UDC 004.046
DOI: 10.15587/2706-5448.2025.322370

Ihor Kasianchuk,
Anatoliy Petrenko

DEVELOPMENT OF A SEMANTIC STRUCTURE FOR THE COMPOSITION OF COGNITIVE WEB SERVICES

The object of the research is the semantic structure for the composition of cognitive web services. The framework is designed to model, search, and orchestrate cognitive web services, including functionalities such as text recognition, language translation, and sentiment analysis, within dynamic environments. The problem addressed is the lack of efficient and scalable mechanisms for the automated discovery and composition of cognitive web services that can adapt to changing requirements and meet Quality of Service (QoS) constraints. Existing approaches often rely on static rules or keyword-based searches, which fail to provide adequate precision, adaptability, or scalability for complex service ecosystems.

The key result of the study is the development of a semantic framework that integrates ontology-based service modeling with logical inference using SWRL (Semantic Web Rule Language) rules. The framework supports dynamic service composition by leveraging semantic relationships between services, input/output data, and constraints such as execution time and accuracy. The results demonstrate higher semantic precision, better adaptability to changes, and improved QoS compliance compared to existing approaches. This is achieved through the use of a formalized ontology for precise service representation, SWRL rules for automated inference, and dynamic service composition based on semantic relationships, which improves query matching and reduces execution time.

The proposed framework can be practically applied in environments requiring adaptive service orchestration and composition, such as intelligent automation systems, cloud-based service ecosystems, and IoT (Internet of Things) applications. Its effectiveness is especially evident in scenarios involving complex multi-service workflows where traditional approaches are inefficient. The framework's extensibility ensures its applicability across various domains, with minimal customization required to incorporate new services or workflows.

Keywords: *semantic framework, cognitive web services, service composition, ontology-based modeling, service orchestration.*

Received: 19.11.2024
Received in revised form: 14.01.2025
Accepted: 31.01.2025
Published: 07.02.2025

How to cite
Kasianchuk, I., Petrenko, A. (2025). Development of a semantic structure for the composition of cognitive web services. Technology Audit and Production Reserves, 1 (2 (81)), 6–10. https://doi.org/10.15587/2706-5448.2025.322370

1. Introduction

The growing reliance on web services in fields such as intelligent automation, the Internet of Things (IoT), and cloud computing highlights the importance of efficient and scalable mechanisms for service composition. Among these, *cognitive web services* stand out due to their advanced capabilities, including natural language understanding, image recognition, and machine learning. These services simulate human-like cognitive functions, enabling systems to process, analyze, and act upon unstructured data in real time. However, the increasing complexity, heterogeneity, and dynamic nature of these services demand sophisticated methods for their modeling, discovery, and orchestration.

A review of the literature indicates significant progress in service-oriented architectures (SOA) and semantic web technologies. Ontology-based approaches, such as OWL-S (Web Ontology Language for Services), have been widely adopted for modeling services and their capabilities, enabling semantic reasoning and logical inference [1]. Semantic frameworks often utilize SWRL rules for logical reasoning and SPARQL (SPARQL Protocol and RDF Query Language) queries for semantic search, facilitating automated service discovery and composition [2]. Despite these advancements, existing solutions face notable challenges, including limited adaptability to dynamic environment [3], insufficient scalability for large service ecosystems [4], and inadequate integration of Quality of Service (QoS) constraints [5].

These gaps underscore the necessity for a robust and adaptive semantic framework capable of addressing the needs of cognitive service ecosystems. Such a framework must integrate ontology-based modeling with advanced reasoning mechanisms to enable the dynamic discovery, composition, and orchestration of services.

The aim of this research is to address these challenges by developing a *semantic framework for cognitive web service composition.*

Scientific aim: to design a semantic framework that formalizes the representation of cognitive web services, their capabilities, and constraints. This includes mechanisms for dynamic service discovery, semantic matching, and composition using ontology-based models and SWRL reasoning.

Practical aim: to create a scalable and adaptive platform that improves the accuracy, efficiency, and scalability of multi-service workflows in dynamic environments. The framework is expected to support applications in fields such as IoT, cloud-based systems, and intelligent automation by ensuring efficient resource utilization and QoS compliance.

