Approaches of Neuro-Symbolic Integration: Large Language Models and Knowledge Bases

Vladimir Golovko
Brest State Technical University
Brest, Belarus
gva@bstu.by

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

Mikhail Kovalev
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
kovalev@bsuir.by

Aliaksandr Kroshchanka
Brest State Technical University
Brest, Belarus
kroschenko@gmail.com

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

Valerian Ivashenko
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
ivashenko@bsuir.by

***Abstract*—The paper reviews modern approaches to the integration of neural network and symbolic artificial intelligence, including architectures using generative models in combination with knowledge bases and agents. Special attention is paid to the limitations of large language models (LLMs) in solving problems related to long-term context, as well as RAG-type mechanisms and modern agent-based systems are analyzed. An architecture of intelligent systems based on OSTIS Technology is proposed, in which LLMs are integrated into the problem solver. The necessity of formal description of generative AI methods as part of the problem solver is substantiated.**

The authors emphasize the advantages of moving the knowledge base from the position of a source of context for LLMs to the position of a shared semantic memory that combines different types of knowledge, from problem descriptions and solution models to the history of agent interaction. This approach provides automatic verification of knowledge, accumulation of experience and transparency of decisions.

***Keywords*—neurosymbolic AI, logical-semantic models, artificial neural networks, large language models, knowledge base, ontologies, intelligent systems architecture, OSTIS Technology**

I. Introduction

Symbolic and neural network approaches to building artificial intelligence (AI) have historically been developed based on different principles: the former on logical-semantic structures, rules and knowledge bases, the latter on learning from examples and generalization. For decades, attempts have been made to combine the strengths of both approaches to create more versatile and explainable intelligent systems. However, it is only in recent years, with the growing capabilities of large language models and the development of the corresponding infrastructure, that the integration of these approaches has received a new breath and is reaching the application level.

The urgency of finding effective hybrid solutions has become particularly important with the rapid proliferation of generative models capable of producing text, code, and other types of data at a near-human level. Despite their impressive capabilities, they face limitations in solving problems that require knowledge management, logical reasoning, long-term memory, and decision transparency. These limitations are driving the rethinking of intelligent systems architectures and the implementation of solutions that combine the flexibility of artificial neural networks (ANNs) with the formal rigor of logical-semantic models.

In this paper, current approaches to the integration of neural networks and logical-semantic models are reviewed, and we propose our own approach to integrating artificial neural networks (including large language models) with a knowledge base and a problem solver.

II. Modern approaches to the integration of artificial neural networks and logical-semantic models

Studies of integrations of neural network and symbolic approaches to AI have developed into a separate direction – neurosymbolic AI [1]. In the works [2]–[5] an analysis of various approaches to the integration of these approaches, including their advantages and limitations, is presented.

In recent years, systems that combine various AI methods within a single architecture have been actively developed. Special attention is paid to the integration of ANNs with knowledge bases, which in foreign studies are increasingly referred to as knowledge graphs [6], [7].

Let us consider the key directions and approaches to the implementation of such integration.

A. ANNs as a sensor of an intelligent system

This approach is that ANNs serve as sensors that extract information from various inputs (text, images, sounds) and translate them into formats suitable for use in a knowledge base. For example, ANNs extract entities and relationships from textual data or images. The information obtained by ANNs is further used to solve various problems within an intelligent system [8]–[11].

The main advantage of this approach is the ability to integrate ANNs with a knowledge base to extract knowledge from various sources without human involvement. However, the quality of the extracted knowledge depends on the quality of the input data and the accuracy of the ANNs.

B. Knowledge base for processing and verification of training datasets and ANN output data

In modern problems solved with ANNs, the requirements to the degree of semantic consideration in the solution are constantly growing. Taking into account the peculiarities of ANNs, the degree of semantic consideration is increased by searching for logical and semantic contradictions in the ANNs training and output data. In case of detection of such contradictions in the output data it becomes possible to adjust the ANNs weights in the process of training. In addition, based on knowledge bases, synthetic data can be generated, which is then used to train ANNs, which is especially useful in cases where the amount of real data is limited or difficult to obtain [12]–[14].

This approach improves the accuracy and validity of results through logical reasoning. However, it requires a complex system of logical rules and knowledge, and is problematic to scale for large amounts of data.

C. ANNs process and generate fragments of the knowledge base

The approach has received a major boost due to the introduction of graph neural networks capable of taking graph structures as input. Such networks are used to predict relationships in the knowledge base [15], classify and cluster fragments of the knowledge base [16].

The main advantage of this approach is the ability to work with complex graph data structures and identify new relationships between entities. However, it requires the development of specialized graph neural networks for specific classes of problems.

D. Knowledge base for explaining how ANNs work

In this approach, the knowledge base stores logical rules that correspond to specific neural elements or layers of ANNs. This approach is part of the Explainable AI trend [17] and is designed to solve the classic problem of ANNs – the "black box" problem [18]. ANNs are trained on historical data, they are not able to explain why exactly such results were obtained. The essence of this approach to solving this problem is to try to generate plausible logical rules that are put in correspondence with the stages of computation within the ANNs. When the ANNs operate, the stages of computation that had the greatest impact on the result are evaluated, and then a chain of logical rules is constructed to explain it [19]–[21].

The main advantage of this approach is the ability to ensure that ANNs solutions are explainable at some level, but it requires a complex system of logical rules and knowledge, which is not always applicable for ANNs with a large number (billions) of parameters.

E. Neurosymbolic architectures for sustainable continual learning

Research in continual learning has identified one of the key problems of ANNs – **catastrophic forgetting**, in which training a model on new data leads to complete or partial loss of previously learned knowledge [22].

