this kind of interpreter. On the contrary, the problem lies in the model of asynchronous access to sc-memory, which prevents the collection of sc-agents that are part of the *Implementation of the SCP Language interpreter* to work smoothly. To implement a full-fledged collective of *ostis-systems* interacting with each other, it is necessary to transfer the Software implementation of the ostis-platform from the specialized ostis-platforms class to the basic ostis-platforms class. Thus, it is necessary to switch to a new version of the ostis-platform (not a modification (!)), which will contain the current Implementation of the SCP Language interpreter.

- The current Implementation of ostis-platform scmemory is efficient for storing large amounts of knowledge in ostis-systems knowledge bases. However, in information retrieval problems, rather complex tools and subsystems are required to ensure the most effective solution of these problems. So, for example, to find all pairs of a given relation whose first component is a given sc-element, it is necessary to check the entire list of outgoing sc-connectors of a given sc-element, including those sc-connectors that do not have the specified syntactic or semantic sc-element class. The solution to this problem is possible by modifying the existing Implementation of ostis-platform sc-memory, namely, the implementation of a new sc-memory model (for example, on the file system of the modern Linux operating system).
- Implementation of the OSTIS Ecosystem [20], [21] requires strong development of the Implementation of the subsystem for interacting with the external environment using languages of network interaction, with the help of which ostis-systems, which are developedon the current Software implementation of the ostisplatform, will be able to fully communicate with each other. The transition of Software implementation of the ostis-platform from the class of server platforms to the class of clientserver platforms is required.

VII.Conclusion

Let us briefly list the main provisions of this work:

- The current Software implementation of the ostisplatform is cross-platform, which allows:
  - developing and maintaining the state of its components, regardless of the implementation of the platforms on which the tools for their design and development are used;
  - using it to solve problems on any available devices.
- The current Software implementation of the ostisplatform is multi-user, that is, it allows processing several actions at the same time.
- The current Implementation of memory in the ostisplatform is complete enough to:
  - one-to-one interpret sc-models of ostis-systems, including external information constructions that do not belong to the SC-code;
  - develop platform-specific components that require access to sc-memory (for example, the Software interface of Implementation of ostis-platform scmemory).
- The current Software implementation of the ostisplatform is specialized, that is, it allows creating only platform-dependent ostis-systems.
- On the basis of the current *Software implementation of the ostis-platform, the interpreter of sc-models of ostis-systems user interfaces, interpreter of logical models for solving problems in ostis-systems*, as well as the *manager of reusable ostis-systems components* are used.

In this article, the principles underlying the Software implementation of the ostis-platform, the principles of documenting its components, as well as the prospects for further development are described.

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] Golenkov V., Guliakina N., Golovko V., Krasnoproshin V. (2021). 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]*, 17–24.

[2] A. Sokolov, A. Golubev, "Sistema avtomatizirovannogo proektirovaniya kompozicionnyh materialov. CHast' 3. Grafoorientirovannaya metodologiya razrabotki sredstv vzaimodejstviya pol'zovatel'-sistema [System of computeraided design of composite materials. Part 3. Graph-oriented methodology for the development of user-system interaction tools]," *Izvestiya SPBGETU LETI*, pp. 43–57, 2021.

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

[4] A. M. Ouksel and A. Sheth, "Semantic interoperability in global information systems," SIGMOD Rec., vol. 28, no. 1, p. 5–12, Mar. 1999.

[5] 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.

[6] 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.

[7] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheski sovmestimykh gibridnykh intellektual’nykh komp’yuternykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint, 2021.

[8] 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.

[9] V. Golenkov, N. Guliakina, G. V., and E. V., “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]*, pp. 73–88, 2018.

[10] D. Shunkevich, “Universal model of interpreting logical-semantic models of intelligent computer systems of a new generation,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, p. 285–296, 2022.

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

[12] 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.

[13] C. Chen et al., “Multi-perspective evaluation of relational and graph databases,” 2022.

[14] 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.

[15] “Ontology-based design of intelligent systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 37–56, 2017.

