[12] D. Bohm, *Wholeness and the Implicate Order.* London: Routledge, 2002.

[13] A. N. Gordei. Theory for automatic generation of knowledge architecture: Tapaz-2. Available at: http://tapaz.by (accessed 18.04.2021).

[14] A. Hardzei, M. Sviatoshchik, L. Bobyor, and S. Nikiforov, "Universal semantic markup and top-level ontology representation," in *Open Semantic Technologies for Intelligent Systems.* Springer International Publishing, 2022, pp. 192–221. [Online]. Available: https://doi.org/10.1007/978-3-031-15882-711

[15] M. A. Armstrong, Basic Topology. M.: Springer, 1983.

[16] L. Collatz, *Functional Analysis and Numerical Mathematics.* New York, San Francisco, London: Academic Press, 1966.

[17] A. Narin'yani, "Ne-faktory: kratkoe vvedenie [non-factors: a brief introduction]," *Novosti iskusstvennogo intellekta [AI news]*, no. 2, pp. 52–63, 2004.

[18] V. P. Ivashenko, "Ontologicheskaya model' prostranstvennovremennykh otnoshenii sobytii i yavlenii v protsessakh obrabotki znanii [ontological model of the space-time relations of the event and the phenomenon in the processes of knowledge processing," *Vestnik Brestskogo gosudarstvennogo tekhnicheskogo universiteta [Bulletin of Brest State Technical University]*, vol. 107, no. 5, pp. 13–17, 2017, (In Russ.).

[19] V. M. Kuz'mitskii, "Printsipy postroeniya grafodinamicheskogo parallel'nogo komp'yutera, orientirovannogo na reshenie zadach iskusstvennogo intellekta [principles of constructing a graphdynamic parallel computer focused on solving artificial intelligence problems]," dis. ... kand. tekhn. nauk : 05.13.11 [dis. ... Cand. of tech. sciences: 05.13.11], Minsk, 2000, 236 p.

[20] L. Kalinichenko, *Metody i sredstva integratsii neodnorodnykh baz dannykh [Methods and tools for integrating heterogeneous databases].* M.: Nauka, 1983, (In Russ.).

[21] V. Ivashenko, "Application of an integration platform for ontological model-based problem solving using an unified semantic knowledge representation," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 179–186, 2022.

[22] 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 : BSUIR, 2020, (In Russ.).

[23] A. Tarski, "On the calculus of relations," *J. Symbolic Logic*, vol. 6, 09 2014.

[24] W. Li and L. Qian, "Semantic approach to user answer verification in intelligent tutoring systems," in *Open Semantic Technologies for Intelligent Systems*, V. Golenkov, V. Krasnoproshin, V. Golovko, and D. Shunkevich, Eds. Springer International Publishing, 2022, pp. 242–266. [Online]. Available: https://doi.org/10.1007/ 978-3-031-15882-713

[25] T. Imielinski and W. Lipski, "The relational model of data and cylindric algebras," *J. Comput. Syst. Sci.*, vol. 28, pp. 80–102, 1984.

[26] Z. Lintao, C. Madigan, M. Moskewicz, and S. Malik, "Efficient conflict driven learning in a boolean satisfiability solver," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers*, 02 2001, pp. 279–285.

[27] A. Teiz, P. Gribomon, G. Yulen, A. Pirot, D. Rolan, D. Snaiers, M. Vokler, P. Goshe, P. Vol'per, E. Greguar, and F. Del'sart, *Logicheskii podkhod k iskusstvennomu intellektu: ot modal'noi logiki k logike baz dannykh [Logic approach to Artificial Intelligence: from modal logic to database logic]*, G. Gavrilov, Ed. M.: Mir, 1998, (In Russ.).

[28] S. K. Klini, *Matematicheskaya logika [Mathematical logic].* Moskva : Mir, 1973, (In Russ.).

[29] A. Teiz, P. Gribomon, and Z. L. i dr., *Logicheskii podkhod k iskusstvennomu intellektu: ot klassicheskoi logiki k logicheskomu programmirovaniyu [Logic Approach to Artificial Intelligence: From Classical Logic to Logic Programming]*, G. Gavrilov, Ed. M.: Mir, 1990, (In Russ.).

[30] A.G.Dragalin, *Konstruktivnaya teoriya dokazatel'stv i nestandartnyi analiz [Constructive evidence theory and non-standard analysis]*, G. E. Mints, M. K. Valiev, E. G. Dragalina-Chernaya, N. M. Nagornyi, N. N. Nepeivoda, V. N. Sadovskii, and E. D. Smirnova, Eds. Editorial URSS, 2003, (In Russ.).

[31] V. N. Vagin, E. Y. Golovina, A. A. Zagoryanskaya, M. V. Fomina, and D. A. Pospelov, *Dostovernyi i pravdopodobnyi vyvod v intellektual'nykh sistemakh [Reliable and plausible inference in intelligent systems]*, 2nd ed., pod red. V. N. Vagina and D. A. Pospelova, Eds. M. : Fizmatlit, 2008, (In Russ.).