In the paper [23] the NeSyBiCL (Neuro-Symbolic Brain-Inspired Continual Learning) framework is presented, which consists of three main modules: a convolutional neural network (CNN) for feature extraction, ANN for fast inference (System 1), and a symbolic mechanism based on knowledge graphs (System 2). In the symbolic mechanism, information from problems is formalized in the form of entities (graph nodes) and relations (graph edges). Problems are solved by logical inference based on these knowledge graphs.

The basic idea is that while ANNs are prone to catastrophic forgetting, the symbolic mechanism, due to its representation of knowledge as graphs and logical inference, may be more resistant to this problem. The interaction between the two systems ensures knowledge transfer, with the symbolic mechanism being resistant to catastrophic forgetting but less accurate than ANNs. The authors show an average 41% reduction in forgetting on two composite benchmarks.

F. Summary

All the problems solved by the above approaches are undoubtedly relevant, but their solutions are often narrowly focused on a specific problem from some subject domain. Neural network and logical-semantic models of problem solving are used in a clearly defined sequence, are manually integrated with each other and are not aimed at solving complex problems of arbitrary formulation. The figure 1 shows the scheme of interaction of AI modules integrated into an intelligent system in a "manual" way.

III. Race of generative models: opportunities, limitations, benchmarks

The development of neurosymbolic approaches opens new opportunities for integrating ANNs with knowledge

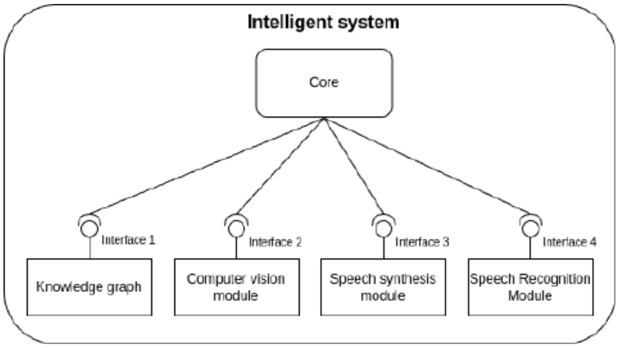

Figure 1. Scheme of interaction between the problem solver and AI modules integrated in a "manual" way

bases, which allows solving complex problems in various domains. However, in parallel, generative models such as large language models are being actively developed, showing impressive results in text processing and generation. Understanding the current state of these technologies is essential in order to assess their potential and capabilities in future integrations with other components of intelligent systems. This chapter reviews the current state of generative AI and large language models.

The current rapid development of neural network models for natural-language information processing began with the publication of a revolutionary paper "Attention is All You Need" in 2017, which marked the emergence of a new architecture for neural network models – the attention-based transformer [24]. Transformer models do not use recurrent feature layers to implement them.

The key advantages that transformer networks have are:

1) Self-attention – a mechanism that allows the model to take context into account at any distance (unlike recurrent neural networks, which suffer from loss of context as the test fragment being analyzed increases);
2) Parallelization – ability to learn faster;
3) Scalability – the ability to build deep and broad models.

With the ability to represent any data (video, audio sequences, single images, etc.) in the format of a sequence of numerical representations (embedding vectors) that can then be processed by the model, Transformers have become the cornerstone of all major SOTA models for processing natural language, visual (computer vision) and multimodal information.

The next milestone in the development of Transformers can be considered the emergence of the BERT architecture in 2018, which has significantly improved the quality of NLP problem solving [25].

In 2019, there is a research shift towards Causal Language Models (CLMs). Unlike Masked Language Models (e.g., BERT), whose application focuses on text comprehension problems, CLMs are used for text generation. The research findings have led to the ability to apply a single model to multiple problems without the need for fine-tuning.

The year 2020 marked the launch of OpenAI's GPT-3, which kicked off the race for large language models as the basis for realizing commercial products [26]. The key milestones in this race were:

- PaLM [27], Chinchilla [28] (2022): the development of these models showed that it is not only the size of the models that matters in implementation, but also the effectiveness of their training;
- OPT [29], BLOOM [30] (2022) were attempts to create open analogs of GPT-3;
- LLaMA [31] (2023) – a compact model achieving excellent quality, which started a wave of fine-tuning applications by a wide range of professionals, not only scientists and practitioners;
- LoRA [32] (2021) / QLoRA [33] (2023) – the introduction of easy ways to fine-tune and run LLMs on consumer GPUs.

The time period of 2022–2024 was the development of instructional fine-tuning (GPT-3.5, InstructGPT [34] – mass training of models to follow human instructions). This made the way of interacting with AI systems more understandable and efficient. In addition, models began to be trained in step-by-step reasoning and tool invocation (Chain-of-Thought [35], Toolformer, ReAct [36]).

The time period of 2024-2025 can be characterized as the emergence of reasoning models (Claude 3, DeepSeek R1, GPT-4, o3, o4-mini, etc.) and the blossoming of multimodal models (GPT-4V, Gemini, Claude 3 – text and image processing models). These models are widely used in the fields of medicine, industrial design, law, etc.

The future development of LLMs is determined by industry leaders who have the resource capabilities to implement training of such models and their deployment. These are such companies as OpenAI, Google DeepMind, Meta, xAI, various Chinese and Russian organizations. Among the main trends can be identified:

1) Development of AGI models (Artificial General Intelligence) – models capable of performing any intellectual problem at the human level or higher, demonstrating versatility, flexibility, and self-learning ability in various domains.
2) Focus on "rational agents" by designing language models with an emphasis on making their behavior as consistent as possible with the principles of rational decision-making.
3) Emphasis on openness and accessibility of models.
4) Intensive development in closed and open formats.

