Ontology control system in Neo4j

Serhii Vladov[1,†], Anna Bakurova[2,*,†], Ivan Omelianenko[3,*,†], Ruslan Sirenko[4,*,†], Elina Tereschenko[5,*,†]

[1] *Kharkiv National University of Internal Affairs, Kharkiv, Ukraine*
[2,3,4,5] *National University «Zaporizhzhia Polytechnic», Zaporizhzhia, Ukraine*

Abstract
The paper presents the results of a study on the integration of ontologies dedicated to sunflower cultivation in Ukraine and located on pan-European resources using the Neo4j graph database. This research is a component of the task of integrating the subject ontology of sunflower cultivation in Ukraine into the pan-European community, a task outlined by the authors in previous works. The paper introduces an ontology control system in Neo4j, which is part of the task to integrate the subject ontology of sunflower agrotechnology in Ukraine with the pan-European community's resources. The main advantage of this system is its ability to manage ontology models without requiring in-depth knowledge of query languages for graph databases. The ontology control system in Neo4j allows users to load, store, import, and merge ontologies in the Neo4j graph database, with an intuitive user interface. Further development of the ontology control system based on the Neo4j graph database aims to develop effective algorithms for ontology matching, merging, and integration.

Keywords
Ontology, Ontology integration, Ontology merging, Graph database, Control System

1. Introduction

In previous works of the authors, the task of integrating data on sunflower research obtained in scientific research laboratories of Ukraine into international databases was set, which requires a strategic approach that will include the integration of scientific results, data standardization and the use of modern technologies for their management [1]. The presented work is devoted to the integration of subject ontologies of sunflower cultivation agrotechnology [1], which were created by the authors in close cooperation with specialists of the Institute of Agricultural Sciences of the National Academy of Sciences of Ukraine, with pan-European resources containing information in this field.

In international practice, there is a certain list of ontologies containing information about sunflower. PLANT Ontology [2] includes terms for describing morphological and physiological characteristics of plants. Crop Ontology [3] contains information about sunflower, and describes genetics, agronomy, pest resistance, yield characteristics, agronomic factors. Sunflower Ontology is the Crop Ontology part with number 359 contains 351 records about sunflower phenotypes [4]. Sunflower Genome Database (SunGene) [5] contains data on genetic markers and variants of sunflower genomes, as well as their interaction with agronomic characteristics. Gene Ontology (GO) [6] is a general ontology for biological data and contains terms that can be applied to the study of genes related to sunflower. Plant Trait Ontology (PTO) [7] contains data about sunflower, describes plant characteristics regarding disease resistance, development and yield.

MoMLeT-2025: 7th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine
*Corresponding author.
† These authors contributed equally.
serhii.vladov@univd.edu.ua (S. Vladov); abaka111060@gmail.com (A.Bakurova); ivan.omelyanenko0@gmail.com (I. Omelianenko); skriler222000@gmail.com (R. Sirenko); elina_vt@ukr.net(E. Tereschenko)
0000-0001-8009-5254 (S. Vladov); 0000-0001-6986-3769 (A.Bakurova); 0009-0009-0885-2957 (I. Omelianenko); 0009-0009-2767-8296 (R. Sirenko); 0000-0001-6207-8071 (E. Tereschenko)

HeliantHOME [8,9] is a public database focused on phenotypic data of various sunflower species, particularly Helianthus annuus. The unification of ontologies allows for the unification of knowledge, creating a single system of terms and concepts, which simplifies their understanding and exchange between scientists and specialists from different countries. This is important for the development of international standards for quality control, protection of plant variety rights and determination of their properties.

The combined ontology stores data from different countries, allowing breeders and agronomists from different regions to work with a single knowledge base. This can contribute to faster data exchange between countries, as well as accelerating scientific research and the implementation of innovative approaches in breeding. This contributes to improving international cooperation in the field of agronomy and biotechnology, and also makes it possible to use the experience of other countries to improve breeding programs. Plant breeding is a global process, and combining ontologies allows you to study global trends in breeding, for example, the development of new varieties resistant to climate change or pests.

Shared access to knowledge from different countries allows for an increase in the number of genetic resources used in breeding to adapt plant varieties to different climatic and environmental conditions, which reduces risks for agriculture.

The aim of the work is to create a control system for managing subject ontologies based on graph databases. The main goal of creating the control system is to provide the ability to manage ontological models without the need for in-depth knowledge of query languages for graph databases.

2. Related works

According to [10], to ensure the integrity, compatibility and effective use of different ontologies, to preserve their semantic accuracy and to support interoperability between systems, the tasks of matching, merging and restoration are solved when combining ontologies. Ontology Matching is the process of finding correspondences between concepts (terms) of different ontologies, which ensures semantic compatibility of data presented in different systems. A number of problems arise when matching. In the survey [11], a systematic review of publications that solve the problems of Ontology Matching that have been presented within the years 2004–2021 was conducted. Difference in terms, where the same concept may be represented by different terms in different ontologies. Fixed Different levels of abstraction, where concepts in one ontology may be more general or more detailed. A term that has one meaning in one ontology may have a different meaning in another, depending on the context of its use.