[32] V. Golenkov, V. Tarasov, O. Eliseeva, N. Gulyakina, T. Lemesheva, N. Bezzubenok, V. Ivashenko, T. Gavrilova, E. Vasil'eva, V. Emel'yanov, R. Serdyukov, S. Akulich, O. Lubnevskii, and Y. Tel'nov, *Intellektual'nye obuchayushchie sistemy i virtual'nye uchebnye organizatsii: Monografiya [Intelligent Learning Systems and Virtual Learning Organizations: Monograph]*, V. Golenkov and V. Tarasov, Eds. Mn.: BGUIR, 2001, (In Russ.).

[33] I. Z. Batyrshin, A. O. Nedosekin, A. A. Stetsko, V. B. Tarasov, A. V. Yazenin, and N. G. Yarushkina, *Nechetkie gibridnye sistemy. Teoriya i praktika [Fuzzy hybrid systems. Theory and practice]*, pod red. N. G. Yarushinoi, Ed. M. : Fizmatlit, 2007, (In Russ.).

[34] L. Behounek, P. Cintula, and P. Hájek, *Introduction to mathematical fuzzy logic*, 01 2011, vol. 37, pp. 1–101.

Интеграция логических моделей представления и обработки знаний в смысловом пространстве

Ивашенко В. П.

В данной статье рассматривается подход к представлению логических структур и схем, как моделей логической обработки знаний, в семантическом пространстве в виде семантических сетей. Исследуются некоторые свойства семантического пространства и логических моделей, такие как топологические, метрические и валюационные (нормовые) свойства. Предложены понятия для онтологического представления классических и неклассических логических формул, классов и отношений.

Received 13.03.2023

Principles of Problem Solving in Distributed Teams of Intelligent Computer Systems of a New Generation

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

***Abstract*—The article considers an approach to the organization of problem solving within the distributed collective of intelligent computer systems that are part of the OSTIS Ecosystem (ostis-systems). The classification of agents within such a system and the principles of their interaction are considered.**

***Keywords*—OSTIS Technology, OSTIS Ecosystem, problem solver, multi-agent approach**

I. Introduction

One of the key components of the *intelligent system*, which provides the ability to solve a wide range of *tasks*, is *problem solver*. Their peculiarity in comparison with other modern *software systems* is the necessity to solve *tasks* in conditions when the needed information is not explicitly localized in *knowledge base* of *intelligent system* and must be found in the process of solving tasks based on any criteria.

The development of intelligent systems problem solvers at this moment is usually considered in the context of individual (independent) intelligent systems operating in some environment (the user is part of this environment, if there is one). At the same time, there is an obvious tendency of modern information technologies to move from individual systems to collectives of distributed interacting computer systems, in particular, to distributed data storage and distributed computing. In the case of intelligent computer systems, the most important property of the systems involved in such collectives becomes interoperability, which is the ability of the system to interact in a coordinated manner with other similar systems in order to solve any problems. Therefore, the transition from the development of problem solvers of individual intelligent systems to problem solvers of interacting interoperable intelligent systems is especially relevant, including the development of principles for solving problems in such distributed teams, in light of the solution of all the problems outlined above.

The expansion of the application areas of *intelligent systems* requires them to be able to solve the so-called *complex problems*, the solution of each of which requires combining several models of problem solving, while it is not known a priori in what order and how many times a particular model will be used. *problem solvers*, in which several problem solving models are combined, are called *hybrid problem solvers*, and intelligent systems, in which various *types of knowledge* and various *problem solving models* are combined – *hybrid intelligent systems* [1].

Improving the efficiency of the development and use of *hybrid intelligent systems* requires the unification of models for the representation of various *types of knowledge* and *models of knowledge processing*, which would make it easy to integrate components based on it correspond to different models of problem solving. Such models based on the unified semantic representation of information are proposed within the framework of the OSTIS Technology [2]–[4]. The systems developed using this technology are called *ostis-systems*. The encoding of information in the memory of *ostis-systems (sc-memory)* is based on the language of unified semantic networks, called *SC-code*. The elements of such a semantic network (sc-text) are called *sc-elements (sc-nodes, sc-arcs, scedges)*.

Within the frame of this article, it is proposed to clarify the principles of solving problems by a distributed team of interacting intelligent computer systems on the basis of the previously proposed principles of solving problems within the framework of individual intelligent computer systems (see [3]). To solve this problem, it is proposed to consider such a system of interacting intelligent computer systems as a multi-agent system and clarify the principle of agent behavior in such a system.

II. Principles of developing problem solvers for individual computer systems

The proposed approach to problem solving is based on a number of ideas related to the concept of situational management proposed in the work of D. Pospelov [5] and developed in works in the field of multi-agent knowledge processing [6].