Currently, various benchmarks are used to compare large language models. The main ones are: **AIME 2024**, **GPQA Diamond**, **HLE (Humanity's Last Exam)**, **MMLU (Massive Multitask Language Understand-**

ing), **ARC (AI2 Reasoning Challenge)**, **GSM8K**, **HELLOWORLD**, **HumanEval**, **MT-Bench**, **Arena Elo**.

For example, **AIME 2024** – a benchmark with "Olympiad math problems" taken from the actual 2024 round. AIME is the second in a series of two rounds used as a qualifying round for the U.S. Math Olympiad. It is attended by those in the top percent of the first round, approximately 3,000 people from across the country.

GPQA Diamond – a benchmark that contains questions in biology, physics, and chemistry, but such that even PhDs from these fields and with internet access get only 65% correct (spending no more than half an hour on each problem) [37].

Humanity's Last Exam (HLE) – a benchmark covering 3,000 unambiguous and easily verifiable academic questions in math, humanities, and science provided by nearly 1,000 subject matter experts from more than 500 institutions in 50 countries, providing expert-level human performance on closed-ended academic questions. It was developed in partnership with the Center for AI Safety and Scale AI [38].

Other benchmarks aim to test models in different aspects, e.g. **GSM8K** tests models for correct arithmetic and reasoning, **ARC** tests models for logic and common sense, **MT-Bench** tests models for general dialog competence, **Arena Elo** crowdsources ratings from people (e.g. **Chatbot Arena**). Large language model rankings are regularly updated for new models and common benchmark types [39].

IV. Large language models and long-term memory

Large language models have shown impressive results in text processing and generation, but they also have their drawbacks, such as limited context and lack of built-in ability to deal with external sources of information. To overcome these limitations, researchers are actively developing new approaches such as **RAG** (Retrieval-Augmented Generation) [40] that allow models to retrieve and utilize external data, which is an attempt to provide such models with long-term memory. In addition, approaches to integrating these models with agents are explored to enable intelligent systems to autonomously interact with the environment. This chapter considers how RAG and similar approaches are used to improve the capabilities of large language models, and explores the role of agents in this context.

RAG is an architectural approach in which a large language model accesses external data through search before generating a response. Unlike classical LLMs, systems built using RAG:

- keep up-to-date as there is the possibility of updating knowledge;
- are controlled (only sources that have been explicitly selected are used);
- is explainable in the sense that it is possible to determine where the information for an answer came from.

A common RAG application architecture (fig. 2) includes the following components:

- **User Interface.** The entry point for interacting with the system. Receives natural language (NL) user requests and passes them to the context manager. Provides visualization of responses and input error handling.
- **Context Manager.** Generates query context by accessing repositories, ranks results, passes them to LLMs, and provides source citations to justify responses.
- **Vector Database.** Provides vector-based context search by comparing the embedding vectors of document fragments stored there with the embedding vector of the user's question. Other search options are also available, such as keyword search.
- **Knowledge Graph/Knowledge Base.** Stores structured knowledge in the form of semantic relationships. Allows you to search related entities extracted from documents [41].
- **Large Language Model (LLMs).** Generates natural language responses using promts and context from repositories. Can be a cloud service or an on-premises model.
- **Storage Manager.** Responsible for document pre-processing and indexing. Includes structure parsing, text extraction, and repository interaction.

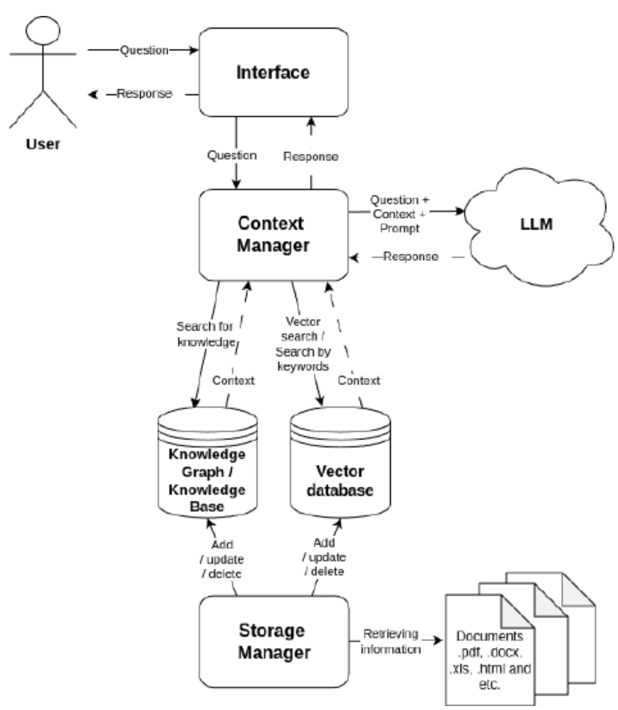

Figure 2. Common architecture of modern RAG applications

In addition to the ability to actualize knowledge and explain decisions, the development and deployment of various AI agents that extend the functionality of LLMs is becoming increasingly important.

The **LLMs AI agent** is a software module that allows the LLMs not only to generate a response to a user request based on context from various repositories, but also to perform actions using external tools. In addition, AI agents can interact with each other. Tools are external components or implementations of problem-solving models that an agent can use when executing a query.

Examples of tools are: search, calculator, calendar, various forms, parsers, API requests. Modern large language models can interact with tools by means of agents, call them with the required parameters, and use the responses received from tools.