[16] T. Dillon, E. Chang, and P. Wongthongtham, “Ontology-based software engineering-software engineering 2.0,” 04 2008, pp. 13– 23.

[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] 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.

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

[20] A. Zagorskiy, “Principles for implementing the ecosystem of nextgeneration intelligent computer systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, p. 347–356, 2022.

[21] T. Dillon, C. wu, and E. Chang, “Gridspace: Semantic grid services on the web-evolution towards a softgrid,” in *3rd International Conference on Semantics, Knowledge, and Grid, SKG 2007*, 11 2007, pp. 7–13.

Принципы проектирования, структура и перспективы развития Программной платформы ostis-систем

Зотов Н. В.

Данная работа является краткой спецификацией текущего Программного варианта реализации ostisплатформы. Работа показывает принципы, структуру и перспективы развития программной платформы для логико-семантических моделей систем, построенных по принципам Технологии OSTIS.

Received 13.03.2023

Implementation of Information Retrieval Subsystem in the Software Platform of ostis-systems

Nikita Zotov
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: nikita.zotov.belarus@gmail.com

***Abstract*—The article describes the purpose and implementation variants for information retrieval subsystems of next-generation intelligent computer systems. This paper is a formal specification of how the information retrieval subsystem in the current Software implementation of the ostis-platform, as well as its software interface, are implemented, and is a continuation of a series of works on the design and implementation of the basic Software implementation of the ostis-platform [1], [2].**

***Keywords*—information retrieval, information retrieval problem, isomorphic search, graph template, graph information retrieval system, ontological design, graph storage, ostis-platform**

I. Introduction

One of the most important tasks of *intelligent computer systems* [3] is to satisfy the information needs of users. *Intelligent computer systems* should not only find the necessary (relevant) information for the user, but also provide quality answers to the user's questions. Thus, *intelligent computer systems* based on *graph representation of knowledge* should include entire software complexes for searching information relevant to the user — *graph information retrieval subsystems* [4], [5], [6].

Existing *graph information retrieval systems* are based on the use of *graph algorithms* for searching, storing and presenting information [7], [8]. Graphs are used to model relationships between objects, such as web pages on the Internet, users on social networks, or others. In such systems, users can use search queries to find information in a graph. Queries may be similar to those used in traditional information systems, but instead of searching by keywords, the user searches for objects and the relationships between them.

Information retrieval tasks are of great relevance, since at present the amount of information available on the Internet is too large for a person to handle without using appropriate search engines [9]. Information flows are growing every day, and therefore a more efficient and accurate use of information is becoming increasingly important for decision-making, planning,scientific research and other activities. Moreover, the ability to conduct high-quality and accurate information searches is a key skill for people in the modern world.

II. Existing analogs of graph information retrieval systems

Modern *graph information retrieval systems* use the *PageRank* [10] algorithm to determine the relevance of search results. PageRank evaluates the importance of each object in the graph based on the links it contains from other objects, and those objects that are considered more important are ranked higher. Graphs also allow the use of analytical algorithms, such as community detection algorithms, to identify subgraphs that group objects according to certain criteria. This can help users find information that might not be found in a traditional keyword search.

Examples of *graph information retrieval systems are Google Knowledge Graph* [11], *Facebook Graph Search, LinkedIn Skills Graph* [12] and *Neo4j* [13].

The use of *graph data models* in solving *information retrieval tasks* is explained as follows:

- Data processing performance is improved by one or more orders of magnitude when representing data as *graphs*, due to the properties of *graphs* themselves. Unlike *relational databases*, where query performance degrades as the dataset grows with increasing query intensity, *graph data model* performance remains constant even as the dataset grows. This is due to the fact that data processing is localized in some part of a *graph*. As a result, the execution time of each request is proportional only to the size of the *graph* part traversed to satisfy this request, and not to the size of the entire *graph* [14].

- *Graph data models* have tremendous expressive power. *Graph databases* offer an extremely flexible data model and way of representing it. *Graphs* are additive, which provides the flexibility to add new data relationships, new nodes, and new subgraphs