detail in the lower part of the figure, performing the interpretation of methods within the OSTIS Ecosystem (shown by larger rectangles with the label “M”).

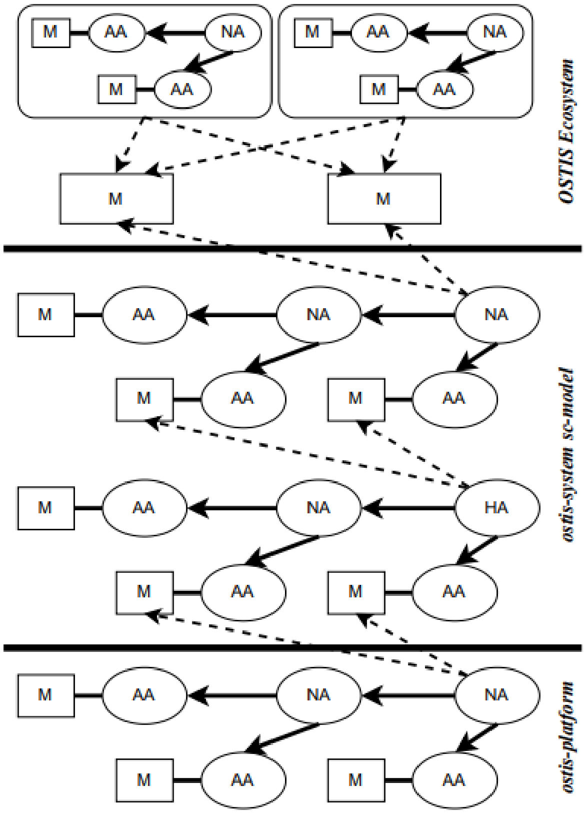

Figure 1. Figure. Hierarchy of sc-agents

V. Conclusion

The paper considers an approach to the organization of problem solving within a distributed team of intelligent computer systems that are part of the OSTIS Ecosystem (ostis-systems).

Further development of the presented principles of problem solving by distributed teams of ostis-systems involves:

- Development of formal criteria for assessing the feasibility or inexpediency of the formation of temporary individual ostis-systems;
- Development of the language and principles of messaging between ostis-systems that are part of the ostis-systems team that solves any task. Despite the fact that from a logical point of view, each ostissystem is treated as a sc-agent and the principles of their interaction remain the same, the implementation, for example, of the ability to respond to events in the knowledge base and make changes to this knowledge base for internal sc-agents and external ostis-systems will be different and requires clarification.

Acknowledgment

The authors would like to thank the scientific teams 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.

References

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. 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.
3. D. Shunkevich, “Hybrid problem solvers of intelligent computer systems of a new generation,” *Open semantic technologies for intelligent systems*, no. 6, pp. 119–144, 2022.
4. V. Golenkov, N. Guliakina, and D. Shunkevich, *Otkrytaja tehnologija ontologicheskogo proektirovanija, proizvodstva i jekspluatacii semanticheski sovmestimyh gibridnyh intellektual’nyh komp’juternyh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint [Bestprint], 2021.
5. D. Pospelov, *Situacionnoe upravlenie. Teorija i praktika [Situational management. Theory and practice]*. M.: Nauka, 1986.
6. P. Skobelev, “Situacionnoe upravlenie i mul’tiagentnye tehnologii: kollektivnyj poisk soglasovannyh reshenij v dialoge [Situational management and multi-agent technologies: collective search for agreed solutions in a dialogue],” *Ontologija proektirovanija [Ontology of designing]*, no. 2, pp. 26–48, 2013.
7. M. Wooldridge, *An Introduction to MultiAgent Systems — Second Edition*. Wiley, 2009.
8. V. Gorodetskii, V. Samoilov, and D. Trotskii, “Bazovaya ontologiya kollektivnogo povedeniya avtonomnykh agentov i ee rasshireniya [Basic ontology of autonomous agents collective behavior and its extension],” *Izvestiya RAN. Teoriya i sistemy upravleniya [Proceedings of the RAS. Theory and control systems]*, no. 5, pp. 102–121, 2015, (in Russian).
9. V. Tarasov, *Ot mnogoagentnykh sistem k intellektual’nym organizatsiyam [From multi-agent systems to intelligent organizations]*. M.: Editorial URSS, 2002, (in Russian).
10. A. N. Kopajgorodskij and L. V. Massel, “Fraktal’nyj podhod k proektirovaniju arhitektury informacionnyh sistem [Fractal Approach to Information Systems Architecture Design],” *Vestnik IrGTU [Bulletin of ISTU*, no. 6(46), pp. 8–12, 2010.