For example, if a user needs to find a cheap airfare and book a hotel near a train station, he formulates a query. The AI agent, receiving the query, searches for airline tickets for the specified destination, compares prices, goes to the hotel website, and books the room. In this example, the LLMs AI agent contains a number of components that allow it to perform the above actions:

- planning (what needs to be done);
- implementation (which tools to use);
- memory (how to store intermediate data);
- observation (how to process the results of tool calls).

Popular libraries and frameworks that implement work with agents include: **LangChain, LangGraph, CrewAI, AutoGen, BabyAGI, OpenAI Functions / Tools API**.

Figure 3 shows a common architecture of systems that utilize AI agents.

Despite significant progress in RAG applications and the use of LLMs AI agents providing a new wave of automation in various domains, existing approaches still face a number of fundamental challenges:

- **Knowledge quality and consistency issues.** Context stores can contain conflicting information, leading to incorrect agent inferences.
- **Limited self-learning and experience accumulation issues.** Agent actions are typically not captured in context stores, which prevents error analysis and system improvement. Communication between agents, agents and repositories, agents and tools takes place either in natural language or through specialized protocols. Examples of such protocols are Model Context Protocol (MCP) and Agent2Agent (A2A). There are also no effective mechanisms for accumulating and transferring experience between different systems.
- **Explicability and transparency issues.** Some agent decisions remain non-transparent, especially if their justification does not fall within the context of LLMs, which makes it difficult to trace causal relationships between agent's actions and the final result. It is worth noting that ensuring transparency in modern AI systems is key in many applications, and in some domains, mandatory [42].

The evolution of intelligent systems requires a shift from simple problem-solving to hybrid systems with advanced mechanisms for self-analysis, continuous improvement, and synergy of different problem-solving models.

V. Proposed Approach

Current approaches to building intelligent systems show a clear tendency to centralize the architecture around LLMs, where external repositories, including knowledge bases, play a supporting role. However, as described in the previous chapter, this approach has significant limitations.

Hybrid intelligent systems ([43], [44]) are designed to overcome such limitations through unified memory and synergy of different problem-solving models.

A. General Architecture

In this paper, we propose to use the OSTIS Technology [44] to build multi-agent systems, which is used to design and implement **ostis-systems** – intelligent systems that solve complex problems based on the unification of knowledge and problem-solving models. A unified semantic network with a set-theoretic interpretation is used as a formal basis for representing knowledge and problem-solving models within the framework of OSTIS Technology. Such a representation model is called SC-code (Semantic computer code) [44]. The elements of the semantic network are called sc-nodes and sc-connectors (sc-arcs, sc-edges).

The basis of ostis-systems is an architectural paradigm that goes back to the classical principles of building intelligent systems. It is based on three interrelated components: interface, knowledge base and problem solver.

Interface realizes interaction with the user, including natural language. The interface communicates with the problem solver through a knowledge base.

A knowledge base provides a structured representation of information using semantic networks and ontologies. The basis of the knowledge base is a hierarchical system of **subject domains** (SDs) and their corresponding **ontologies** [45]. Ontologies contain descriptions of concepts necessary to formalize knowledge from the corresponding SDs ontology. Any knowledge describing some problem, its context and specification of solution methods can be represented in the form of SC-code constructions. By representing knowledge in machine-readable code instead of NL-texts, it becomes possible to automatically verify and improve the quality of the knowledge base.

The knowledge base may contain NL-texts. Due to the possibility to regulate the degree (depth) of formalization it becomes possible to regulate the labor intensity of

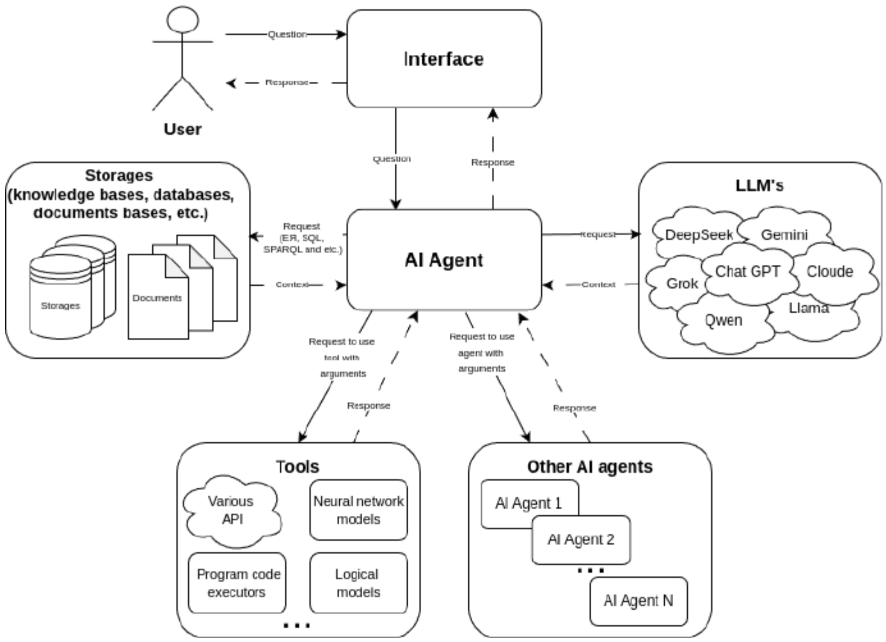

Figure 3. Common architecture of systems utilizing AI agents

knowledge base development. Thus, a gradual transition from NL-texts to partially formalized texts in SC-code is provided.