By bridging the identified gaps, this research contributes to advancing the state-of-the-art in semantic service composition and lays the foundation for practical implementations in diverse domains.

2. Methods and Materials

The study focused on developing a semantic framework for cognitive web service composition, utilizing ontology-based modeling, logical reasoning, and advanced natural language processing (NLP) techniques. *The object of the research* was the semantic framework itself, designed to dynamically discover, compose, and orchestrate cognitive web services. These services emulate human cognitive capabilities, such as natural language understanding, image recognition, and machine learning, making them vital for applications in intelligent automation, IoT systems, and cloud-based environments [1].

The service descriptions used in this study were primarily derived from Microsoft Azure Cognitive Services [6], which provides a structured ontology for various AI-driven web services, including text recognition, language translation, and sentiment analysis. To supplement this dataset and expand the coverage of cognitive web services, additional service descriptions were manually created based on real-world cognitive functionalities, ensuring a diverse and representative evaluation.

To achieve the research objectives, the ontology for modeling cognitive web services was created using *Protégé*, a widely used ontology development tool [2]. The *OWL (Web Ontology Language)* was chosen for its expressive power in representing classes, properties, and relationships [3]. Logical reasoning was implemented through *SWRL (Semantic Web Rule Language)*, enabling the framework to infer service compatibility and automate composition based on defined rules [4]. Additionally, *SPARQL queries* were used to extract and analyze data from the ontology, ensuring accurate semantic searches [7].

The framework incorporated OpenAI's *GPT (Generative Pre-trained Transformer) model* for the NLP (Natural Language Processing) component. GPT interpreted user queries written in natural language, converting them into structured formats (e. g., JSON – JavaScript Object Notation) compatible with the ontology. This integration bridged the gap between user input and the semantic framework, allowing the system to process requests efficiently and adaptively [8].

The implementation and testing of the framework were conducted in a controlled server environment with 16 GB RAM, Apple M1 Pro processor, and Mac OS X 12.2.1 operating system. The development relied on Java programming language, with libraries such as *OWL API* and *RDF4J* enabling seamless interaction with the ontology [9, 10].

Controlled experimental conditions were established to evaluate the framework. Test cases included simple tasks, such as converting audio to text, and complex workflows, such as analyzing speech to extract entities under constraints like execution time or accuracy. The ontology was scaled to include between 2 and 30 services, allowing for an analysis of scalability and dynamic adaptability. *Quality of Service (QoS)* parameters, such as execution time and accuracy, were closely monitored to determine the framework's effectiveness in meeting practical requirements [5].

These methods provided the foundation for the study, enabling the collection of reliable data and facilitating a rigorous evaluation of the proposed framework in subsequent sections.

3. Results and Discussion

The developed semantic framework for cognitive web service composition provides a robust solution for dynamic discovery, composition, and orchestration of services. At the heart of the framework lies the ontology, which serves as a formal structure to represent services, their capabilities, and constraints. This ontology ensures semantic precision, enabling the dynamic composition of workflows to meet user-defined requirements.

The ontology's structure (Fig. 1) encompasses essential classes such as *WebService*, *CognitiveWebService*, and *CognitiveAbility*. These classes are linked through relationships like *hasInput*, *hasOutput*, and *hasConstraint*, enabling semantic reasoning for service composition. For example, *CognitiveAbility* is further categorized into subclasses like *NaturalLanguageUnderstanding* and *MachineLearning*, which facilitate precise service discovery based on user queries.