Let's consider the principles underlying the proposed approach to the development of *hybrid problem solvers* [3], [4]:

- as a foundation for creating a hybrid *problem solver* model, it is proposed to use *multi-agent approach*. This approach allows to provide a foundation for the construction of parallel asynchronous systems with a distributed architecture, to increase the modifiability and performance of the developed *problem solvers*;

- *problem solver* is proposed to be considered as a hierarchical system consisting of several interconnected levels;
- it is proposed to record <u>all</u> information about the solver and the problems solved by them using *SC-code* in the same *knowledge base* as the actual subject of systems *knowledge*. This will allow, firstly, to ensure the independence of the developed *problem solvers* from the platform for interpreting semantic models of ostis-systems (*ostis-platforms*, see [4]), secondly, to enable the system to analyze the processes occurring in it, optimize and synchronize their execution, that is, to ensure *reflexivity* of the designed *intelligent systems*.

The focus on *multi-agent approach* as a foundation for building *hybrid problem solvers* is owing a number of advantages of such an approach [3], [4], [7]–[9].

In accordance with these principles, the *ostis-system problem solver* is proposed to be divided into components corresponding to classes of logically atomic actions in semantic memory, called sc-agents.

Logical atomicity of performed *sc-agent actions* assumes that each *sc-agent* reacts to its corresponding class of *situations* and/or *events* occurring in sc-memory and performs a certain transformation of *sc-text* located in the semantic neighborhood of the processed *situations* and/or *events*. At the same time, each sc-agent generally does not have information about which other *sc-agents* are currently present in the system and interacts with other *sc-agents* exclusively through the formation of some constructs (usual—action specifications) in the common *sc-memory*. Such a message can be, for example, a question addressed to other *sc-agents* in the system (it is not known in advance which one specifically), or an answer to a question posed by other *sc-agents* (it is not known in advance which one specifically). Thus, each *sc-agent* at any given time controls only a fragment of *knowledge base* in the context of the task being solved by this agent *task*, the state of the rest of the *knowledge base* is generally unpredictable for *sc-agent*.

A certain *method* can be assigned to an action class, that is, a description of how any or almost any (with explicit exceptions) action belonging to this *action class* can be performed. Since a specific *class of actions* corresponds to some specific *class of tasks*, we can say that the method describes a way to solve any tasks belonging to a given class. The concept of a method can be considered a generalization of the concept of "program and therefore within the framework of *OSTIS Technology* the terms "method"and "program"are synonymous, and the term "method representation language"is synonymous with the term "programming language".

Since it is assumed that copies of the same sc-agent or functionally equivalent *sc-agents* can work on different ostis-systems, while being physically different *sc-agents* , then it is expedient to consider the properties and classification not of *sc-agents*, but of classes of functionally equivalent *sc-agents*, which we will call *abstract scagents*. *abstract sc-agent* is understood as a certain class of functionally equivalent *sc-agents*, different instances (that is, representatives) of which can be implemented in different ways.

III. Hierarchy of sc-agents from the point of view of the method representation language lev

Sc-agents can be classified according to various criteria. Since we can talk about the hierarchy of *methods* (methods of interpretation of other methods) and, accordingly, the hierarchy of skills, then there is a need to talk about the hierarchy of *sc-agents* that provide interpretation of a particular method. In this context, we can talk about the hierarchy of *sc-agents* in two aspects:

- *abstract sc-agent* (and, accordingly, *sc-agent*) can uniquely correspond to method (sc-agent program) describing the activity of this *sc-agent*. Such agents will be called *atomic abstract sc-agents*;
- *abstract sc-agents* sometimes it is advisable to combine such agents into collectives, which can be considered as one integral *abstract sc-agent*, from a logical point of view, working on the same principles as *atomic abstract sc-agents*, that is, reacting to events in *sc-memory* and describing its activities within this memory.Such a abstract sc-agent will not correspond to any specific method stored in scmemory, but the rest of the specification of *abstract sc-agent* (initiation condition, description of the initial situation and the result of *sc-agent* and so on) remains the same as that of *atomic abstract sc-agent*. Therefore, we can say that the concept of atomicity/non-atomicity of abstract *sc-agent* indicates how the implementation of this *abstract sc-agent* is specified – by specifying a specific method (program sc-agent) or by decomposition of *abstract sc-agent* to simpler ones. It is important to note that *non-atomic abstract sc-agents* can also be part of other, more complex *non-atomic abstract sc-agents*. Thus, a hierarchical system of *abstract sc-agents* is formed, generally having an random number of levels.
- In turn, the corresponding *sc-agent* method should be interpreted by some other *sc-agent* of a lower level, and most often by a team of such agents, each of which is assigned its own *method* describing the behavior of this agent, but already at a lower level. Therefore, we can say that the concept of atomicity/non-atomicity of *abstract sc-agents* is applicable within the framework of one *method description language*. In turn, we can talk about the hierarchy of *abstract sc-agents* from the point of view of the level of the language of description of the methods corresponding to such agents. In general, such a hierarchy can also have an unlimited