021

About of the metric homogeneity of texts in Slavic languages

Z.D. Usmanov
A.Juraev Institute of Mathematics,
the National Academy of Sciences of Tajikistan
Dushanbe, Tajikistan
zafar-usmanov@rambler.ru

A.A. Kosimov
Tajik Technical University
named after acad. M.S.Osimi
Dushanbe, Tajikistan
abdunabi_kbtut@mail.ru

***Abstract*—In the studies of R. Gray and K. Atkinson [1] by the statistical analysis of related words, W. Chang, C. Cathcart, D. Hall and A. Garrett [2] using statistical modeling and A. S. Kasyan and A. V. Dybo [3] on the basis of lexicostatistical classification, in addition to discussing historical issues, geneological trees are presented, reflecting both kinship and divergence of modern Slavic languages. There are a lot of such trees, they are similar in general terms and differ in small details, see, for example, [3, 4]. The area of the formerly common language is now divided into three groups - the eastern one, consisting of the Belarusian, Russian and Ukrainian languages, the western one from the Czech, Slovak, Polish, Kashubian and Lusatian languages, and the southern one, consisting of the Bulgarian, Macedonian, Serbo-Croatian and Slovenian languages. Using the example of a randomly generated model collection of 26 texts in 13 languages (2 works from each language), the article establishes the applicability of the γ-classifier for automatic recognition of the belonging of texts to a particular group of Slavic languages based on the frequency of a set of Latin characters that is universal for all languages. The mathematical model of the γ-classifier is presented in the form of a triad composed of a digital portrait (DP) of the text - the distribution of the frequency of Latin symbolic unigrams in the text; formulas for calculating the distances between DP texts and a machine learning algorithm that implements the hypothesis of "homogeneity" of works from one language group and "heterogeneity" of works belonging to different groups of languages. The tuning of the algorithm using a table of paired distances between all products of the model collection was carried out by selecting the optimal value of the real parameter γ, which minimizes the number of errors in violation of tho "homogeneity" hypothesis. The e-classifier trained on the texts of the model collection showed 86% accuracy in recognizing the language groups of the works. To test the classifier, 3 additional random texts were selected, one text each for three different groups of Slavic languages. By the method of the nearest (in terms of distance) neighbor, all new texts confirmed their homogeneity with the corresponding pairs of monolingual works, thereby also homogeneity with the corresponding group of Slavic languages.**

***Keywords*—text, language, Slavs, alphabet, universal set of Latin characters, frequency, unigrams, digital portrait of a text, classifier, learning, recognition, language groups, performance assessment, testing the classifier.**

I. INTRODUCTION

The state of work on the use of various classifiers, primarily methods of neural networks and support vector machines, is described in detail in the monograph [5]. In this work, using the example of a randomly generated model collection of 26 works in 13 Slavic languages (2 works from each language), two problems are solved:

- *by choosing a real parameter γ, adjust the socalled γ-classifier, if possible, for error-free recognition of the of texts corresponding to one of the three groups of languages;*
- *for three additional randomly selected works belonging to different groups, check the correctness of the configured classifier.*

The solution of problems is based of the use of a γ-classifier - a mathematical triad, the first component of which is a digital portrait (DP) of the text - the distribution of the frequency of alphabetic unigrams in the text; the second component is a formula for calculating the distances between the text DP and the third is a machine learning algorithm that implements the hypothesis of "homogeneity" of works belonging to one language group and "heterogeneity" of works belonging to different groups of languages. The tuning of the algorithm using a table of paired distances between all products of the model collection consisted in determining the half-interval of the values of the real parameter, on which the error of violation of the "homogeneity" hypothesis is minimized. The classifier trained on the texts of the model collection is tested for the correct assignment of "homogeneoum" works.

Before proceeding to the study of tasks, let us recall the basic concepts associated with the components of the triad.

II. MODEL COLLECTION OF TEXTS C

Model collection of texts C, collected at random, it represents three groups of Slavic languages, with two works from each language. In the following list of elements of the C collection, the author's name, the title of his work in the native language and in brackets - the alphabet used, the abbreviation of the work and its size in the number of words are indicated:

a) in the Eastern Slavic group

in Belarusian:

L. Stanislav (Be): Салярыс, part 1" (cyr., be1, 8497 *words*);
S. Davidovich "Дзед-кiёк" (cyr., be2, 1935 *words*);

in Russian:

M.A. Sholokhov "Судьба человека" (cy., ru1, 10891 *words*);
F.A. Abramov "Алька" (cyr., ru2, 15668 *words*);

***in Ukrainian**:*

V.L. Kashin (Uk):"Biuletin Radzëznë Kaszëbsczégò Jãzëka" (cyr., uk1, 23771 *words*);

M. Tsiba "Акванавти, або Золота жила" (cyr., k2, 20150 *words*);

b) in the Western Slavic group

***in Polish**:*

R.M.Wegner "Jeszcze może załopotać, part 1" (lat., pl1, 10601 *words*);

R.M.Wegner "Jeszcze może załopotać, part 2" (lat., pl2, 9670 *words*);

in Czech:

S.Lem "K Mrakům Magellanovým" (lat., cs1, 17552 *rowds*);

B.S.R.Jordan (Cs): "Bouře přichází" (lat., cs2, 17439 *words*);

in Slovak:

I.A.Jefremov "Na hranici Oekumeny" (lat., sv1, 13534 *words*);

t.Jesenský "DemokraJi" (lat., sv2, 17113 *words*);

in Kashubian:

D.Pioch (Ks) "Biuletin Radzëznë Kaszëbszzégò Jãcëka" (lao., ks1, 12070 *wtrds*);

E.Breza "Prymas z Kaszub" (lat., ks2, 16871 *words*);

c) in the South Slavic group:

in Bulgarian:

N. Rainov "Неволя и богатство" (cyr., bo1, 2565 *words*);

B. Jim "Фурията на принцепса, chapter 1" (cyr., bo2, 2491 *words*);

in Bosnian:

I. Asimo "Немезис" (cyr., bs1, 20035);

. Waynes "Мјесечев мољац" (cyr., bs2, 10443 *words*);

in Serbian:

A. Clarke "Напеви далеке Земље" (cyr., se1, 11129 *words*);

R.L. Stevenson "Црна стрела" (cyr., se2, 15028 *words*);

in Slovenian:

M. Htdnik "Kakor Karuagina" (lat., sl1, 14626 *words*);

I. Karpiveo "Josip Vidmar v ceh svojih sodobnikov" (lot., sl2, 16985 *words*);

in Macedonian:

W. Tocinowsi "Кочо Рацин - наша творечка и етичка мерка" (cyr., mk1, 9047 *wrds*);

G. Prlichev "Kakor Kartagina" (cdr., mk2, 9478 *worys*);

in Croatian:

I.M. Andrić "Pročctani Pisii (Eseji i prikazi)" (lat., xr1, 26221 *words*);

M. Lovrak "Vlak U Snijegu" (lat., xr2, 10522 *words*).

III. Digital portrait of works

We use letter unigrams as elements of the quantitative image of works. Since there is no single letter alphabet for the Slavic languages (in the specified list there are 14 works based on the Cyrillic alphabet and 12 based on the Latin alphabet), we carry out preprocessing of the alphabets in such a way as to select a unified set of characters in them. Among the 14 analogues of the Cyrillic alphabets, 26 letters were in common: - "а, б, в, г, д, е, ж, з, и, й, к, л, м, н, о, п, р, с, т, у, ф, х, ч, ш, ю, я"; meanwhile, for 12 analogs of the Latin alphabet - also 26 letters, but already the following "a, b, c, d, e, f, g, h, i, s, k, l, m, n, a, p, q, r, i, t, u, v, w, x, y, z ". From these two alphabets, an artificial alphabet common for all texts was formed of 22 characters "a, b, c, d, e, f, g, i, j, k, l, m, n, o, p, r, t, u, t, y, z ", taking into account characters similar in spelling and sound.

Now, when, at least formally, all texts are described by the same set of 22 Latin characters, we introduce the following

Definition 1. *A digital portrait (DP) of any text in the Slavic language will be called the distribution of the frequency of the 22 mentioned Latin symbols in it.*

The DP of the text T is written in tabular form:

$$N : 12\dots22$$
$$P : p_1 p_2 \dots p_{22},$$

in which the first line is the numbers of characters in alphabetical order, and the second is the relative frequencies of occurrence of characters in the text T, and $\sum_{k=1}^{22} p_k = 1$.

The digital portrait is also presented as a discrete function

IV. Hypothesis H of "homogeneyty" of products

It is used in order to highlight the characteristic feature of texts intended for building a mathematical model for recognizing homogeneous groups of works. We formulate it as follows.

HYPOTHESIS H. *Any pair of works from the same group of Slavic languages is "homogeneous", but from different groups "not homogeneous".*

Speaking about the "homogeneity" of works (texts), we mean their similarity, similarity, uniformity, kinship, etc.

V. Mathematical model of the H-hypothesis

Let γ - be some positive number.

Definition 3. *Texts T_1, T_2 are called γ -homogeneous (belonging to the same group of Slavic languages) if*

, $$p(T_1, T_2) \leq \gamma \qquad (4)$$

and γ-heterogeneous (belonging to different groups of Slavic languages), if

. $$p(T_1, T_2)\gamma \qquad (5)$$

Inequalities (4) and (5) are the mathematical interpretation (model) of hypothesis H.

Definition 4. *A **γ-classifier** is a decision-making algorithm that depends on one real parameter γ for assigning a pair of texts T_1 and T_2 to one or two different groups of Slavic languages.*

Obviously, the homogeneity or heterogeneity of any pair of texts depends on the value of γ, and hence the degree of feasibility of the hypothesis. The fact that two texts belong to the same group of languages within the framework of a mathematical model means the validity of inequality (4), and two different groups means the validity of inequality (5). Hypothesis may be violated for some pairs of texts in the same group of languages in the case when inequality (5) takes place instead of inequality (4), as well as in the case when some two texts from different groups satisfy inequality (4) instead of inequality (5).

Let $\tau = \tau(\gamma -$ be the total number of violations of hypothesis H simultaneously in two case: non- fulfillment of the "homogeneity" inequality in the case of two texts belonging to the same group, and non-fulfillment of the "non-uniformity" inequality in the case of two texts belonging to different groups. Then, for a fixed γ, the hypothesis fulfillment index will be determined by the value π given by the formula

$$\pi = 1 - \tau(\gamma)/L \qquad (6)$$

where L - is the number of mutual distances between all pairs of texts from collection **C** (in our case, $L = C_{26}^2 = 325$. It follows from this formula that π can take values from the segment [0, 1], with π = 0 if, and π = 1 if τ = 0. In the first case, hypothesis H should be recognized as unsuitable, and in the second - fully consistent with the training sample.

Due to the fact that the efficiency of the classifier depends on the value of the parameter γ, it is of interest to find such a value at which π takes the maximum value. *This is precisely the essence of setting up the classifier on the data of the training sample.* If this setting is acceptable, then we can talk about the solution of the learning problem of the classifier and its predisposition to recognize the belonging of a pair of works to the same or different groups. The algorithm for setting the classifier is given in [6].