For ontology comparison, syntactic approaches are used to compare terms based on their structure, such as using text strings or searching for similarities between class names and properties. Lexical methods are based on comparing terms using dictionaries or thesauri [12]. Semantic methods are based on analyzing the meaning of a term in the context of the ontology [13]. Also, machine learning-based methods are used to detect similarities between classes, which can take into account context and complex relationships between concepts [14].

Ontology merging is the process of combining two or more ontologies into a single, common ontology while preserving all the important information from each. This allows for a more complete and consistent knowledge model that combines the advantages of several different ontologies [10]. During the merging process, conflicts may arise between concepts that are the same in different ontologies but are represented differently. There may be duplicate terms or properties, which requires a decision as to which ones should be kept and which ones should be removed. Merging a large number of ontologies can lead to performance and processing efficiency problems.

A rule-based ontology merging method for integrating concepts and terms to avoid duplication and contradictions [15]. Automated ontology merging using algorithms that automatically process mapped concepts and generate a single merged ontology [16]. Hybrid methods are a combination of different approaches, such as machine learning and proof theories, to automatically and manually resolve conflicts during merging [17, 18].

Ontology recovery is the process of restoring or reconstructing an ontology after it has been lost, damaged, or obsolete [10]. It may also include the process of improving or updating the ontology to ensure its relevance and accuracy. Problems encountered in recovery are related to incompleteness, if part of the knowledge has been lost or not preserved; Obsolescence, recovery may require taking into account new knowledge and changes in the field. Ontology recovery should ensure compatibility with other existing ontologies and systems.

The method of ontology restoration is Reconstruction through data analysis: Using existing data or ontologies to automatically create a new ontology. Analyzing existing ontologies by searching for similar or analogous concepts. Updating based on external sources, using up-to-date knowledge bases or updated standards to improve the restored ontology [19].

Ontology mapping, merging, and restoration are critical processes for knowledge integration and management in complex and distributed information systems. They ensure semantic compatibility, avoid duplication of information, and keep ontologies up-to-date. However, each of these processes faces challenges associated with different approaches, data representations, and changes in knowledge, so their successful implementation requires the development of effective algorithms and tools, as well as the harmonization of standards.

The operations of comparison, merging are the most expensive and ambiguous due to the fact that it is necessary to compare large sets of structured data according to different criteria (semantic, structural or other). There are software tools, for example Chimaera [20], OntoMerge [21], FCA-Merge – Formal Concept Analysis and Merging [22], Protege [23] , which have their own characteristics and application conditions.

3. Our Approach

Ontology is a form of organizing the representation of concepts of a certain field of knowledge [24]. Ontology is an ordered triple of the form: O=(O,R,F), where T is a finite non-empty set of terms of the field of knowledge, which is represented by ontology O; R is a finite set of relations between concepts of the field of knowledge; F is a finite set of interpretation functions specified on the concepts and/or relations of ontology O. An ontology, which is created for further use in decision support systems, is an important tool for ensuring effective knowledge management in complex systems that are actively developing. The main purpose of such an ontology is to preserve embedded knowledge, as well as the ability to form new knowledge based on user requests [25]. In the context of decision support systems, an ontology organizes various concepts and their relationships into a logically organized structure, which allows users to quickly find the necessary information [26, 27, 28]. Ontology provides consistency and compatibility of knowledge between different systems or subsystems, which is important for integrating different data sources. The main one is the ability to automate the formulation of new knowledge. Using logic and other knowledge processing methods, ontology can generate new knowledge based on existing data, which makes decision support systems more effective.

Let us consider the possibilities of expanding existing ontologies dedicated to sunflower cultivation through integration. As a source, we will take the ontologies developed by the authors Helianthus Ontology& Helianthus phenotype (seed & head) (Fig.1,2), the stages of development are described in [1]. The ontologies were built using Protege [23]. The work was carried out with the involvement of industry experts, researchers of the Institute of Oilseed Crops of the National Academy of Sciences of Ukraine. When building the ontology, the experience of national experts in breeding, genetics and agrotechnology of sunflower cultivation, as well as state standards of Ukraine [29], was used. Another ontology that we consider as a source is the Sunflower Ontology [4] from the European resource the Crop Ontology [3].

The basis for the merger is the following rule: merged ontology is O'=O1∪O2=(O1−O2)∪(O2−O1)∪(O1∩O2) [30]. With three input ontologies, the merged ontology O" can be viewed as the union of O1, O2 and O3:O" = O1 ∪ O2 ∪ O3 = (O1 − O2) ∪ (O2 − O1) ∪ (O1 ∩ O2) ∪ ((O3 − O1) ∩ (O3 − O2)).