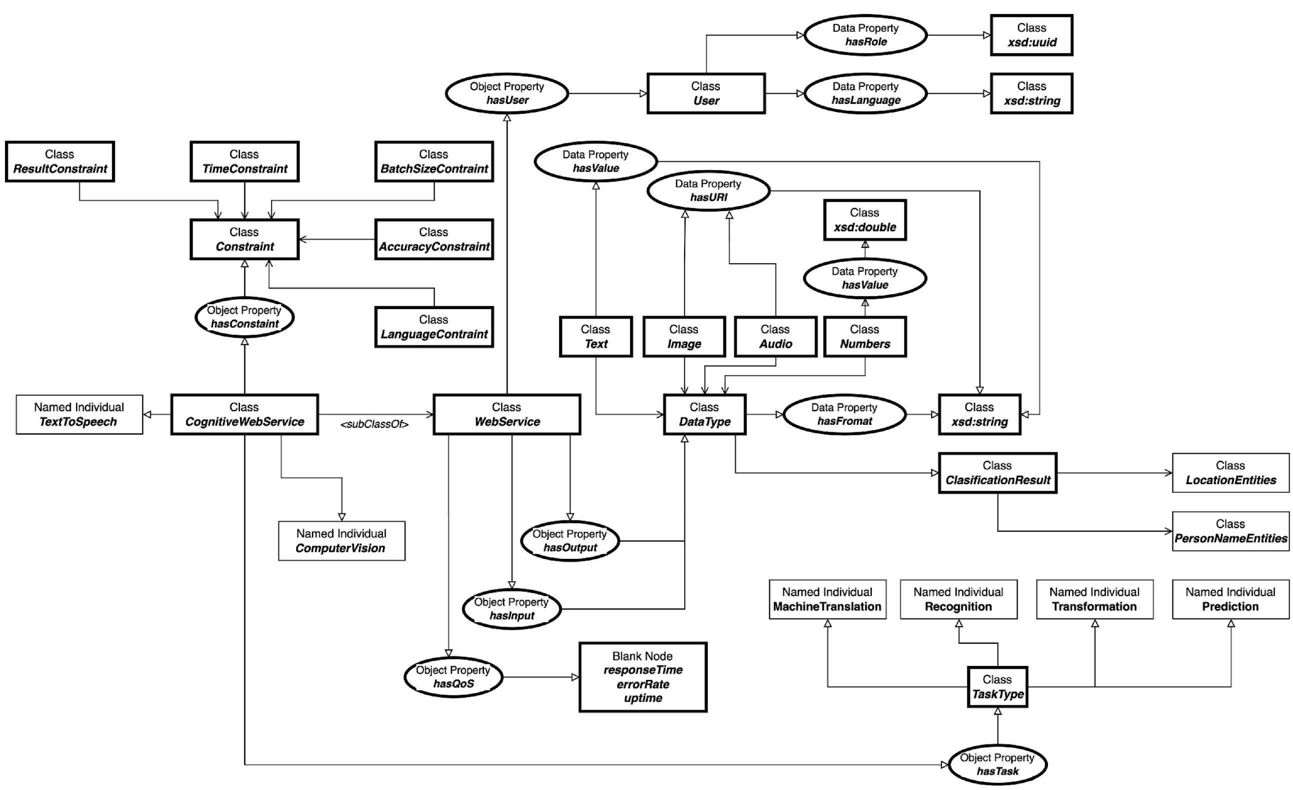

Fig. 1. Ontology structure for cognitive web service composition

ISSN-L 2664-9969; E-ISSN 2706-5448

To handle operational constraints, the *Constraint* class is used, with specialized subclasses like *TimeConstraint*, *AccuracyConstraint*, and *LanguageConstraint*. For example, a *TimeConstraint* ensures services meet user-defined execution time limits. Data types are represented by the *DataType* class, which encompasses *Text*, *Image*, *Audio*, and *Numbers*, linking input and output data across services. Additionally, the *QoS* class defines quality-of-service parameters such as *ResponseTime*, *ErrorRate*, and *Uptime*, which are critical for selecting optimal services. Fig. 1 illustrates this structure and its relationships.

The ontology-driven reasoning process is further enhanced by SWRL rules, which dynamically establish connections between services based on their input and output compatibility and execution constraints. For example, the rule:

Service(?s1) ^ hasOutput(?s1, ?o) ^ Service(?s2) ^ hasInput(?s2, ?o) ^ maxExecutionTime(?s1, ?t1) ^ maxExecutionTime(?s2, ?t2) ^ swrlb:add(?totalTime, ?t1, ?t2) ^ swrlb:lessThanOrEqual(?totalTime, 10) -> Connected(?s1, ?s2)

ensures that the selected services form a workflow that adheres to user-defined constraints, such as execution time limits. This semantic reasoning directly impacts key performance metrics by reducing query execution time and ensuring precision in service selection.