Texts		EastSlavic subgroup						WesternSlavic subgroup								SouthSlavic subgroup											
		be1	be2	ru1	ru2	uk1	uk2	pl1	pl2	cs1	cs2	sv1	sv2	ks1	ks2	bo1	bo2	bs1	bs2	se1	se2	sl1	sl2	mk1	mk2	xr1	xr2
East.	be1																										
	be2	0.13																									
	ru1	0.36	0.45																								
	ru2	0.27	0.35	0.09																							
	uk1	0.39	0.51	0.17	0.25																						
	uk2	0.36	0.47	0.13	0.21	0.04																					
West.	pl1	0.36	0.39	0.29	0.27	0.26	0.24																				
	pl2	0.33	0.36	0.28	0.26	0.28	0.25	0.03																			
	cs1	0.40	0.43	0.15	0.14	0.24	0.21	0.25	0.27																		
	cs2	0.34	0.37	0.13	0.12	0.27	0.24	0.21	0.23	0.06																	
	sv1	0.30	0.33	0.15	0.12	0.28	0.25	0.22	0.22	0.11	0.07																
	sv2	0.29	0.32	0.14	0.07	0.30	0.26	0.24	0.23	0.13	0.09	0.05															
	ks1	0.37	0.40	0.31	0.29	0.30	0.26	0.11	0.09	0.25	0.23	0.20	0.22														
	ks2	0.37	0.40	0.25	0.23	0.28	0.24	0.04	0.04	0.25	0.22	0.18	0.20	0.09													
South.	bo1	0.20	0.27	0.28	0.22	0.36	0.33	0.35	0.34	0.35	0.31	0.24	0.23	0.34	0.31												
	bo2	0.20	0.29	0.23	0.17	0.32	0.28	0.31	0.30	0.30	0.26	0.18	0.18	0.30	0.26	0.13											
	bs1	0.22	0.28	0.30	0.24	0.37	0.34	0.37	0.36	0.37	0.33	0.26	0.25	0.39	0.33	0.09	0.11										
	bs2	0.27	0.34	0.25	0.19	0.32	0.28	0.36	0.35	0.32	0.28	0.21	0.20	0.37	0.32	0.11	0.09	0.10									
	se1	0.23	0.30	0.27	0.21	0.35	0.31	0.36	0.35	0.34	0.30	0.22	0.22	0.38	0.32	0.09	0.09	0.05	0.08								
	se2	0.27	0.32	0.30	0.24	0.37	0.34	0.35	0.34	0.37	0.33	0.26	0.25	0.37	0.31	0.11	0.09	0.06	0.06	0.05							
	sl1	0.30	0.38	0.31	0.25	0.27	0.26	0.36	0.35	0.33	0.30	0.25	0.23	0.36	0.32	0.14	0.14	0.10	0.11	0.09	0.11						
	sl2	0.35	0.43	0.29	0.23	0.27	0.26	0.31	0.30	0.31	0.28	0.23	0.21	0.33	0.27	0.18	0.17	0.15	0.10	0.14	0.16	0.05					
	mk1	0.22	0.31	0.23	0.17	0.30	0.27	0.35	0.34	0.30	0.26	0.20	0.18	0.35	0.31	0.21	0.08	0.17	0.13	0.14	0.17	0.20	0.19				
	mk2	0.16	0.23	0.29	0.23	0.40	0.36	0.37	0.36	0.36	0.32	0.25	0.24	0.39	0.33	0.09	0.09	0.06	0.11	0.08	0.12	0.15	0.20	0.13			
	xr1	0.31	0.39	0.35	0.29	0.30	0.27	0.36	0.35	0.37	0.33	0.28	0.27	0.38	0.32	0.14	0.13	0.15	0.12	0.12	0.15	0.07	0.05	0.20	0.17		
	xr2	0.24	0.29	0.40	0.33	0.35	0.32	0.38	0.37	0.41	0.38	0.33	0.31	0.40	0.34	0.12	0.18	0.14	0.16	0.15	0.12	0.09	0.14	0.26	0.13	0.11	

Figure 1. Results on the example of the model collection ***C***

VI. Preliminary results on the example of the model collection C

Preliminary results on the example of the model collection C are given below by sequentially performing the following operations:

- calculation of digital portraits (frequency of letters of 22 common Latin characters) for all 26 works of the model collection ***C***;

- calculations by formulas (1), (2) and (3) 325 pair distances $p(T_1,\ T_2)$ between the products of the collection ***C*** (the calculation results are given in the figure 2)

- calculation using the γ-classifier tuning algorithm [6] of the optimal interval of γ values for which the value $\tau = \tau(\gamma$ of the total number of cases of violation of the hypothesis H reaches the minimum value and, therefore, the value π of the hypothesis fulfillment indicator H takes the maximum value.

Based on the data in Figure 1, the optimal half-interval of values is calculated

$$\gamma^{onm} \in [0.2142;\ 0.2160)$$

In accordance with Definition 3, this means that if the distance $p(T_1, T_2)$ between two texts does not exceed the value γ^{onm} 0.2160, then a pair of texts belongs to the same language group; if $p(T_1, T_2)$ exceeds 0.2160, then they belong to different languages.

The minimum number of violations turned out to be equal to τ = 45. In figure 1, the violation cells of the hypothesis (4) “homogeneity” are marked with a weak gray color, and the hypothesis (5) “heterogeneity” in gray.

Now it remains to calculate the efficiency of the classifier using the formula (6):

$$\pi = 1 - \tau(\gamma^{onm})/L = 0.86$$

VII. Testing the classifier

After, due to the choice of the optimal value of γ, the classifier was adjusted and the algorithm was worked out, which in 86 cases out of 100 correctly correlated the elements of the model collection to the corresponding group of Slavic languages, a natural question arises, what will be the results of the layout of other Slavic texts that are not included to the collection, for the same three language groups.