We choose semantic methods of ontology matching and automated merging of ontologies using algorithms that automatically process matched concepts and generate a single unified ontology [10]. The semantic method of matching was chosen due to the content component of the task of integrating ontologies dedicated to sunflower growing agrotechnology. It is the use of terminology in the field of sunflower growing agrotechnology to introduce ontology classes that makes it possible to use semantic methods.

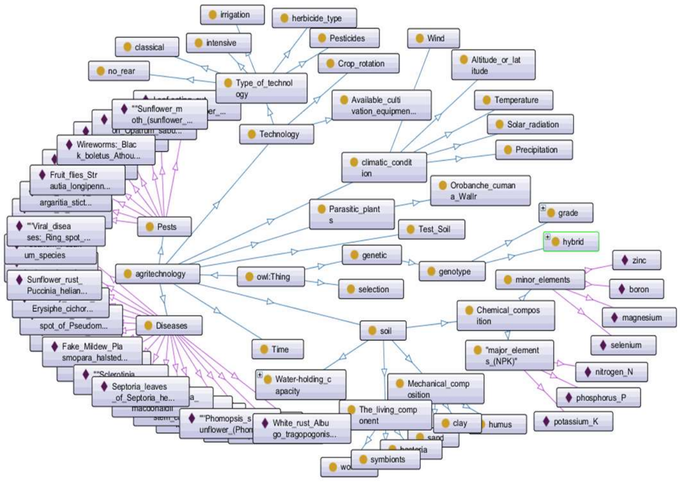

Figure 1: Ontology graph the Helianthus Ontology(the part).

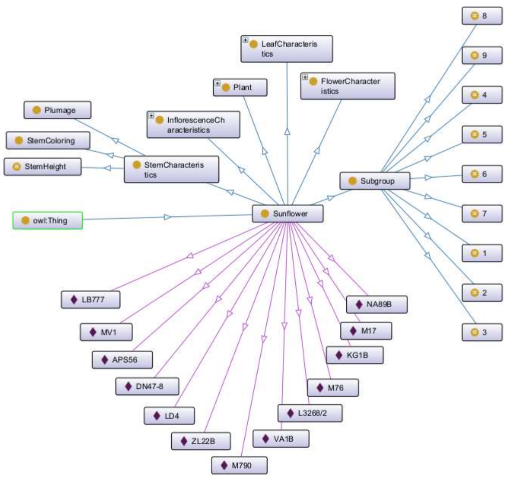

Figure 2: Ontology graph the Helianthus phenotype (seed & head) (the part).

We choose semantic methods of ontology matching and automated merging of ontologies using algorithms that automatically process matched concepts and generate a single unified ontology [10]. The semantic method of matching was chosen due to the content component of the task of integrating ontologies dedicated to sunflower growing agrotechnology. It is the use of terminology in the field of sunflower growing agrotechnology to introduce ontology classes that makes it possible to use semantic methods.

Our approach is based on the integration of multiple ontologies using a graph database. Exporting an ontology to a graph database has several important advantages, as graph databases are well suited for representing complex, interconnected data. Graph databases facilitate the integration of data from different sources. They allow information from different sources to be combined and stored, even if the data has different formats or structures, which is important for ontologies that can integrate knowledge from different domains.

Ontologies have a complex structure with numerous relationships between different concepts, notions, and classes. A graph database allows for efficient storage of these relationships, since graphs directly reflect relationships between elements through vertices and edges [31]. In graph databases, queries related to finding relationships and navigating graphs (for example, finding paths between nodes, proximity analysis, etc.) are executed faster. This allows for efficient work with ontologies, where it is important to infer and analyze complex relationships. In the article [32], it is concluded that graph databases provide faster response times when querying data with spatial-temporal constraints in real-world conditions, using examples such as the open graph database JanusGraph, the object-relational database PostgreSQL, and the enterprise graph analytics platform TigerGraph. Graph databases provide more flexibility for making changes to the data model, since adding new classes or properties to the ontology can be implemented without significant changes to the database structure. Graph databases allow you to perform complex queries on the relationships between objects without having to use complex joins, as in relational databases. This can be important for ontologies, since it is often necessary to perform queries at different levels of abstraction. Thanks to the graph structure, ontologies can be conveniently visualized, which allows you to better understand and analyze the relationships between concepts, properties and classes. As a result, exporting an ontology to a graph database allows you to efficiently store, query and visualize complex knowledge and its relationships, which is useful for research, data analysis, integration and processing of knowledge in various fields, as well as for the generation of new knowledge. The study showed that the graph database Neo4j is the most efficient for processing data with a large number of complex relationships [33]. The graph database Neo4j is used for storing semantic relationships between data, as well as for navigation and visualization of integrated resources, particularly during the computational-analytical stage. By using the Cypher query language, the Neo4j graph database ensures high performance when working with large graphs.