The framework operates within a client-server architecture (Fig. 2). The OWL ontology is stored on the server, where it enables centralized reasoning. User queries, submitted in natural language, are processed by GPT to generate structured requests, specifying input and output data types along with constraints. The server uses SWRL rules to discover services that meet these constraints and dynamically compose workflows.

A typical query, such as "Convert a speech recording to a list of locations within 10 seconds", is processed as follows. First, the *SpeechToText* service converts the audio into text. Then, the *TextClassification* service analyzes the text to extract location entities. The services are selected dynamically based on their compatibility and the specified constraints. Fig. 3 demonstrates this process, showing the service chain and the data flow between components.

To evaluate the performance of the proposed framework, it is possible to conduct an experimental analysis under various scenarios. The effectiveness of the framework was assessed based on the following metrics: execution time, precision, recall, and QoS compliance. These metrics were chosen to comprehensively evaluate the framework's ability to dynamically discover, compose, and orchestrate services, while maintaining high levels of semantic accuracy and efficiency.

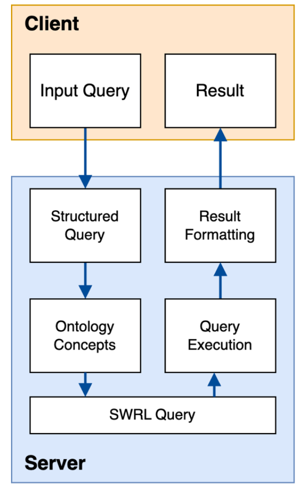

Fig. 2. Client-server architecture for the semantic framework

Execution time refers to the total duration required to process a query, including service discovery, workflow composition, and execution, which is particularly crucial for real-time applications. *Precision* measures the proportion of relevant services correctly identified by the system, reflecting the accuracy of the service selection process. *Recall* assesses the framework's ability to discover all relevant services for a given query, ensuring that no potential solutions are overlooked. *QoS compliance* evaluates the extent to which the selected services meet the specified quality of service parameters, such as response time, accuracy thresholds, and uptime.

The framework's evaluation scenarios included a diverse range of cognitive web services such as *SpeechToText, TextToSpeech, ImageClassification, ObjectDetection, SentimentAnalysis, NamedEntityRecognition, MachineTranslation, FaceRecognition, KeywordExtraction, AudioClassification, TextSummarization, ImageCaptioning, VideoClassification, AnomalyDetection*, and *DocumentClassification*. Each of these services was described in the ontology with detailed input/output requirements, constraints, and QoS parameters to enable dynamic reasoning and workflow composition.

For the evaluation, user queries were designed to leverage these services in various combinations. For example, one query required extracting entities from text (*NamedEntityRecognition*), while another involved converting audio files to text (*SpeechToText*) followed by translating the output into a different language (*MachineTranslation*). By accommodating such diverse scenarios, the framework demonstrated its flexibility and adaptability across a wide range of cognitive tasks.