Принципы решения задач в распределенных коллективах интеллектуальных компьютерных систем нового поколения

Шункевич Д. В.

В работе рассмотрен подход к организации решения задач в рамках распределенного коллектива интеллектуальных компьютерных систем, входящих в состав Экосистемы OSTIS (ostis-систем). Рассмотрена классификация агентов в рамках такой системы, а также принципы их взаимодействия.

Received 13.03.2023

Versioning Model of Neural Network Problem-Solving Methods in Intelligent Systems

Mikhail Kovalev
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
michail.kovalev7@gmail.com

***Abstract*—In the article, the design process of neural network problem-solving methods in the knowledge bases of intelligent systems is considered. The versioning model of neural network problem-solving methods, described in a specialized language for representation of neural network problem-solving methods, is proposed.**

***Keywords*—problem-solving method, ontological approach, neuro-symbolic AI, artificial neural network**

I. Introduction

The modern development of all directions of *Artificial intelligence* is aimed at building *intelligent systems* that automate more and more complex human activities. The current state in the field of developing *intelligent systems of a new generation[1]* shows that such systems should provide:

- unification of representation and coherence of different knowledge types and problem-solving methods;
- integration and convergence of different problemsolving methods in a single knowledge base to ensure consistency in the semantics of that set of methods;
- representation and interpretation of as many classes of problem-solving methods (programs) as possible.

Integration of different problem-solving methods in a single knowledge base guarantees consistency of semantics of this set of methods. When solving problems using such methods, the system does not communicate with the external environment by transferring input and output data. Instead, a single knowledge base allows the system to track changes in input knowledge in real time using a wide range of methods, which provides the ability to introspect and explain the decisions made by the system.

A single *knowledge base for problem-solving methods and knowledge* used to solve them allows the system to reflect on the process of problem solving, to explain the reasons for its solutions, and to find mistakes there. The actively developing *class of problem-solving methods is artificial neural networks* (ANNs). This is conditioned, on the one hand, by the rapid development of the theoretical foundations of artificial neural networks and on the other hand, by the increasing computing power of the machines used to train them.

Impressive results have been obtained in problem solving with artificial neural networks [2]. Among the positive characteristics of ANNs are their ability to effectively solve problems in the absence of known regularities, as well as their ability to solve problems without necessarily developing problem-oriented *problemsolving methods*.

However, there are serious problems with neural network problem-solving methods:

- Heuristic nature of the design process of *neural network problem-solving methods*. The process of selecting ANNs architectures and their training parameters places high demands on the knowledge level of ANNs engineers
- Lack of explicit allocation of semantic connections between knowledge in the process of problem solving. They are highlighted implicitly, statistically, based on the data that was used for training. Lack of explicit allocation of meaning leads to the problem of the *"black box"* [3]. An entire field of Explainable AI has emerged, in which researchers attempt to explain the ANNs solutions [4], [5]

Formalization of *ANNs* in the *knowledge base* of the *intelligent system* together with *other problem-solving methods* allows negating the listed *ANNs* problems, since in such systems, the design problem of ANNs and the explanation problem of solutions for these ANNs are represented in one form for the whole knowledge base and can be solved using any of the represented *problemsolving method* from this *knowledge base*.

Frequently, the *ANN* is actively changed during the design and interpretation process (configuration of connections, number of layers, synapse weights, activation functions, etc.). To solve the problem of *ANNs* design, the system must be able to analyze the solutions of the same problem on different versions of the same *neural network problem-solving method* in order to evaluate the success of certain solutions in the design of this method, for example, the success of selection of activation functions, training sample, training algorithm, configuration of connections in layers, etc.