4. Experiments

Creating an ontology control system in Neo4j is part of the task of integrating the subject ontology of sunflower cultivation agrotechnology in Ukraine into the pan-European community.

Neo4j demonstrates clear advantages over Protégé in the context of integration with real-world systems. It offers official drivers and SDKs for multiple programming languages, enabling seamless embedding into web services, backend applications, and microservice architectures. Its powerful Cypher query language, combined with real-time data access and update capabilities, supports dynamic environments that require rapid interaction with complex graph structures. Unlike Protégé, which serves primarily as a local ontology modeling tool without runtime functionality or cloud-ready deployment support, Neo4j is production-ready, scalable, and compatible with modern DevOps practices. Moreover, its rich ecosystem of connectors (e.g., Kafka, GraphQL, BI tools) further facilitates integration into enterprise technology stacks.

The main goal of the system is to provide an interactive, researcher-friendly interface for managing ontology models without the need for in-depth knowledge of graph database query languages. The ontology control system in Neo4j allows you to load, store, import, and merge ontologies in the Neo4j graph database. Further development of the ontology control system based on the Neo4j graph database aims to develop effective algorithms for matching, merging, and integrating subject ontologies.

Currently, the ontology control system in Neo4j provides the function of integrating ontologies using semantic matching methods and automated merging of ontologies [10]. Currently, among the first matching options, the semantic matching method using the internal Neo4j tool *apoc.text.sorensenDiceSimilarityWithLanguage*, which returns Sørensen–Dice coefficient [33]. Mathematically, the formula for calculating the Sørensen–Dice coefficient for two strings string1, string2 is

$$D(string1, string2) = \frac{2|string1 \cap string2|}{|string1| + |string2|},$$

where $|string1 \cap string2|$ is the common items between $\mathrm{string1}, \mathrm{string2}$.

This method of comparison was applied in view of the practical task of integrating ontologies dedicated to agrotechnology of sunflower cultivation. Preliminary analysis demonstrated the similarity/coincidence of class names, which correspond to the terminology adopted in the field.

The fragment of the program code for determining node name similarity via *apoc.text.sorensenDice*.

Compare all :Class pairs of nodes.

Merge nodes if similarity > threshold (default 0.85).

```
"""Merge similar Class nodes if similarity > threshold"""
count_before_query = "MATCH (n:Class) RETURN count(n) AS count"
merge_query = f"""
MATCH (a:Class), (b:Class)
WHERE id(a) < id(b)
AND apoc.text.sorensenDice(a.name, b.name) >= {threshold}
CALL apoc.refactor.mergeNodes([a, b], {{properties: 'combine'}}) YIELD node
RETURN count(node) AS merged
"""
```

The selection of candidates for merging is performed in the following order

1. All nodes that have the owl__Class label or have the rdfs__subClassOf relationship are selected.

For each node, a label is defined for comparison. It can be:

name

or rdfs__label, obo__label, skos__prefLabel, uri — if name is missing.

The goal is to find the most useful text information to identify the node, even if the RDF format is slightly different.

2. Similarity calculation

Pairs of nodes are compared in which id(a) < id(b) (to avoid repetitions).

For each pair, Sorensen-Dice similarity is calculated — a method based on bigrams (two characters in a row).

For example: "seed_size" → ["se", "ee", "ed", "_s", "si", "iz", "ze"]

If similarity ≥ threshold (e.g. 0.85), they are considered to be the same concepts.

3. Pair merging

Nodes with sufficient similarity are merged using:

cypher

Copy

Edit

```
CALL apoc.refactor.mergeNodes([a,b], {properties:'combine'})
```

This means:

All connections that these nodes had are merged.

Attributes are merged (according to the 'combine' setting).

4. Result :After completion:

The number of merged pairs (merged_classes) is returned.

And the total number of classes after merging (total_classes).

Advantages of this approach are

1. Does not depend on the strict name of the labels (takes into account owl__Class, rdfs__subClassOf, different labels).
2. Works even if you import RDF from another domain area (CO, SKOS, OBO...).
3. You can set the similarity threshold.

The experiments were conducted on Helianthus Ontology& Helianthus phenotype (seed & head) [1] and Sunflower Ontology [4](Fig3, Fig4).