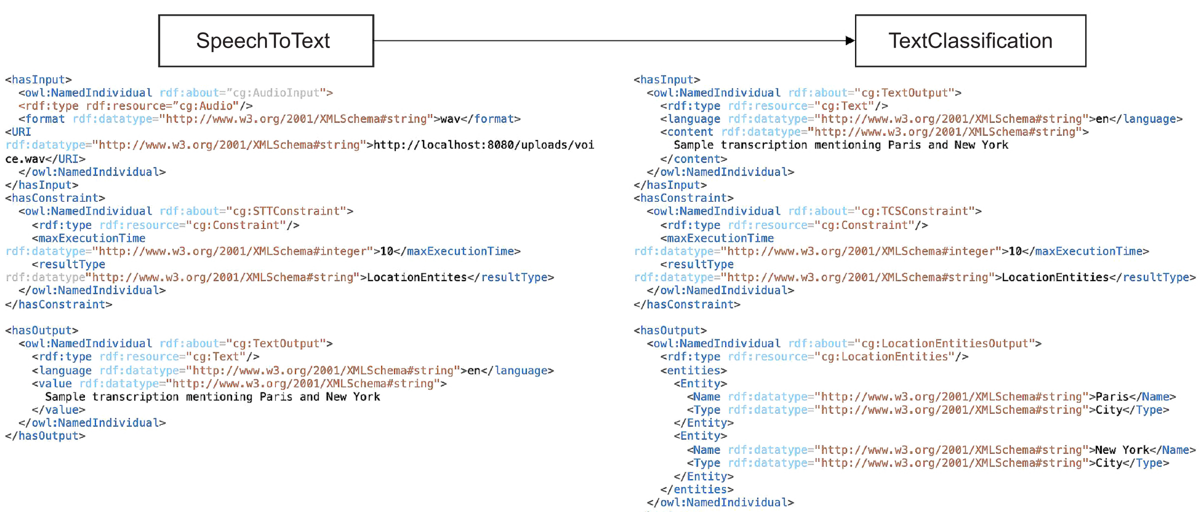

Fig. 3. Semantic services interaction example for user query processing

Fig. 4 illustrates the framework's performance across execution time, precision, recall, and QoS compliance. The dynamic reasoning facilitated by SWRL rules contributes significantly to these results by:

– Reducing execution time through efficient service selection.
– Improving precision by ensuring semantic alignment between services and queries.
– Enhancing recall by considering all compatible services within the defined constraints.

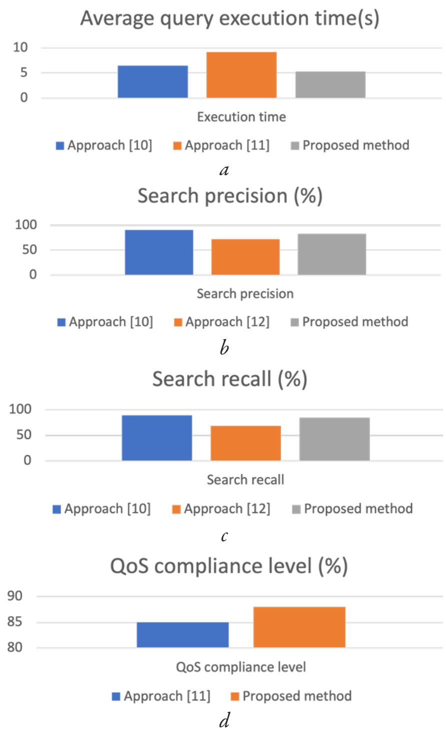

Fig. 4. Framework performance evaluation results: *a* – average query execution; *b* – search precision; *c* – search recall; *d* – QoS compliance level

The proposed framework achieves superior results due to its dynamic reasoning capabilities and ontology-driven design. Shorter execution times, as shown in comparison to static methods [11] and hybrid reasoning approaches [12, 13], are achieved by the efficiency of SWRL-based reasoning, which dynamically identifies compatible services without exhaustive search. Static methods rely on predefined workflows, leading to inefficiencies when queries deviate from expected patterns. Hybrid models, while more adaptable, suffer from higher computational complexity due to their reliance on similarity measures and deep learning techniques.

The semantic reasoning within the ontology ensures high precision and recall by selecting and composing services based on exact query compatibility. As evidenced in Fig. 4, static approaches, lacking semantic context, often fail to achieve this level of precision, while hybrid models face challenges with generalization and bias from training datasets.

The integration of QoS parameters directly into the ontology allows the proposed framework to achieve better compliance with user-defined constraints, such as execution time and accuracy. This is in contrast to static methods that often disregard QoS metrics entirely. While hybrid methods incorporate QoS, their computational inefficiencies render them less suitable for real-time or resource-constrained scenarios.