The purpose of this article is to develop an approachto versioning of *neural network problem-solving methods in the knowledge base of the* *intelligent system.*

II. Proposed aproach

In order to solve the above problems, the OSTIS Technology is proposed. Intelligent systems developed using the OSTIS Technology are called ostis-systems. Any ostis-system consists of a knowledge base, a problem solver, and a user interface.

The problem solver performes the processing of fragments of the knowledge base. At the operational level, processing means adding, searching, editing, and deleting sc-nodes and sc-connectors of the knowledge base. On the semantic level, such an operation is an *action performed in the memory of an action subject*, where, in the general case, the subject is an ostis-system and the knowledge base is its memory. An action is defined as the influence of one entity (or some set of entities) to another entity (or some set of other entities) according to some purpose.

Actions are performed according to the set problems. *A problem* is a formal specification of some action, sufficient to perform this action by some subject. Depending on a particular class of problems, it is possible to describe both the internal state of the intelligent system itself and the required state of the external environment [6].

For classes of problems, classes of methods for their solution are formulated. A *problem-solving method* is defined as a problem-solving program of the corresponding class, which can be either procedural or declarative. In turn, a *class of problem-solving methods* is defined as a set of all possible problem-solving methods having a common language for representing these methods. The method representation language allows describing the syntactic, denotational, and operational semantics of this method.

Approaches to **integration of ANNs with knowledge bases** in ostis-systems are considered in [7]. Input-output integration approaches have been tested and described in [8], [9]. Full *integration of ANNs with knowledge bases,* i.e., using neural network problem-solving methods by formalizing them in an ostis-system knowledge base, are described in [10], which describes the *Denotational and Operational semantics of the Language for representation of neural network problem-solving methods (Neuro-SCP).* The present work is a development of this language.

The *Language for representation of neural network problem-solving methods* allows representing and interpreting neural network methods in the ostis-system memory. This language is a sublanguage of the SCP *Language* [11]. Any method represented in the SCP *Language* is a fragment of the knowledge base, so the problem of *versioning of neural network problem-solving methods in the knowledge base of the intelligent system* is reduced to the problem of versioning any *knowledge base fragments.*

During the existence of the *intelligent system*, the state of fragments of its *knowledge base* (that is, the configuration of connections between signs in this knowledge base), as well as the neural network problem-solving method represented in the knowledge base, can change over time [12]. The need to account for the dynamics of knowledge changing over time in *knowledge bases of intelligent systems* is conditioned by the qualities inherent in intelligent systems, as well as the range of problems they solve. *Intelligent* systems must:

- maintain the relevance, adequacy, and accuracy of the knowledge stored in it at any point in time;
- remember the history of user and developer actions performed on *fragments* of the knowledge base in order to analyze them and support decision-making in other problems;
- allow performing reverse actions in case of abnormal situations;
- allow verifying the sources of unreliable knowledge and warn about inconsistencies in knowledge bases;
- adapt to the characteristics of its users and other *intelligent systems;*
- plan and initiate different kinds of problem solving.

Thus, to provide a higher level of intelligence of the system and support its life cycle, it is necessary to specify a set of methods and tools that allow solving these problems quickly and efficiently. One of such means for solving these problems is a ***Subsystem for versioning of knowledge base fragments***, embedded in any *intelligent system*, developed on the principles of the OSTIS *Technology* (i.e., in ostis-system), providing continuous versioning for various knowledge base fragments and analysis of their states.

A *knowledge base fragment* means a formal specification of any entity or concept sign represented in the *knowledge base* of a given system. That is, a *knowledge base fragment* is nothing but some semantic neighborhood or structure that includes knowledge about an object in *the subject domain of the knowledge base* of this system. The process of *versioning of a knowledge base fragment* implies a complete representation and description of its states, as well as providing capabilities and tools for processing and analyzing the states of this *knowledge base fragment.*

The state of the knowledge base fragment means the integration of the results of actions performed on this knowledge base fragment since its existence in the knowledge base (that is, since the initial state of this knowledge base fragment).

The *versioning of the knowledge base fragment* requires strict identification of all states from the beginning of the existence of this *knowledge base fragment*, that is, it requires the construction of the *bijective correspondence* between the *state of the knowledge base fragment* and its unique identifier in the whole knowledge base of the