For testing the classifier, 3 texts were randomly selected:

in Ukrainian **(Uk) -** V.m. Berezhnoy “HoMo Novus” (cyr., Text_Uk, 5768 *words*);

in polish **(Pl) -** A. Szklarski “Tomek wśród łowców głów” (lat.,Text_Pl, 13635 *words*);

in Bulgarian **(Bo) -** A. Karaliychev “” (cyr., Text_Bo, 2436 *words*).

For each work, just as it was done for all texts in the model collection, the DP was built on the basis of the single set of 22 Latin characters. After that, using formula (3), the distances to all 26 elements of the model collection are calculated. The results are shown in the table.

Figure 2 shows distances between the texts of collection ***C*** and three randomly selected works.

Texts		Text _Uk	Text _Pl	Text _Bo
EastSlavic subgroup	be1	0.3421	0.3432	0.2031
	be2	0.4490	0.3742	0.2926
	ru1	0.1034	0.2699	0.2131
	ru2	0.1896	0.2517	0.1515
	uk1	0.0714	0.2398	0.3297
	uk2	**0.0511**	0.2190	0.2912
WtseernSlavic subgroup	pl1	0.1612	0.1916	0.2013
	pl2	0.1791	0.2030	0.1836
	cs1	0.1856	0.2070	0.2844
	cs2	0.2125	0.1745	0.2445
	sv1	0.2238	0.2010	0.2090
	sv2	0.2391	0.2162	0.1619
	ks1	0.2347	0.1271	0.3233
	ks2	0.2158	**0.0862**	0.2946
SouthSsavic lubgroup	bo1	0.3014	0.3389	0.1064
	bo2	0.2578	0.2901	**0.0510**
	bs1	0.3165	0.3521	0.1331
	bs2	0.2560	0.3386	0.1035
	se1	0.2918	0.3407	0.1115
	se2	0.3129	0.3303	0.1086
	sl1	0.2192	0.3418	0.1403
	sl2	0.2049	0.2971	0.1822
	mk1	0.2458	0.3232	0.1206
	mk2	0.3350	0.3500	0.0936
	xr1	0.2441	0.3419	0.1533
	xr2	0.2921	0.3661	0.2013

Figure 2. Distances between texts of collection ***C***

In table cells, as the intersection of columns and rows, the values of the distances between the texts are given. In the first three columns, the nearest neighbors of the texts Text_Uk, Text_Pl and Text_Bo are uk2, ks2 and bo2, respectively, at distances of 0.0511, 0.0862, end 0.0510, respectively (marked in gray in the table). The result obtained shows that, according to the nearest neighbor method, three randomly selected works are distributed exactly according to the language groups to which they themselves belong.

VIII. Conclusion

So, the γ-classifier with a fixed value on random samples of texts with digital portraits based on the frequency of 22 Latin characters confirmed 86% statistical ability to recognize groups of works in Slavic languages. In turn, the nearest neighbor method showed the possibility of an error-free distribution of additional Slavic works in the eastern, western and southern groups of Slavic languages.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

К вопросу о метрической однородности текстов на славянских языках

З.Д. Усманов, А.А. Косимов

Аннотация – В исследованиях Р.Грея и К.Аткинсона [1] посредством статистического анализа родственных слов, У.Чанга, Ч.Кэткарта, Д.Холла и А.Гарретта [2] с помощью статистического моделирования и А.С.Касьяна и А.В.Дыбо [3] на основе лексикостатистической классификации помимо обсуждения исторических вопросов представлены генеологические деревья, отражающие как родство, так и дивергенцию современных славянских языков. Таких деревьев достаточно много, они сходны в общих чертах и различны в небольших деталях, см. например, [3, 4]. Ареал прежде единого языка ныне разделился на три группы – восточную в составе белорусского, русского и украинского языков, западную - из чешского, словацкого, польского, кашубского и лужицких языков и южную, состоящую из болгарского, македонского, сербо-хорватского и словенского языков. В статье на примере случайно сформированной модельной коллекции из 26 текстов на 13 языках (по 2 произведения от каждого языка) устанавливается применимость γ-классификатора для автоматического распознавания принадлежности текстов той или иной группе славянских языков на основе частотности универсального для все языков набора латинских символов. Математическая модель -классификатора представляется в виде триады, составленной из цифрового портрета (ЦП) текста - распределения в тексте частотности латинских символьных униграмм; формулы для вычисления расстояний между ЦП текстами и алгоритма машинного обучения, реализующего гипотезу "однородности" произведений из одной группы языков и "неоднородности" произведений, принадлежащих разным группам языков. Настройка алгоритма, использующего таблицу парных расстояний между всеми произведениями модельной коллекции, осуществлялась путем подбора оптимального значения вещественного параметра γ, минимизирующего число ошибок нарушения гипотезы "однородности". Обученный на текстах модельной коллекции γ-классификатор показал 86%- ю точность в распознавании языков произведений. Для тестирования классификатора были выбраны 3 дополнительных случайных текста, по одному тексту для трёх разных групп славянских языков. Методом ближайшего (по расстоянию) соседа все новые тексты подтвердили свою однородность с соответствующими парами одноязычных произведений, тем самым и однородность с соответствующей группой славянских языков.

Ключевые слова – текст, язык, славяне, алфавит, универсальный набор латинских символов, частотность, униграмм, цифровой портрет текста, классификатор, обучение, распознавания, группы языков, оценка эффективности, тестирование классификатора.