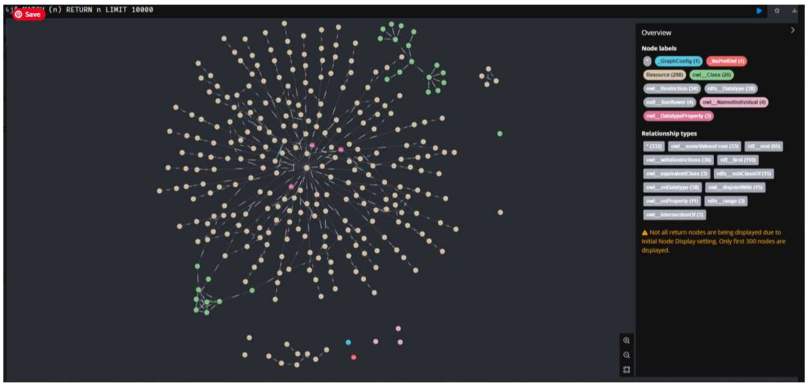

Figure 3: The first source ontology O1 is Helianthus Ontology.

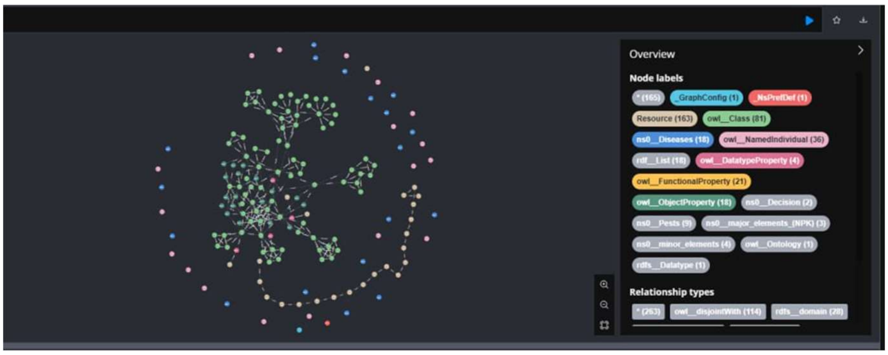

Figure 4: The second source ontology O2 is Helianthus phenotype (seed & head).

In the first step, an empty graph is created that contains one vertex (Fig.5).

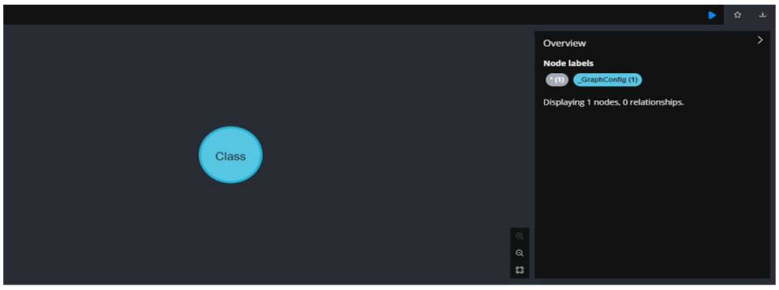

Figure 5: The first step of merging two ontologies with Neo4j.

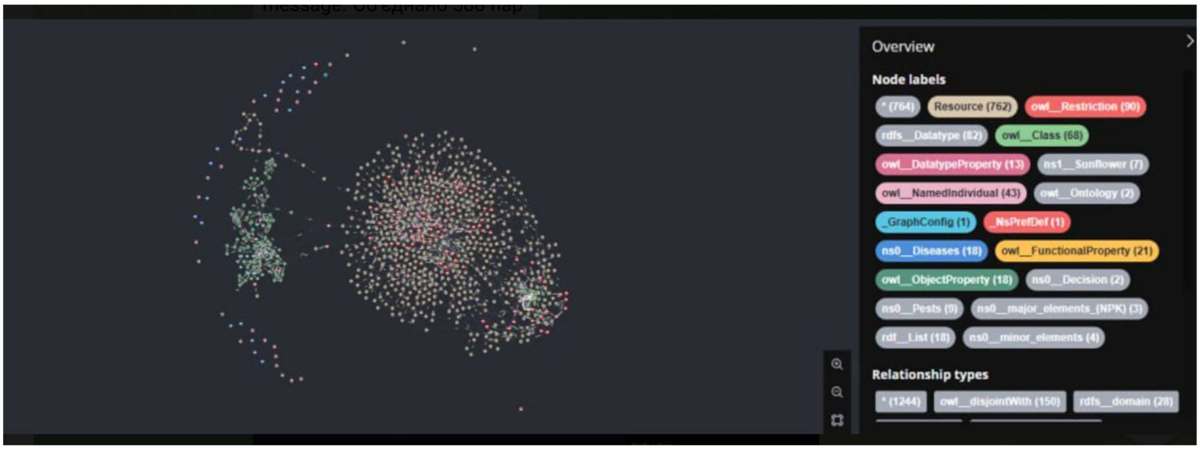

Figure 6: The Target ontology O' is merged ontology with Helianthus Ontology & Helianthus phenotype (seed & head).

