platform with components that use the methods of *Implementation of the sc-memory in the ostis-platform* and are part of various plug-ins of the current *Software interface of Implementation of sc-memory in the ostis-platform.*

- Providing different levels of access to *Implementation of sc-memory in the ostis-platform*, including the levels of access for different users of the *Software implementation of the ostis-platform.*

It is worth noting that *Software interface of Implementation of sc-memory in ostis-platform* cannot exist separately from the current *Implementation of sc-memory in the ostis-platform.* In addition, it is part of the *Implementation of sc-memory in the ostis-platform*, that is, it is designed and developed in accordance with the implementation of the sc-memory itself. However, if necessary, it can be used for various modifications or versions of the current *Implementation of sc-memory in the ostis-platform.*

Despite the wide range of functionality of the current *Software interface of Implementation of sc-memory in the ostis-platform*, its disadvantages include the following:

- At the level of the *Software interface of Implementation of sc-memory in the ostis-platform*, there is no limit to the range of classes of sc-elements in sc-memory that can be set as arguments, for example, to the *Method of creating an sc-memory element of a given class, corresponding to an sc-node* and *Method of creating an sc-memory element of a given class, corresponding to some sc-connector.*
- Due to shortcomings in the current implementation of the agent architecture in the *software implementation of the ostis-platform* it is impossible for the *Software interface of Implementation of sc-memory in the ostis-platform* to use *Implementation of sc-memory in ostis-platform* stored as a compiled file. First of all, this is due to the fact that platform-specific agents are implemented by means that utilize creation of source files when building the entire platform. Thus compiled files remain dependent on the device where they were built.

In general, isomorphic search can be a useful tool in theoretical studies and some specialized applications, but in most cases there are better ways to work with graphs.

VII. Conclusion

Let us list the main ideas of this work:

- to solve information retrieval tasks in ostis-systems, the *Implementation of the information retrieval subsystem* of the current *Software implementation of the ostis-platform* is used;
- *Implementation of the information retrieval subsystem in the Software implementation of the ostis-platform* has a software interface that can be used in any platform-dependent component (subsystem);
- *Implementation of the information retrieval subsystem in the Software implementation of the ostis-platform* includes iterative methods for searching for sc-memory constructions and methods for searching for sc-memory constructions according to the specified graph template;
- to solve most information retrieval problems, it is sufficient to use iterative methods for searching for sc-memory constructions;
- the current implementation of isomorphic search is not universal and is limited to a certain set of graph templates, and also strongly depends on the state of the knowledge base.

When designing graph templates in one of the languages of the external representation of SC-code [20], one should:

- Minimize the number of cycles by splitting, for example, key constant sets into subsets that are not interconnected in this graph template. If the cycle in the graph cannot be eliminated, then leave it as it is, or reconsider the original problem for the possibility of simplifying its solution.
- Select among all those sc-constructions that can be selected by the search procedure as the first sc-construction, only the one that simplifies the work of the search procedure as much as possible for the specified one in the subject domain.
- Minimize the number of sc-constructions, the removal of which does not change the meaning of the found constructions and/or can be specified/checked later (for example, when the entity is already found and the class membership can be checked later) and/or the removal of which simplifies the choice of path search in a graph isomorphic to the specified graph template.

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.

References

[1] D. Shunkevich, D. Koronchik, “Ontological approach to the development of a software model of a semantic computer based on the traditional computer architecture,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 75–92, 2021.

[2] N. Zotov, “Software platform for next-generation intelligent computer systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 297—-326, 2022.

[3] A. Zagorskiy, “Factors that determine the level of intelligence of cybernetic systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, p. 13–26, 2022.

[4] R. Reinanda, E. Meij, M. de Rijke et al., “Knowledge graphs: An information retrieval perspective,” *Foundations and Trends® in Information Retrieval*, vol. 14, no. 4, pp. 289–444, 2020.

[5] Y. WHAN KIM and J. H. Kim, “A model of knowledge based information retrieval with hierarchical concept graph,” *Journal of Documentation*, vol. 46, no. 2, pp. 113–136, 1990.

[6] A. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom, “Human-assisted graph search: it’s okay to ask questions,” *arXiv preprint arXiv:1103.3102*, 2011.

[7] S. Ma, J. Li, C. Hu, X. Lin, and J. Huai, “Big graph search: challenges and techniques,” *Frontiers of Computer Science*, vol. 10, pp. 387–398, 2016.

[8] J.-A. Fernández-Madrigal and J. González, “Multihierarchical graph search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 103–113, 2002.

[9] G. J. Kowalski, *Information retrieval systems: theory and implementation*. springer, 2007, vol. 1.

[10] W.-C. Yeh, W. Zhu, C.-L. Huang, T.-Y. Hsu, Z. Liu, and S.-Y. Tan, “A new bat and pagerank algorithm for propagation probability in social networks,” *Applied Sciences*, vol. 12, no. 14, p. 6858,2022.

[11] H. Paulheim, “Knowledge graph refinement: A survey of approaches and evaluation methods,” *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.

[12] I. Kivimäki, A. Panchenko, A. Dessy, D. Verdegem, P. Francq, H. Bersini, and M. Saerens, “A graph-based approach to skill extraction from text,” in *Proceedings of TextGraphs-8 graph-based methods for natural language processing*, 2013, pp. 79–87.

[13] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, and D. Wilkins, “A comparison of a graph database and a relational database: a data provenance perspective,” in *Proceedings of the 48th annual Southeast regional conference*, 2010, pp. 1–6.