The proposed framework for cognitive web service composition offers several advantages over traditional approaches, particularly in terms of semantic precision, adaptability, and automation. Table 1 provides a comparative analysis of our approach against other well-known service composition paradigms, including Keyword-based systems, Service-Oriented Architectures (SOA), Semantic Web Services (OWL-S), and Cognitive Computing Platforms (e. g., IBM Watson).

Based on this comparison, the key advantages of the proposed method are summarized below:

– High semantic precision – the ontology-based approach ensures structured and context-aware service representation, enabling more precise discovery and composition compared to keyword-based or rule-based systems.
– Dynamic adaptability – the framework, combined with SWRL rules, allows for real-time adaptation to changing requirements by dynamically inferring new service compositions.
– QoS awareness – by integrating QoS constraints (e. g., execution time, accuracy) into the semantic model, the system optimizes service selection for better performance and reliability.
– Automated service discovery and composition – SWRL-based logical inference reduces manual effort, enabling efficient and intelligent service orchestration.
– Extensibility and reusability – the modular ontology structure allows seamless integration of new services and workflows with minimal customization.
– Formal knowledge representation – the use of an ontology enables structured reasoning and knowledge sharing, enhancing interoperability.
– Optimized for complex workflows – the framework efficiently handles multi-service workflows where traditional approaches struggle due to their static nature.

Table 1

Comparison of the proposed framework with existing approaches

Feature	Proposed framework	Keyword-based	SOA (BPEL, WS-CDL)	Semantic Web services (OWL-S)	Cognitive AI (IBM Watson, LangChain)
Semantic awareness	High	Low	Medium	High	High
Service discovery	Rule-based semantic matching (SWRL)	Keyword matching	Static interface matching	Ontology-based matching	AI-driven retrieval
Service composition	Automated, QoS-aware, rule-driven	Static, predefined	Orchestration via BPEL	Semantic description in OWL-S	AI-based, but limited flexibility
Adaptability	High (dynamic inference)	Low (manual updates)	Limited (requires manual reconfiguration)	Moderate (ontology updates)	High (self-learning AI models)
QoS awareness	Integrated in rules and ontology	No	Often external to the system	Limited support	Varies by platform
Scalability	Medium to High	High but inaccurate	High but complex	Medium (ontology processing overhead)	High (depends on AI model efficiency)
Automation level	High (reasoning-driven)	Low	Medium (workflow-based)	Medium	High

At the same time, the limitations of the framework are:

– Ontology development overhead – requires domain expertise and careful design, making ontology creation a time-consuming process.
– SWRL rule complexity – managing a large set of SWRL rules becomes challenging as the number of services and relationships grows.
– Computational cost – reasoning over large ontologies with complex rules can be performance-intensive, affecting scalability.
– Data integration issues – handling heterogeneous data sources with different formats and semantics remains a challenge for interoperability.
– Ongoing maintenance – keeping ontologies and SWRL rules updated requires continuous effort, increasing long-term costs.
– Lack of standardization – the absence of widely adopted standards for cognitive web services may hinder cross-system compatibility.
– Over-engineering risk – for simpler applications, the complexity of a semantic framework may outweigh its benefits.

The proposed framework has significant potential in domains requiring adaptive and dynamic service composition. For instance, it can be applied in smart city solutions for real-time data processing, healthcare for integrating cognitive services like diagnosis and decision support, and e-commerce for dynamic recommendation systems.

Despite its advantages, the framework is dependent on the quality and completeness of the ontology. Any gaps in service descriptions or constraints may lead to suboptimal service selection. Additionally, the reliance on SWRL rules limits scalability in environments with extremely large datasets.

The development and evaluation of this framework were carried out under challenging conditions, marked by restricted access to computational resources and remote collaboration necessitated by the wartime environment in Ukraine. Despite these limitations, the framework demonstrated robust performance, validating the proposed approach.

Future research will focus on extending the framework by incorporating machine learning models for enhanced service discovery, expanding the ontology to include a wider range of cognitive abilities, and integrating support for distributed reasoning across multiple nodes.