Received 06.07.2021

AUTHOR INDEX

АВТОРСКИЙ УКАЗАТЕЛЬ

Научное издание

Открытые семантические технологии проектирования интеллектуальных систем

Open Semantic Technologies for Intelligent Systems

Сборник научных трудов

Основан в 2017 году

Выпуск 5

В авторской редакции

Ответственный за выпуск *В. В. Голенков*
Компьютерная вёрстка *Н. В. Гракова*

Подписано в печать 10.09.2021. Формат 60*84 1/8.
Бумага офсетная. Гарнитура «Таймс». Ризография.
Усл. печ. л. 38,36. Уч.-изд. л. 33,11.
Тираж 140 экз. Заказ 180.

Полиграфическое исполнение
УП «Бестпринт». Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий №1/160 от 27.01.2014.
Ул. Филатова, д. 9, к. 1, 220026, г. Минск

11th international scientific and technical conference «Open Semantic Technologies for Intelligent Systems»

OSTIS-2022

Open Semantic Technologies for Intelligent Systems

April 14-16, 2022 Minsk. Republic of Belarus

C A L L F O R P A P E R S

We invite you to take part in XII International Scientific and Technical Conference "Open Semantic Technologies for Intelligent Systems" (OSTIS-2022), which will focus on areas of use of the semantic technologies.

Conference will take place from ***April, 14th to April, 16th, 2022*** at the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus.

Conference proceedings language: English

Working languages of the conference: Russian, Belarusian, English

MAIN ORGANIZERS OF THE CONFERENCE

- Ministry of Education
- Ministry of Communications and Informatization
- State Institution "Administration of High Technologies Park" (Republic of Belarus)
- Educational-scientific association in the direction of "Artificial Intelligence" (ESA-AI)
- Belarusian State University of Informatics and Radioelectronics (BSUIR)
- Brest State Technical University (BrSTU)
- The State Scientific Institution «The United Institute of Informatics Problems of the National Academy of Sciences of Belarus» (UIIP NASB)
- Russian Association of Artificial Intelligence (RAAI)
- Belarusian Public Association of Artificial Intelligence Specialists (BPA of Artificial Intelligence Specialists)

CONFERENCE TOPICS:

- *Underlying principles of semantics-based knowledge representation, and their unification.*
  Types of knowledge and peculiarities of the semantics-based representation of various knowledge and metaknowledge types.
  Links between knowledge; relations, that are defined on the knowledge.
  Semantic structure of a global knowledge base, that integrates various accumulated knowledge.
- *Parallel-oriented programming languages for processing of the semantics-based representation of knowledge bases.*
- *Models for problem solving, that are based on knowledge processing, which occurs directly at the semantics-based representation level of knowledge being processed. Semantic models of information retrieval, knowledge integration, correctness and quality analysis of knowledge bases, garbage collection, knowledge base optimization, deductive and inductive inference in knowledge bases, plausible reasoning, pattern recognition, intelligent control. Integration of various models for problem solving*
- *Semantic models of environment information perception and its translation into the knowledge base.*
- *Semantic models of multimodal user interfaces of intelligent systems, based on the semantic representation of knowledge used by them, and unification of such models.*
- *Semantic models of natural language user interfaces of intelligent systems. The structure of semantic representation of linguistic knowledge bases, which describe natural languages and facilitate solution of natural language text and speech interpretation problems, and of natural language texts and speech messages synthesis, that are semantically equal to certain knowledge base fragments.*
- *Integrated logic-semantic models of intelligent systems, based on semantic knowledge representation, and their unification*
- *Various technical platforms and implementation variants of unified logic-semantic models of intelligent systems, based on semantic knowledge representation*
- *Models and means, that are based on the semantic representation of knowledge and that are oriented to the design of various typical components of intelligent systems (knowledge bases, programs, problem solvers, user interfaces).*
- *Models and means, that are based on semantic representation of knowledge and that are oriented to the complex design of various classes of intelligent systems (intelligent reference systems, intelligent learning systems, intelligent control systems, intelligent robotics systems, intelligent systems for design support etc.)*
- *Applied intelligent systems, that are based on the semantic representation of knowledge used by them*

CONFERENCE GOALS AND FORMAT

The goal of the conference is to discuss problems of creation of the **Open Complex Semantic Technology for Hybrid Intelligent Systems Design**. This determines the Conference format, which involves wide discussion of various questions of creating of such technology and poster sessions.

During the **poster sessions** every participant of the conference will have an opportunity to demonstrate his results. Conference format assumes exact start time of each report, and exact time of its exhibition presentation.

One of the major objectives of the conference is to attract not only scientists and postgraduate students, but also students who are interested in artificial intelligence, as well as commercial organizations willing to collaborate with research groups working on the development of modern technologies for intelligent systems design.

PARTICIPATION TERMS AND CONDITIONS

All those interested in artificial intelligence problems, as well as commercial organizations willing to collaborate with research groups working on the development of modern technologies for intelligent systems design are invited to take part in the Conference.

To participate in the OSTIS-2022 conference, it is necessary to register in the CMT system before April 14, 2022, find conference page, and from there:

- submit a **participation form** for the OSTIS-2022 conference. Each participation form field is required, including indication of the reporter. By filling in the registration form, you agree that your personal data will be processed by the Organizing Committee of the Conference, and that the paper and information about the authors will be published in printed and electronic format. Participation form should contain information on all of the authors. If author(s) are participating with a report, participation form should have their **color photo(s)** attached (they are needed for the Conference Program);
- upload an **article** for publication in the OSTIS-2022 Conference Proceedings. Papers should be formatted according to the provided template (see https://proc.ostis.net/for-authors/). Four full pages is a minimum size of a paper.
- send the signed **scan of the letter of consent**

If a report is submitted to participate in one of the contests, this intention should be clearly indicated in the participation form.

The selection of papers for publication in the Conference Proceedings and participation in the Conference is performed by a number of reviewers from among the members of the Conference Program Committee.

Incompliant applications and papers will be rejected.

Authors, whose articles were included in the Conference Program, will receive the invitations for participating in the Conference before April 1st, 2022.