[14] Ian Robinson, Jim Webber and Emil Eifrem, *Graph databases*. O’Reilly Media, Inc., 2015.

[15] V. Golenkov, N. Guliakina, V. Golovko, V. Krasnoproshin, “Methodological problems of the current state of works in the field of artificial intelligence,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 17–24, 2021.

[16] N. Zotov, “Semantic theory of programs in next-generation intelligent computer systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 297—-326, 2022.

[17] D. Shunkevich, “Ontology-based design of hybrid problem solvers,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 101–131, 2022.

[18] M. K. Orlov and A. P. Vasilevskaya, “Non-procedural problem-solving models in next-generation intelligent computer systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 161–172, 2022.

[19] K. Bantsevich, “Metasystem of the ostis technology and the standard of the ostis technology,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 357–368, 2022.

[20] V. Ivashenko, “General-purpose semantic representation language and semantic space,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 41–64, 2022.

[21] S. Fortin, “The graph isomorphism problem,” 1996.

[22] J. R. Ullmann, “An algorithm for subgraph isomorphism,” *Journal of the ACM (JACM)*, vol. 23, no. 1, pp. 31–42, 1976.

[23] P. Foggia, C. Sansone, and M. Vento, “A performance comparison of five algorithms for graph isomorphism,” in *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*. Citeseer, 2001, pp. 188–199.

[24] B. D. McKay, “Nauty user’s guide (version 2.4),” *Computer Science Dept., Australian National University*, pp. 225–239, 2007.

[25] M. Orlov, “Comprehensive library of reusable semantically compatible components of next-generation intelligent computer systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya in-tellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 261–272, 2022.

[26] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, “A (sub) graph isomorphism algorithm for matching large graphs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 10, pp. 1367–1372, 2004.

Реализация информационно-поисковой подсистемы в Программной платформе ostis-систем

Зотов Н.В.

Описываются назначение и варианты реализации информационно-поисковых подсистем интеллектуальных компьютерных систем нового поколения. Данная работа является формальной спецификацией Реализации информационно-поисковой подсистемы в текущем Программном варианте реализации ostis-платформы, а также её программного интерфейса, и является продолжением серии работ по проектированию и реализации базового Программного варианта реализации ostis-платформы [1], [2]

Received 13.03.2023

Semantic Space Integration of Logical Knowledge Representation and Knowledge Processing Models

Valerian Ivashenko
Department of ntelligent Information Technologies
Belarusian State University of Informatics and Radiolectronics
Minsk, Republic of Belarus
Email: ivashenko@bsuir.by

Abstract—This article deals with an approach to represent both logical structures and schemes, as well as logical knowledge processing models within semantic space in the form of semantic networks. Some properties of the semantic space and logical models such as topological, metric, and valuation (norm) properties are investigated. Concepts for ontological representation of classical and non-classical logical formulas, classes, and relations are proposed.

Keywords—semantic space, integration, problem-solving model integration, logical knowledge representation model, logical knowledge processing model, semantic topological space, semantic metric space, distensible set, unified semantic representation model, language key element, right proposition, uncorrupted proposition, finite set deduction

I. Introduction

The main quality of such intelligent systems as intelligent computer systems of a new generation is the ability to solve problems. Let us consider the indicators of quality which are necessary to ensure, maintain, and develop this quality. Intelligent computer systems of a new generation are classified as cybernetic systems (adaptive systems [1]). For such systems, some of the important quality indicators are [2]:

- self-learning, implemented (or automated) through self-improvement, provided by the universality of the intelligent system operating on a variety of knowledge types and (hybrid) problem-solving models, as well as their deep integration (degree of interosculation);
- interoperability that requires mutual understanding based on semantic compatibility;
- ability to coordinate their plans and intentions and coordinate them in a decentralized manner that requires the integration of plans or their parts;
- semantic compatibility (for types of knowledge and problem-solving models) that is matching between systems and concepts and requires formalization of semantic representation of information through unification, where the latter is the main indicator of degree of convergence between intelligent computer systems and their components.

Thus, the requirement of deep integration (knowledge types and problem-solving models) is represented in the intelligent computer systems of a new generation. At the same time, intelligent computer systems of a new generation are characterized by the degree of convergence, unification, and standardization of intelligent computer systems and their components and the corresponding degree of integration (depth of integration) of intelligent computer systems and their components. Currently, there is a strong need for focusing on potentially universal (that is, capable of quickly acquiring any knowledge and skills), synergistic intelligent computer systems with "strong" intelligence, since the maximum level of cybernetic system processor quality in terms of the variety of problem-solving models interpreted by the processor of a cybernetic system is its universality, that is, its "principal ability to interpret any model for solving both intelligent and non-intelligent problems". In order to ensure quality, the importance of moving to hybrid individual intelligent computer systems is highlited, where convergence and integration of various problem-solving models and various knowledge types is carried out. Due to the fact that different intelligent computer systems may require different combinations of problem-solving models (models for representing and processing knowledge of various types), which have been developed by a large number at the present time, and in the development and implementation of various intelligent computer systems, the appropriate methods and tools must guarantee logical semantic compatibility of the developed components and, in particular, their ability to use common information resources, then for this, the need to unify these models is indicated. The creation of intelligent computer systems of a new generation involves the creation of an appropriate technology for integrated design and integrated support for the stages of the life cycle of these systems. It is noted that in order to create a technology for integrated design