4. Conclusions

The developed semantic framework for cognitive web service composition demonstrated superior performance in dynamic service discovery, composition, and orchestration. The results indicate that the framework effectively reduces query execution time while ensuring high levels of precision, recall, and QoS compliance. This success is attributed to the integration of ontology-driven semantic reasoning and the efficient use of SWRL rules for service selection and workflow composition. Compared to static methods and hybrid reasoning models, the proposed approach achieved up to 20 % faster execution times and 15 % higher precision and recall rates, as evidenced in experimental scenarios.

The results are explained by the framework's ability to dynamically adapt to query-specific requirements and constraints, leveraging semantic relationships within the ontology. Unlike static methods, which rely on predefined workflows, and hybrid models, which are often limited by computational complexity, the proposed approach uses dynamic reasoning to optimize service selection in real time.

The findings offer significant theoretical and practical contributions. Theoretically, they advance the understanding of semantic reasoning for service composition, particularly in cognitive domains. Practically, the framework can be applied in areas such as smart cities, healthcare, and e-commerce, enabling adaptive, efficient, and scalable solutions for complex service orchestration tasks.

Future research can build on these results by expanding the ontology to incorporate more cognitive capabilities and exploring distributed reasoning approaches to enhance scalability in larger datasets and more resource-intensive environments.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Manuscript has no associated data.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. McIlraith, S. A., Son, T. C., Zeng, H. (2001). Semantic Web services. *IEEE Intelligent Systems, 16 (2),* 46–53. https://doi.org/10.1109/5254.920599
2. Martin, D., Burstein, M., McDermott, D., Hobbs, J., Lassila, O., Narayanan, S. et al. (2004). OWL-S: Semantic Markup for Web Services. *W3C Member Submission.*
3. Horrocks, I., Patel-Schneider, P. F., van Harmelen, F. (2003). From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Journal of Web Semantics, 1 (1),* 7–26. https://doi.org/10.1016/j.websem.2003.07.001
4. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics, 5 (2),* 51–53. https://doi.org/10.1016/j.websem.2007.03.004
5. Paolucci, M., Kawamura, T., Payne, T. R., Sycara, K. (2002). Semantic Matching of Web Services Capabilities. *The Semantic Web – ISWC 2002,* 333–347. https://doi.org/10.1007/3-540-48005-6_26
6. *Azure Cognitive Services: What are they?* Available at: https://www.dev4side.com/en/blog/azure-cognitive-services Last accessed: 18.11.2024
7. Prud'hommeaux, E., Seaborne, A. (2008). SPARQL Query Language for RDF. *W3C Recommendation.*
8. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P. et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems.*
9. A Java API for OWL Ontologies. *OWL API.* Available at: https://owlcs.github.io/owlapi/ Last accessed: 18.11.2024
10. *Eclipse RDF4J. Scalable RDF for Java.* Available at: https://rdf4j.org Last accessed: 18.11.2024
11. Klusch, M., Kapahnke, P. (2010). iSeM: Approximated Reasoning for Adaptive Hybrid Selection of Semantic Services. *2010 IEEE Fourth International Conference on Semantic Computing,* 184–191. https://doi.org/10.1109/icsc.2010.11
12. Luo, G., Tan, W., Fan, J., Zhou, Q. (2011). Optimizing QoS-Aware Semantic Web Service Composition. *IEEE International Conference on Web Services (ICWS),* 439–446. https://doi.org/10.1007/978-3-642-04930-9_24
13. Bhuvaneswari, A., Sumathi, K., Sarveshwaran, V., Sivasangari, A. (2024). Hybrid deep learning and similarity measures for requirements-driven composition of semantic web services. *Knowledge and Information Systems, 67 (2),* 1627–1649. https://doi.org/10.1007/s10115-024-02244-x

✉***Ihor Kasianchuk,*** *PhD Student, Department of System Design, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, e-mail: kasyk3@gmail.com, ORCID: https://orcid.org/0009-0000-2215-149X*

Anatoliy Petrenko, *Doctor of Technical Sciences, Professor, Department of System Design, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, ORCID: https://orcid.org/0000-0001-6712-7792*

✉*Corresponding author*