Problem solver deals with processing of knowledge base fragments using various problem solving models, including logical inference models, neural network models, graph-based problem solving models, etc. At the operational level, such processing is reduced to adding, searching, editing and deleting sc-nodes and sc-connectors of the knowledge base. At the semantic level, such an operation is a *action performed in the memory of the subject of the action*, where, in general, the subject is the ostis-system, and the knowledge base is its memory.

An action is defined as the process of one entity (or some set of entities) influencing another entity (or some set of other entities) according to some goal [46]. Actions are performed according to given problems. A problem is a formal specification of some action sufficient for some entity to perform it. Depending on the specific class of problems, both the internal state of the intelligent system itself and the required state of the external environment can be described [44].

The ostis-systems problem solver is based on a decentralized multi-agent architecture. Each agent is engaged in interpreting actions and, consequently, sc-agents are engaged in solving problems. Communication between agents takes place only through the knowledge base via the event mechanism as follows: an agent or interface generates a problem in the knowledge base and informs in its description that it needs to be solved. Other agents react to this event and, if they can solve it, solve it and leave the solution in the knowledge base. The event of the solution occurrence is reacted to by the agent that put the problem in the knowledge base.

Figure 4 shows the multi-agent architecture of the ostis-systems problem solver.

In such an architecture, the knowledge base moves from the position of a source of context for large language models to the position of a shared semantic memory, in which various types of knowledge describing problems, their context, models of problem solving (including tools), actions of agents, history of problem solving, etc. are represented in a unified form. This provides the following advantages:

- **Knowledge coherence.** Knowledge bases of ostis-systems use a hierarchy of subject domains with well-defined semantic links. SC-code provides unified representation, eliminating inconsistencies through an automatic verification mechanism.
- **Self-learning through reflexive mechanism of experience accumulation.** A decentralized multi-agent architecture captures all agent actions directly in the knowledge base. Each action is formalized with an explicit goal, creating a complete history of actions. Unlike MCP-type protocols, where there is no communication through shared memory, sc-agents operate with formalized knowledge that allows the reproduction and analysis of decision chains.
- **Explainability through semantic transparency.** Agent actions are always linked to specific pieces of the knowledge base, providing an explicit link between inputs, problem-solving models, and outputs. Unlike “black-box’ LLMs, where reasoning is

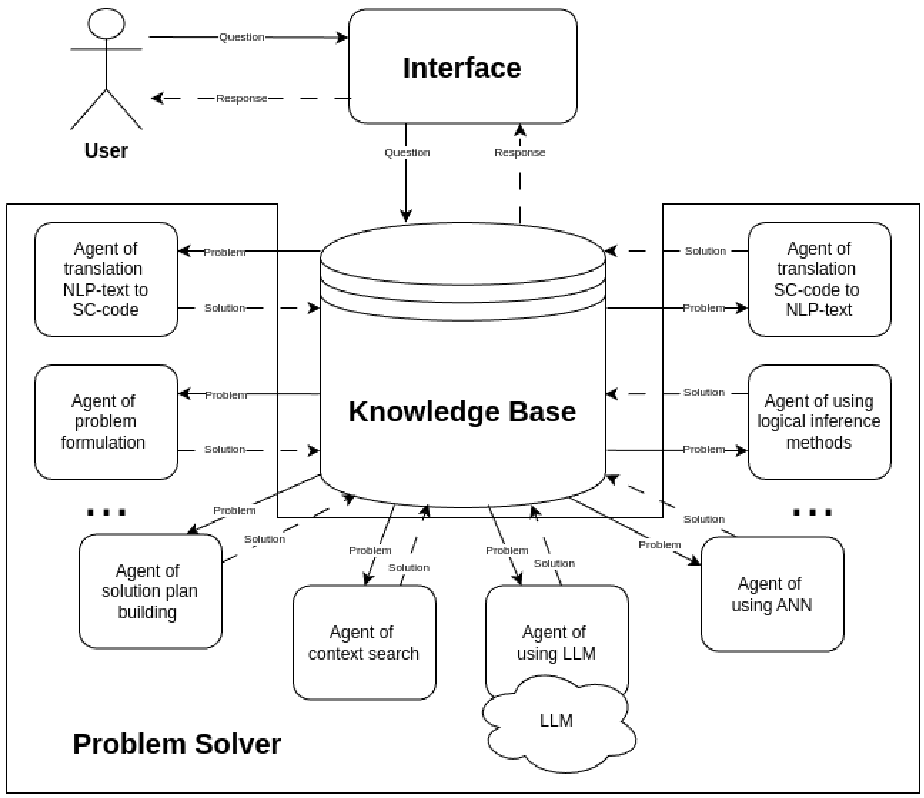

Figure 4. Multi-agent architecture of ostis-systems problem solver.

generated post facto, sc-agents generate solutions through deterministic transformations of SC-code constructions, preserving the complete chain of inference.

- **Decentralized coordination of agents.** The absence of a centralized orchestrator reduces failure risks. Agents interact exclusively through the knowledge base event mechanism, which ensures horizontal scalability and fault tolerance of the system as a whole.

As will be shown later, the proposed architecture does not deny the use of large language models, but assigns them a strictly defined role within the overall system.