Result message:
{merged_classes:588
message:" 588 pairs combined "
total_classes:68
}

The second experiment is integration ontology O' and Sunflower Ontology in the Crop Ontology (EU)).

The next source ontology is Sunflower Ontology in the Crop Ontology (EU). The result of export Sunflower Ontology in the Crop Ontology (EU) to Neo4j is presented in **Figure 7.**

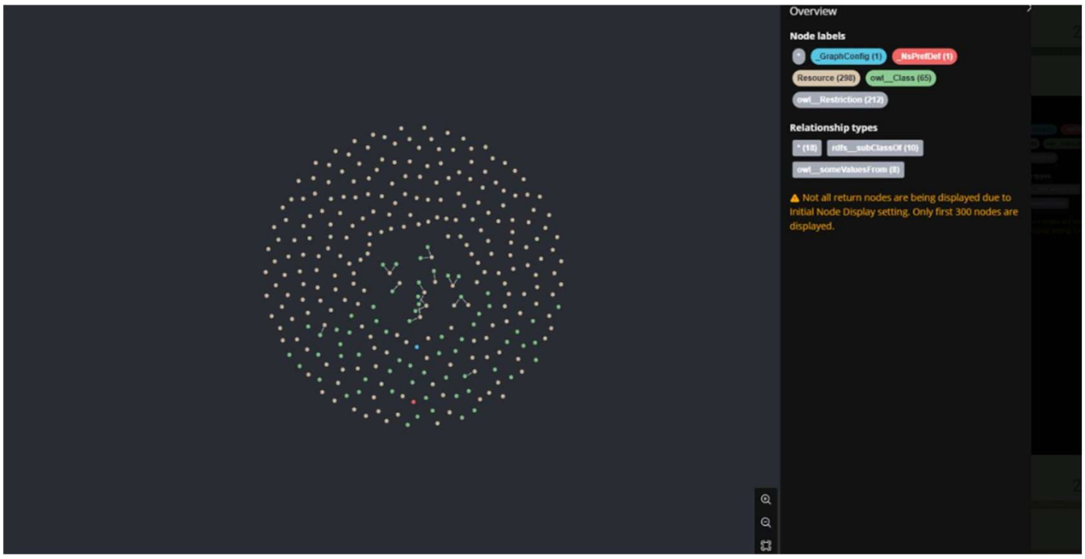

Figure 7: The third source ontology is Sunflower Ontology in the Crop Ontology (EU).

Helianthus EU Ontology is the Target ontology O'' that is merged ontology with Helianthus Ontology & Helianthus phenotype (seed & head)& Sunflower Ontology in the Crop Ontology (EU), it presented in Figure 8.

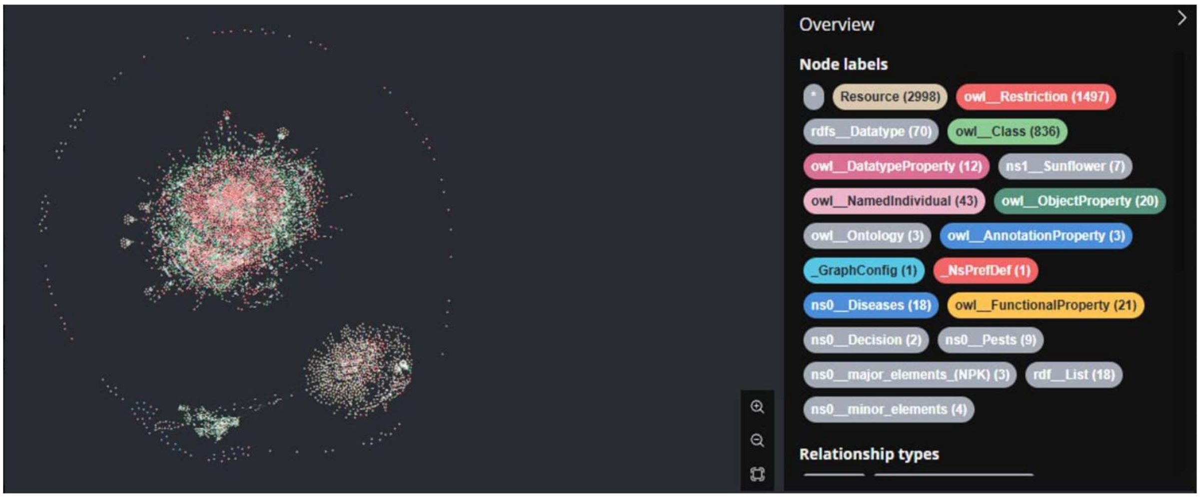

Figure 8: Helianthus EU Ontology.

Result message:
{merged_classes: 832
message:" 832 pairs combined "
total_classes:1183
}

The numerical characteristics and qualitative analysis demonstrate the correct operation of the matching and merging algorithms, which ensures high-quality integration in the target ontology Helianthus EU Ontology.