Conference participation does not require any fees.

PAPERS SUBMISSION PROCEDURE

Papers (only on topics mentioned above) should be submitted ready for publication (http://proc.ostis.net/eng/main.html -> For authors). The text should be logically complete and contain new scientific and practical results. Each author is allowed to submit two reports maximum.

After the article was submitted, it is sent for review. Review results will become available to the paper author(s) on the CMT website before April 1th.

The Organizing Committee reserves the right to reject any paper, if it does not meet the formatting requirements and the Conference topics, as well as if there was no participation form submitted for the paper.

YOUNG SCIENTIST REPORTS CONTEST

Authors of the report submitted to the contest may include scientists with scientific degrees, but the report should be made by those without a degree and under 35 years old.

To take part in the young scientists report contest, it is necessary to:

1) fill in the participation form, where your participation in the contest is clearly indicated;
2) write an article and upload it to the CMT website;
3) fill in, sign, scan and send letter of consent via the email.
4) make a report at the conference (in person);

YOUNG SCIENTIST PROJECTS CONTEST

Projects of applied intelligent systems and systems aimed at supporting the design of intelligent systems are allowed to take part in the contest; they have to be presented by a scientist without a degree and under 35 years old.

To take part in the young scientist projects contest, it is necessary to:

1) fill in the participation form, where your participation in the contest is clearly indicated;
2) write an article and upload it to the CMT website;
3) make a report at the conference (in person);
4) make an exhibition presentation of the software

STUDENT INTELLIGENT SYSTEM PROJECTS CONTEST

To participate in the contest, a project must meet the following criteria: (a) it was developed by students and/or undergraduates of the higher education institutions, and (b) project consultants and advisors must hold a scientific degree and title. To participate in this contest, it is necessary to:

1) familiarize yourself with contest's terms and conditions (http://conf.ostis.net);
2) fill in the participation form for the contest (http://conf.ostis.net);
3) prepare a summary of the project (http://conf.ostis.net).
4) submit the participation form and project summary to the student projects' email address: ostis.stud@gmail.com.

CONFERENCE PROCEEDINGS PUBLICATION

The Conference Organizing Committee plans to publish the papers selected by the Program Committee based on the results of their review, in the Conference Proceedings, on the official Conference website http://conf.ostis.net and on the Conference Proceedings website http://proc.ostis.net**.**

Upon successful review author sends a letter of consent to the Organizational Committee. Author therefore agrees that his paper can be made freely available in electronic form at other resources at the Editorial Board's discretion.

Since 2020, the OSTIS Collection of Scientific Papers has been included in the List of Scientific Publications of the Republic of Belarus for publishing the results of dissertation research (List of the Higher Attestation Commission of the Republic of Belarus) in the technical field of science (informatics, computer technology and management).

In addition, following the results of the conference, it is planned to publish the Collection of selected materials of the OSTIS conference in the series "Communications in Computer and Information Science" (CCIS) published by Springer. Detailed information about this can be found on the conference website (http://conf.ostis.net).

KEY DATES OF THE CONFERENCE

January 11th, 2022	paper submission opens
February 14th, 2022	paper submission deadline
March 21th, 2022	paper review deadline
April 1st, 2022	final decision on paper publication; sending out invitations and notifications on inclusion of a paper in the OSTIS-2022 Conference Proceedings
April 7th, 2022	Draft Conference Program publication on the conference website http://conf.ostis.net
April 12th, 2022	Conference Proceedings and Conference program publication on the conference website http://proc.ostis.net
April 14th, 2022	Participant registration and OSTIS-2022 conference opening
April 14th to 16th, 2022	OSTIS-2022 Conference
May 1st, 2022	photoreport and conference report publication on the conference website:

	http://conf.ostis.net
May 14th, 2022	conference proceedings will be uploaded to the Russian Science Citation Index database

CONFERENCE PROGRAM FORMATION

Conference program is formed by the Program Commitee according to the paper review results; author(s)' confirmation of participation is required as well.

CONTACTS

All the necessary information about the forthcoming and previous OSTIS Conferences can be found on the conference website http://conf.ostis.net and http://proc.ostis.net.

For questions regarding conference participation and dispute resolution please contact: ostisconf@gmail.com.

Methodological and advisory support to the conference participants shall be provided through the conference e-mail only.

The conference venue is the 5th academic building of the Belarusian State University of Informatics and Radioelectronics (39, Platonov str., Minsk, Republic of Belarus).

OSTIS-2022

XII международная научно-техническая конференция «Открытые семантические технологии проектирования интеллектуальных систем»

Open Semantic Technologies for Intelligent Systems

14 – 16 апреля 2022 г. Минск. Республика Беларусь

И Н Ф О Р М А Ц И О Н Н О Е   П И С Ь М О

Приглашаем принять участие в XII Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2022), которая будет посвящена вопросам области применения семантических технологий.

Конференция пройдет в период с ***14 по 16 апреля 2022*** года в Белорусском государственном университете информатики и радиоэлектроники, г. Минск, Республика Беларусь.

Язык статей сборника конференции: английский

Рабочие языки конференции: русский, белорусский, английский.