B. Role of language models in the system

In the architecture of ostis-systems, LLMs are integrated as specialized agents with well-defined functions. Examples of problems that can be solved by LLMs in ostis-systems are:

1) **Bidirectional Knowledge Translation.** LLMs provide the transformation between NL-texts and formal knowledge representations in SC-code. In the input phase, models structure textual data according to the system ontologies, automatically generating nodes and relations in the knowledge base. In the output phase, they create natural-language explanations strictly tied to specific knowledge fragments in the SC-code. Solving the classic problem of knowledge-based systems – the need for manual formalization – is greatly facilitated (but not completely solved) by using LLMs.
2) **Context-dependent generation.** The generation of LLMs responses is based on the context extracted from the knowledge base. LLMs retrieve semantically verified context from the knowledge base, which eliminates the possibility of "hallucinations" and increases the accuracy of responses.
3) **Decomposition a problem into subproblems.** LLMs can be used to analyze complex problems, decomposing them into atomic subproblems based on the problem classification described in the knowledge base.
4) **Forming a solution plan.** Since all problem solving models available to the system are specified in the knowledge base, LLMs can retrieve these specifications along with the problem formulation and build a solution plan for the problem, including specifying the sequence of agent invocation, input and output specifications, quality metrics, etc. If LLMs are used to solve subproblems, the solution plan includes additional promts that may be selected or generated from a knowledge base based on past problem solving experience.

The integration of LLMs into ostis-systems architecture provides a balance between automating natural language processing and preserving the semantic rigor characteristic of formal methods of knowledge representation. It opens up a wide range of classes of problems that can be solved using LLMs within ostis-systems.

C. Prospects for the development of the approach

The development of the described approach opens new horizons due to the synergy of formal methods of knowledge representation and processing and generative models. Examples of directions for further research are:

1) **Ecosystem.** The concept of Ecosystem is one of the key concepts in OSTIS Technology [44]. Ecosystem of ostis-systems implements a fundamentally new approach to the organization of intelligent systems, where problem solving is carried out through the integration of specialized solvers from different intelligent systems in an integrated distributed knowledge base consisting of knowledge bases of intelligent systems included in the Ecosystem.
   Users interact not with individual systems, but with a personal assistant, which acts as an interface to the Ecosystem. The history of problem solving, general SDs updates is available to all systems within the Ecosystem, which ensures reflection and knowledge accumulation at the level of the entire Ecosystem, rather than individual systems.
   This shifts the focus from training LLMs and increasing their context window, to training the shared long-term memory, which is the integrated knowledge base of the Ecosystem. The experience of intelligent systems from different subject domains (medicine, education, manufacturing) will be accumulated in a single ecosystem. Figure 5 shows the architecture of the OSTIS Ecosystem, indicating the role of large language models.
2) **Training large language models on knowledge base fragments.** The development of the OSTIS ecosystem and the expansion of knowledge bases on SC-code are shaping a fundamentally new type of dataset for LLMs training: instead of unstructured NL-texts, models have access to verified semantic constructs with explicit connections between concepts. This opens up hypothesized benefits, including reduced "hallucinations" and increased inference accuracy.
   The key hypothesis is that training LLMs on SC-code fragments shifts the focus from text generation to knowledge transformation: models learn to operate with formal constructs rather than mimic language patterns.
3) **Explainability and comprehension testing.** Despite advances in generative models, the fundamental question of whether AI systems truly "understand" the problems being solved remains open [47]. Modern LLMs show impressive abilities to simulate meaningful responses, but their conclusions are often based on statistical patterns rather than on conscious work with semantics. The increased influence of semantics on problem solving described in the proposed approach has a good chance of advancing researchers in solving the comprehension problem.

VI. Conclusion

In conclusion, it is necessary to emphasize the prospect of integrating neural network and symbolic approaches to create hybrid intelligent systems. The proposed approach moves the knowledge base from the position of a source of context for large language models to the position of shared semantic memory. This architecture allows combining knowledge about problems, their context, solution models, agent actions and solution history into a single form.

Key advantages are knowledge consistency through the use of a hierarchy of subject domains and SC-code, self-learning through a reflexive experience mechanism in a decentralized multi-agent architecture, and explainability through semantic transparency where agent actions are mapped to specific pieces of the knowledge base.

Further research and development in this area will contribute to the development of next-generation intelligent systems that can effectively integrate neural network and logical-semantic models to solve a wide range of problems.

Acknowledgements

The authors express their gratitude to the scientific staff of the departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and Brest State Technical University for their help and valuable comments.

The work was financially supported by the Belarusian Republican Foundation for Fundamental Research (agreement with BRFFR No. F24MV-011 from 15.04.2024).

References

[1] T. R. Besold *et al.*, "Neural-symbolic learning and reasoning: A survey and interpretation," Nov. 2017, (accessed 2020, Jun). [Online]. Available: https://arxiv.org/pdf/1711.03902.pdf

[2] S. Zhu and S. Sun, "A short survey: Exploring knowledge graph-based neural-symbolic system from application perspective," 2025. [Online]. Available: https://arxiv.org/abs/2405.03524

[3] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence," *AI Communications*, vol. 34, pp. 197–209, 2021.

[4] N. G. Shilov, "Analiz variantov integracii simvol'nykh i nejrosetevykh znanij na primere zadachi klassifikacii modelej predpriyatij [Analysis of integration options for symbolic and neural network knowledge using the example of enterprise model classification]," *Vestnik VGU. Seriya: Sistemnyj analiz i informacionnye tekhnologii*, no. 2, pp. 140–151, 2024.