Frontend development is implemented based on JavaScript interface (Fig.9). Backend development is based on FastAPI using libraries for interaction with Neo4j.

Figure 9: User Interface.

The software architecture uses containerization using Docker, which allows for isolation of the work environment and simplifies system deployment. At the backend level, the system provides the ability to load ontologies in various formats, including RDF, OWL, and JSON-LD. The files are processed by the server, after which they are imported into Neo4j. An ontology merging mechanism is implemented to handle duplicate vertices and create a consistent knowledge structure.

Main Functionalities

1. Uploading Ontologies

- The control system allows users to add files via the web interface.
- Files are checked for format and stored in a special repository.

2. Importing into Neo4j

- After uploading, the user has the opportunity to import the ontology into the graph database.
- The file format is automatically determined, and the data is added in the form of graph structures.

3. Merging ontologies based on semantic methods of mapping and automated merging of ontologies
4. Interactive web interface

- The user can view the list of uploaded ontologies.
- Buttons for importing and merging data are available.
- All actions are displayed in the event log, which simplifies process monitoring.

5. Working with API:

- The ability to connect to remote APIs to obtain ontology data has been implemented.
- The control system for manually entering queries for interaction with the graph database is provided.

The key mechanism for interacting with ontologies is built on HTTP request processing. The user sends a request to the server, which checks the correctness of the entered data, stores files in the system, and interacts with Neo4j via Cypher requests. The main import call uses built-in mechanisms for working with RDF graphs. To ensure correct merging of ontologies, an algorithm for comparing graph vertices has been developed. If vertices have the same unique properties, they are linked by the SAME_AS relation, which avoids duplication. Additionally, the ability to connect to external APIs is provided, which opens up prospects for integration with other platforms and services.

5. Discussions, conclusions and further researches

In previous works of the authors, the task of integrating data on sunflower research in Ukraine and international databases was set [1]. Integration of subject ontologies of agrotechnology is an important task, as it contributes to faster data exchange between countries, as well as acceleration of scientific research and implementation of innovative approaches in breeding, in the field of agronomy and biotechnology.

The paper presents the first results of the study on the integration of ontologies dedicated to sunflower cultivation in Ukraine and located on pan-European resources using the Neo4j graph database. This study is a component of the task of integrating the subject ontology of sunflower cultivation in Ukraine into the pan-European community.

The scientific and practical result of the presented study is the built The integrated ontology Helianthus EU Ontology. The integrated ontology Helianthus EU Ontology is created with three source Helianthus Ontology, Helianthus phenotype (seed & head) and Sunflower Ontology in the Crop Ontology. Helianthus Ontology& Helianthus phenotype (seed & head) are subject ontologies of agricultural technology for growing sunflower in Ukraine. They have created by authors and Experts from the Institute of Oilseed Crops of the National Academy of Agricultural Sciences were involved in the work. Sunflower Ontology in the Crop Ontology is pan-European resource. The previous analysis demonstrated the similarity/coincidence of class names, which correspond to the terminology adopted in the field. It allows to use the method of semantic comparison considering the practical component of the task of integrating ontologies, which are dedicated to the agricultural technologies of growing sunflower.

The work presents an ontology control system in Neo4j, which is the scientific novelty of the research conducted. The main goal of creating the practical value of such a system is to provide interactive management of ontology models without the need for deep knowledge of query languages for graph databases. The ontology control system in Neo4j allows you to load, store, import and merge ontologies in the graph database Neo4j. Thanks to containerization, the process of deploying the system on different servers is simplified, with support for multiple formats, which ensures universal use. Additionally, the possibility of connecting to external APIs is provided, which opens up prospects for integration with other platforms and services.

Further development of the ontology control system based on the Neo4j graph database is aimed at developing effective algorithms for matching, merging and integrating subject ontologies.

Acknowledgements

The work was performed as part of the research work "Making effective decisions based on intelligent technologies and open data (2024–2026)", registration number 05014, in the Department of System Analysis and Computational Mathematics of National University "Zaporizhzhia Polytechnic", Ukraine.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A.Bakurova, K. Vedmedeva, S.Vedmedev, E. Tereschenko, Ontological Model of Helianthus Cultivation in Ukrainian Conditions. In: CEUR Workshop Proceedings, 3396, 130–140, 2023.

[2] Planteome. URL: http://www.planteome.org.

[3] Crop ontology for agriculture data. URL: http://www.cropontology.org.

[4] E. Stanton, M.-A. Laporte, Planteome/CO_359-sunflower-traits: CO_359-sunflower-traits Ontology (v1.0) [Data set], Zenodo, 2024. doi:https://doi.org/10.5281/zenodo.11475283.