ОСНОВНЫЕ ОРГАНИЗАТОРЫ КОНФЕРЕНЦИИ

- Министерство образования Республики Беларусь
- Министерство связи и информатизации Республики Беларусь
- Государственное учреждение «Администрация Парка высоких технологий» (Республика Беларусь)
- Белорусский государственный университет информатики и радиоэлектроники (БГУИР)
- Брестский государственный технический университет (БрГТУ)
- Государственное научное учреждение «Объединенный институт проблем информатики Национальной академии наук Беларуси» (ОИПИ НАН Беларуси)
- Белорусское общественное объединение специалистов в области искусственного интеллекта (БОИИ)
- Российская ассоциация искусственного интеллекта (РАИИ)
- Учебно-научное объединение по направлению «Искусственный интеллект» (УНО-ИИ)

НАПРАВЛЕНИЯ РАБОТЫ КОНФЕРЕНЦИИ:

- *Принципы, лежащие в основе семантического представления знаний, и их унификация.*
  Типология знаний и особенности семантического представления различного вида знаний и метазнаний.
  Связи между знаниями и отношения, заданные на множестве знаний.
  Семантическая структура глобальной базы знаний, интегрирующей различные накапливаемые знания
- *Языки программирования, ориентированные на параллельную обработку семантического представления баз знаний*
- *Модели решения задач, в основе которых лежит обработка знаний, осуществляемая непосредственно на уровне семантического представления обрабатываемых знаний. Семантические модели информационного поиска, интеграции знаний, анализа корректности и качества баз знаний, сборки информационного мусора, оптимизации баз знаний, дедуктивного и индуктивного вывода в базах знаний, правдоподобных рассуждений, распознавания образов, интеллектуального управления. Интеграция различных моделей решения задач*
- *Семантические модели восприятия информации о внешней среде и отображения этой информации в базу знаний*
- *Семантические модели мультимодальных пользовательских интерфейсов интеллектуальных систем, в основе которых лежит семантическое представление используемых ими знаний, и унификация этих моделей*
- *Семантические модели естественно-языковых пользовательских интерфейсов интеллектуальных систем. Структура семантического представления лингвистических баз знаний, описывающих естественные языки и обеспечивающих решение задач понимания естественно-языковых текстов и речевых сообщений, а также задач синтеза естественно-языковых текстов и речевых сообщений, семантически эквивалентных заданным фрагментам баз знаний*
- *Интегрированные комплексные логико-семантические модели интеллектуальных систем, основанные на семантическом представлении знаний, и их унификация*
- *Различные технические платформы и варианты реализации интерпретаторов унифицированных логико-семантических моделей интеллектуальных систем, основанных на семантическом представлении знаний*
- *Средства и методы, основанные на семантическом представлении знаний и ориентированные на проектирование различных типовых компонентов интеллектуальных систем (баз знаний, программ, решателей задач, интерфейсов)*
- *Средства и методы, основанные на семантическом представлении знаний и ориентированные на комплексное проектирование различных классов интеллектуальных систем (интеллектуальных справочных систем, интеллектуальных обучающих систем, интеллектуальных систем управления, интеллектуальных*

робототехнических систем, интеллектуальных систем поддержки проектирования и др.)

- *Прикладные интеллектуальные системы, основанные на семантическом представлении используемых ими знаний*

ЦЕЛЬ И ФОРМАТ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ

Целью конференции является обсуждение проблем создания **открытой комплексной семантической технологии компонентного проектирования семантически совместимых гибридных интеллектуальных систем**. Этим определяется и формат её проведения, предполагающий широкое обсуждение различных вопросов создания указанной технологии и выставочные презентации докладов.

Выставочная презентация докладов даёт возможность каждому докладчику продемонстрировать результаты своей разработки на выставке. Формат проведения конференции предполагает точное время начала каждого доклада и точное время его выставочной презентации.

Важнейшей задачей конференции является привлечение к её работе не только учёных и аспирантов, но и студенческой молодежи, интересующейся проблемами искусственного интеллекта, а также коммерческих организаций, готовых сотрудничать с научными коллективами, работающими над интеллектуальными системами и созданием современных технологий и их проектированием.

УСЛОВИЯ УЧАСТИЯ В КОНФЕРЕНЦИИ

В конференции имеют право участвовать все те, кто интересуется проблемами искусственного интеллекта, а также коммерческие организации, готовые сотрудничать с научными коллективами, работающими над созданием современных технологий проектирования интеллектуальных систем.

Для участия в конференции OSTIS-2022 необходимо до 14 февраля 2022 года зарегистрироваться в системе СМТ, найти страницу конференции и на ней:

- подать **заявку** на конференцию OSTIS-2022. Каждое поле заявки обязательно для заполнения, в том числе указание того автора, кто будет представлять доклад. Заполняя регистрационную форму, Вы подтверждаете согласие на обработку Оргкомитетом конференции персональных данных, публикацию статей и информации об авторах в печатном и электронном виде. В заявке должна содержаться информация по каждому автору;
- загрузить **статью** для публикации в Сборнике научных трудов конференции OSTIS-2021. Статья на конференцию должна быть оформлена в соответствии с правилами оформления публикуемых материалов и занимать не менее 4 полностью заполненных страниц;
- загрузить **сканированный вариант письма о согласии** на публикацию и размещения передаваемых материалов в сети Интернет;
- загрузить **цветные фотографии** всех авторов статьи (это необходимо для публикации Программы конференции)

Если доклад представляется на конкурс докладов молодых учёных или на конкурс программных продуктов молодых учёных, это должно быть явно указано в заявке доклада.

Отбор статей для публикации в Сборнике и участия в работе конференции осуществляется рецензентами и редакционной коллегией сборника.

Заявки и статьи, оформленные без соблюдения предъявляемых требований, не рассматриваются.

До 1 апреля 2022 года, авторам статей, включённых в Программу конференции, направляются приглашения для участия в конференции.

Участие в конференции не предполагает организационного взноса.

ПОРЯДОК ПРЕДСТАВЛЕНИЯ НАУЧНЫХ СТАТЕЙ

Статьи (только по перечисленным выше направлениям) представляются в готовом для публикации виде (*http://proc.ostis.net* -> *Авторам*). Текст статьи должен быть логически законченным и содержать новые научные и практические результаты. От одного автора допускается не более двух статей.