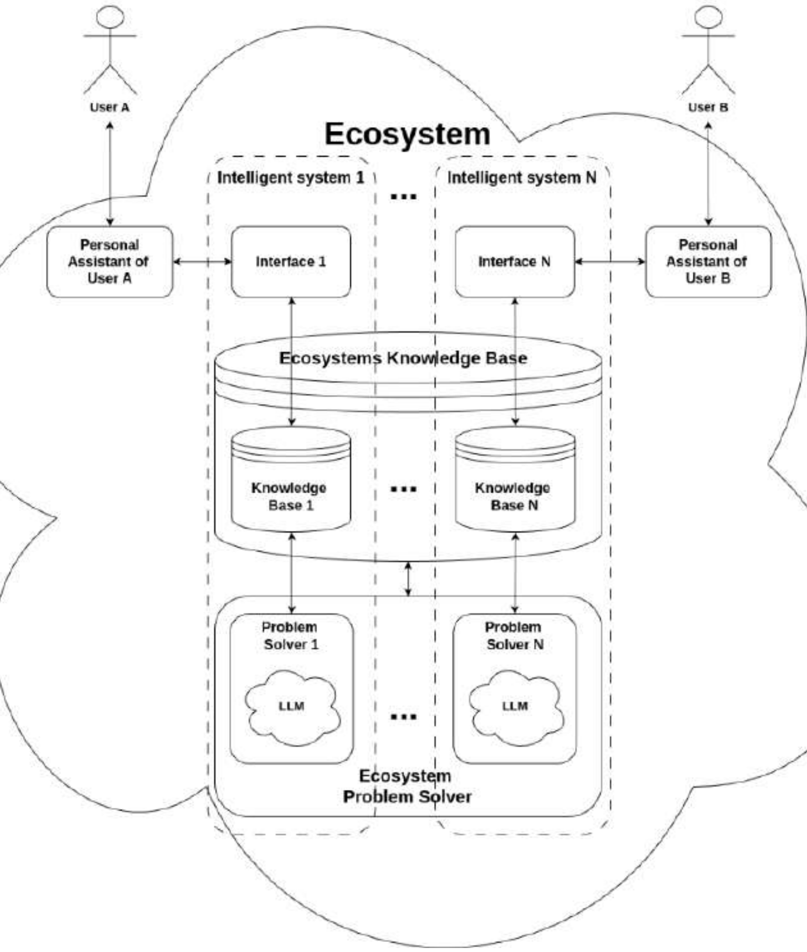

Figure 5. The architecture of the OSTIS Ecosystem and the place of large language models in it

[5] M. V. Kovalev, "Semanticheskie modeli i sredstva razrabotki iskusstvennyh nejronnyh setej i ih integracii s bazami znanij [Semantic models and tools for the development of artificial neural networks and their integration into knowledge bases]," *Informatika*, vol. 20, no. 3, pp. 90–105, 2023.

[6] M. Kejriwal, "Knowledge graphs: A practical review of the research landscape," *Information*, vol. 13, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/4/161

[7] J. Bai, L. Cao, S. Mosbach, J. Akroyd, A. A. Lapkin, and M. Kraft, "From platform to knowledge graph: evolution of laboratory automation," *JACS Au*, vol. 2, no. 2, pp. 292–309, 2022.

[8] A. Ultsch, D. Korus, and T. Kleine, *Integration of Neural Networks and knowledge-based systems in medicine*, 01 2006, pp. 425–426.

[9] A. Kroshchanka, V. Golovko, E. Mikhno, M. Kovalev, V. Zahariev, and A. Zagorskij, "A Neural-Symbolic Approach to Computer Vision," *Open Semantic Technologies for Intelligent Systems*, pp. 282–309, 2022.

[10] A. Hur, N. Janjua, and M. Ahmed, "A survey on state-of-the-art techniques for knowledge graphs construction and challenges ahead," in *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. Laguna Hills, CA, USA: IEEE, Dec. 2021.

[11] N. A. Iskra, V. V. Iskra, and M. M. Lukashevich, "Neirosetevoe raspoznavanie izobrazhenii s ispol'zovaniem ontologicheskogo podkhoda [Neural network image recognition using an ontological approach]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for designing intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2019, pp. 113–122.

[12] V. Glova, K. Anikin, A. Katasyov, and M. Podolskaya, "Formirovanie bazy znanij medicinskoj diagnosticheskoj ekspertnoj sistemy na osnove nechetkoj nejronnoj seti [Formation of knowledge base for medical diagnostic expert system based on fuzzy neural network]," *Issledovaniya po informatike [Informatics Research]*, pp. 31–46, 2007.

[13] 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," *Open semantic technologies for intelligent systems*, pp. 133–145, 2018.

[14] F. Nielsen and L. Hansen, "Inferring visual semantic similarity with deep learning and wikidata: Introducing imagesim-353?" vol. 2106. CEUR-WS, Jan. 2018, pp. 56–61, 1st Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies , DL4KGS 2018 ; Conference date: 04-06-2018 Through 04-06-2018.

[15] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web*, ser. Lecture notes in computer science. Cham: Springer International Publishing, 2018, pp. 593–607.

[16] M. Comet and L. Teresa, "Clustering knowledge graphs (seminar paper)," ResearchGate, mar 2019.

[17] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson, "Explainable artificial intelligence: an analytical review," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 11, no. 5, Sep. 2021.

[18] D. Castelvecchi, "Can we open the black box of ai?" *Nature News*,

vol. 538, no. 7623, Oct. 2016.
[19] A. Saranya and R. Subhashini, “A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends,” *Decision Analytics Journal*, vol. 7, p. 100230, jun 2023.
[20] A. Averkin and S. Yarushev, “Explainable artificial intelligence: Rules extraction from neural networks,” in *11th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions and Artificial Intelligence - ICSCCW-2021*, ser. Lecture notes in networks and systems. Cham: Springer International Publishing, 2022, pp. 102–109.
[21] A. Kroshchanka, “Deep neural networks application in next-generation intelligent computer systems,” *Open semantic technologies for intelligent systems (OSTIS-2022)*, pp. 187–194, 2022.
[22] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, 1989.
[23] M. Mundt, F. Pahde, and K. Kersting, “Disentangling representations in reinforcement learning via symbolic reasoning,” *arXiv preprint*, 2024.
[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023. [Online]. Available: https://arxiv.org/abs/1706.03762
[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019. [Online]. Available: https://arxiv.org/abs/1810.04805
[26] T. B. Brown *et al.*, “Language models are few-shot learners,” 2020. [Online]. Available: https://arxiv.org/abs/2005.14165
[27] A. Chowdhery *et al.*, “Palm: Scaling language modeling with pathways,” 2022. [Online]. Available: https://arxiv.org/abs/2204.02311
[28] J. Hoffmann *et al.*, “Training compute-optimal large language models,” 2022. [Online]. Available: https://arxiv.org/abs/2203.15556
[29] S. Zhang *et al.*, “Opt: Open pre-trained transformer language models,” 2022. [Online]. Available: https://arxiv.org/abs/2205.01068
[30] T. L. Scao *et al.*, “Bloom: A 176b-parameter open-access multilingual language model,” 2023. [Online]. Available: https://arxiv.org/abs/2211.05100
[31] H. Touvron *et al.*, “Llama: Open and efficient foundation language models,” 2023. [Online]. Available: https://arxiv.org/abs/2302.13971
[32] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” 2021. [Online]. Available: https://arxiv.org/abs/2106.09685
[33] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient finetuning of quantized llms,” 2023. [Online]. Available: https://arxiv.org/abs/2305.14314
[34] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback,” 2022. [Online]. Available: https://arxiv.org/abs/2203.02155
[35] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” 2023. [Online]. Available: https://arxiv.org/abs/2201.11903
[36] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “React: Synergizing reasoning and acting in language models,” 2023. [Online]. Available: https://arxiv.org/abs/2210.03629
[37] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman, “GPQA: A Graduate-Level Google-Proof Q&A Benchmark,” 2023. [Online]. Available: https://arxiv.org/abs/2311.12022
[38] L. Phan *et al.*, “Humanity’s last exam,” 2025. [Online]. Available: https://arxiv.org/abs/2501.14249
[39] “Rankedagi – ai models ranked by latest benchmarks,” 2025. [Online]. Available: https://rankedagi.com
[40] P. Lewis *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” 2021. [Online]. Available: https://arxiv.org/abs/2005.11401
[41] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson, “From local to global: A graph rag approach to query-focused summarization,” 2025. [Online]. Available: https://arxiv.org/abs/2404.16130
[42] B. A. Kobrinsky, “Doverie k tekhnologiyam iskusstvennogo intellekta [trust in artificial intelligence technologies],” *Artificial Intelligence and Decision Making*, no. 3, pp. 3–17, 2024.
[43] I. Z. Batyrshin, A. O. Nedosekin, A. A. Stecko, V. B. Tarasov, A. V. Yazenin, and N. G. Yarushkina, *Nechetkie gibridnye sistemy. Teoriya i praktika [Fuzzy Hybrid Systems: Theory and Practice]*, N. G. Yarushkina, Ed. Moscow: FIZMATLIT, 2007.
[44] V. V. Golenkov, Ed., *Technology of Comprehensive Lifecycle Support for Semantically Compatible Next-Generation Intelligent Computer Systems*. Minsk: Bestprint, 2023.
[45] T. A. Gavrilova and V. F. Khoroshevskiy, *Bazy znaniy intellektual’nykh sistem [Knowledge Bases of Intelligent Systems: Textbook]*. Saint Petersburg: Piter, 2000.
[46] D. V. Shunkevich, “Agentno-orientirovannye reshayushchiki zadach intellektual’nykh sistem [agent-oriented problem solvers for intelligent systems],” diss. ... k-ta tekhn. nauk : 05.13.17 [Ph.D. dissertation in Technical Sciences : 05.13.17], Minsk, 2017, 254 l. [254 sheets].
[47] O. P. Kuznetsov, “O mashinnom obuchenii, mifakh o sil’nom iskusstvennom intellekte i o tom, chto takoe ponimanie [on machine learning, myths of strong artificial intelligence, and the nature of understanding],” *Ontology of Designing*, vol. 14, no. 4(54), pp. 466–482, 2024.

ПОДХОДЫ К НЕЙРО-СИМВОЛИЧЕСКОЙ ИНТЕГРАЦИИ: БОЛЬШИЕ ЯЗЫКОВЫЕ МОДЕЛИ И БАЗЫ ЗНАНИЙ

Головко В.А., Голенков В.В.,
Ковалев М.В., Крощенко А.А.,
Шункевич Д.В., Ивашенко В.П.

В статье рассматриваются современные подходы к интеграции нейросетевого и символьного искусственного интеллекта, включая архитектуры, использующие генеративные модели в сочетании с базами знаний и агентами. Особое внимание уделяется ограничениям больших языковых моделей (LLMs) при решении задач, связанных с долгосрочным контекстом, а также анализируются механизмы типа RAG и современные агентные системы. Предлагается архитектура интеллектуальных систем на базе Технологии OSTIS, в которой LLMs интегрированы в решатель задач. Обосновывается необходимость формального описания методов генеративного ИИ в составе решателя задач.

Авторы подчёркивают преимущества перехода базы знаний с позиции источника контекста для LLM в позицию общей семантической памяти, объединяющей различные виды знаний – от описания задач и моделей их решения до истории взаимодействия агентов. Такой подход обеспечивает автоматическую верификацию знаний, накопление опыта и прозрачность принимаемых решений.

Received 01.04.2025