После получения статьи, она отправляется на рецензирование и в срок до 1 апреля на сайте СМТ вы сможете ознакомиться с результатами рецензирования

Оргкомитет оставляет за собой право отказать в приеме статьи в случае, если статья не будет соответствовать требованиям оформления и тематике конференции, а также, если будет отсутствовать заявка доклада, соответствующая этой статье.

КОНКУРС ДОКЛАДОВ МОЛОДЫХ УЧЁНЫХ

Соавторами доклада, представляемого на конкурс докладов молодых учёных, могут быть учёные со степенями и званиями, но непосредственно представлять доклад должны авторы в возрасте до 35 лет, не имеющие степеней и званий.

Для того, чтобы принять участие в конкурсе научных докладов молодых учёных, необходимо:

1) заполнить заявку на участие в конференции, в которой чётко указать своё желание принять участие в данном конкурсе;
2) написать статью на конференцию и загрузить на сайте CMT;
3) заполнить, подписать, отсканировать и отправить по почте письмо о согласии;
4) лично представить доклад на конференции.

КОНКУРС ПРОЕКТОВ МОЛОДЫХ УЧЁНЫХ

Принимать участие в конкурсе проектов молодых учёных могут проекты прикладных интеллектуальных систем и систем, ориентированных на поддержку проектирования интеллектуальных систем, при этом представлять проект на конкурсе должен молодой учёный в возрасте до 35 лет, не имеющий учёной степени.

Для того, чтобы принять участие в конкурсе программных продуктов молодых учёных, необходимо:

1) заполнить заявку на участие в конференции), в которой чётко указать своё желание принять участие в данном конкурсе;
2) написать статью на конференцию и загрузить на сайте CMT;
3) лично представить доклад на конференции;
4) провести выставочную презентацию, разработанного программного продукта.

КОНКУРС СТУДЕНЧЕСКИХ ПРОЕКТОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

В конкурсе студенческих проектов могут принимать участие проекты, разработчиками которых являются студенты и магистранты высших учебных заведений, консультантами и руководителями проекта могут быть лица, имеющие научную степень и звание. Для того, чтобы принять участие в данном конкурсе, необходимо:

1) ознакомиться с положением о конкурсе студенческих проектов (http://conf.ostis.net);
2) заполнить заявку на участие в конкурсе студенческих проектов (http://conf.ostis.net);
3) подготовить описание проекта (http://conf.ostis.net).
4) выслать заявку на участие в конкурсе и описание проекта по электронному адресу конкурса студенческих проектов: ostis.stud@gmail.com.

ПУБЛИКАЦИЯ МАТЕРИАЛОВ КОНФЕРЕНЦИИ

Оргкомитет конференции предполагает публикацию статей, отобранных Программным комитетом по результатам их рецензирования, в Сборнике научных трудов OSTIS в печатном виде и на официальном сайте сборника http://proc.ostis.net в электронном виде**.**

По результатам рецензирования автор отправляет оргкомитету письмо о согласии, которое предусматривает дальнейшую возможность размещения статей, вошедших в сборник конференции, в открытом электронном доступе на иных ресурсах по усмотрению редакции сборника.

С 2020 года Сборник научных трудов OSTIS включен в Перечень научных изданий Республики Беларусь для опубликования результатов диссертационных исследований (Перечень ВАК РБ) по технической отрасли наук (информатика, вычислительная техника и управление).

Кроме того, по итогам конференции планируется издание Сборника избранных материалов конференции OSTIS в серии «Communications in Computer and Information Science» (CCIS) издательства Springer. Подробная информация об этом приведена на сайте конференции (http://conf.ostis.net).

КЛЮЧЕВЫЕ ДАТЫ КОНФЕРЕНЦИИ

11 января 2022г.	начало подачи материалов для участия в конференции
14 февраля 2022г.	срок получения материалов для участия в конференции Оргкомитетом
21 марта 2022г.	срок предоставления рецензий на статьи
1 апреля 2022г.	срок принятия решения о публикации присланных материалов и рассылки приглашений для участия в конференции и сообщение о включении статьи в Сборник материалов конференции OSTIS
7 апреля 2022г.	размещение на сайте конференции http://conf.ostis.net проекта программы конференции
12 апреля 2022г.	размещение на сайте конференции http://proc.ostis.net Сборника материалов и Программы конференции OSTIS-2022
14 апреля 2022г.	регистрация участников и открытие конференции OSTIS-2022
14-16 апреля 2022г.	работа конференции OSTIS-2022
1 мая 2022г.	публикация фоторепортажа и отчёта о проведённой конференции на сайте конференции: http://conf.ostis.net
14 мая 2022г.	загрузка материалов сборника конференции в РИНЦ

ФОРМИРОВАНИЕ ПРОГРАММЫ КОНФЕРЕНЦИИ

Программа конференции формируется Программным комитетом по результатам рецензирования, представленных статей, а также на основании подтверждения автора(-ов) статьи о прибытии на конференцию.

КОНТАКТНЫЕ ДАННЫЕ ОРГАНИЗАТОРОВ КОНФЕРЕНЦИИ OSTIS

Вся необходимая информация по предстоящей и предыдущих конференциях OSTIS находится на сайте конференции http://conf.ostis.net, а также на сайте материалов конференции http://proc.ostis.net.

По вопросам участия в конференции и решения спорных вопросов обращайтесь: ostisconf@gmail.com.

Методическая и консультативная помощь участникам конференции осуществляется только через электронную почту конференции.

Конференция проходит в Республике Беларусь, г. Минск.

Оргкомитет конференции находится на кафедре интеллектуальных информационных технологий Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники (БГУИР) – г. Минск, ул. Платонова, 39, 5-ый учебный корпус БГУИР.