[5] Sunflower Genome Database. URL:https://www.sunflowergenome.org/

[6] M. Ashburner, C.A. Ball, J.A. Blake, et al. Gene ontology: tool for the unification of biology. Nat Genet 25(1):25-9 (2000). doi: 10.1038/75556.

[7] P.M. Mabee, M. Ashburner, Q. Cronk, et al. Phenotype ontologies: the bridge between genomics and evolution. Trends Ecol Evol 22(2007) 345–350. doi: 10.1016/j.tree.2007.03.013.

[8] R.C. Edmunds, B. Su, J.P. Balhoff, et al, Phenoscape: identifying candidate genes for evolutionary phenotypes, Mol Biol Evol 33(2016)13–24. https://doi.org/10.1093/molbev/msv223.

[9] Minimum Information About Plant Phenotyping Experiments, 2019. URL: https://www.miappe.org/

[10] I.Osman, S.B.Yahia, G.Diallo, Ontology integration: approaches and challenging issues. Inf. Fusion, 71 (2021), pp. 38-63, 10.1016/j.inffus.2021.01.007

[11] J. Euzenat, J. Portisch, M, Hladik, H. Paulheim, Background knowledge in ontology matching: A survey. Semantic Web. 2022;15(6):2639-2693. doi:10.3233/SW-223085

[12] D. Faria, B. Lima, MC Silva, FM Couto and C. Pesquita, AML and AMLC results for OAEI 2021, in: Proceedings of the 16th International Workshop on Ontology Matching Co-located with 20th International Semantic Web Conference (ISWC 2021) , Virtual Conference, October 25, 2021.

[13] P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh and C. Trojahn, eds, CEUR Workshop Proceedings, Vol. 3063, CEUR-WS.org, 2021, 131–136. http://ceur-ws.org/Vol-3063/oaei21_paper4.pdf.

[14] I. Mountasser, B. Ouhbi, F. Hdioud, et al. Semantic-based Big Data integration framework using scalable distributed ontology matching strategy. Distrib Parallel Databases 39, 891–937 (2021). https://doi.org/10.1007/s10619-021-07321-6

[15] J. Chakraborty, S.K. Bansal, L. Virgili, K. Konar and B. Yaman, OntoConnect: Unsupervised ontology alignment with recursive neural network, in: SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, Republic of Korea, March 22–26, 2021, C. Hung, J. Hong, A. Bechini and E. Song, eds, ACM, 2021, 1874–1882.
[16] M.S.M.Rudwan, J.V. Fonou-Dombeu, A Novel Algorithm for Multi-Criteria Ontology Merging through Iterative Update of RDF Graph. Big Data Cogn. Comput. 2024, 8, 19. https://doi.org/10.3390/bdcc8030019
[17] N.Kuusik, J. Vain, Ontology Merging Using the Weak Unification of Concepts. Big Data Cogn. Comput. 2024, 8, 98. https://doi.org/10.3390/bdcc8090098
[18] S.Babalou, B. König-Ries, Towards Building Knowledge by Merging Multiple Ontologies with CoMerger: A Partitioning-based Approach. 1 Jan. 2023 : 307 – 341.
[19] S. Ghidalia, et al. Combining machine learning and ontology: A systematic literature review. arXiv preprint arXiv:2401.07744 (2024).
[20] D. McGuinness, Conceptual Modeling for Distributed Ontology Environments. in Proceedings of the Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS 2000). Darmstadt, Germany. August 14-18, 2000.
[21] OntoMerge. URL: http://cs-www.cs.yale.edu/homes/dvm/daml/ontology-translation.html
[22] O. Curé,. Merging Expressive Ontologies Using Formal Concept Analysis. In: Meersman, R., Herrero, P., Dillon, T. (eds) On the Move to Meaningful Internet Systems: OTM 2009 Workshops. OTM 2009. Lecture Notes in Computer Science, vol 5872. Springer, Berlin, Heidelberg, 2009. https://doi.org/10.1007/978-3-642-05290-3_13
[23] A free, open-source ontology editor and framework for building intelligent systems, 2020. URL: https://protege.stanford.edu/
[24] D. Dou, D. McDermott, and Peishen Qi, 2003, Ontology Translation on the Semantic Web. In Proc. Int'l Conf. on Ontologies, Databases and Applications of SEmantics (ODBASE2003). LNCS 2888, 952-969.
[25] M. Uschold, M. Gruninger, Ontologies: principles, methods and applications, Knowledge Engineering Review, Vol. 11, Issue 2 (1996) 93-155. http://dx.doi.org/10.1017/S0269888900007797.
[26] U. Umar, T. A. Sardjono and H. Kusuma, Smart Ontology-Based System for Recommending Practices in Melon Cultivation, in IEEE Access, vol. 12, 162204-162216, 2024, doi: 10.1109/ACCESS.2024.3487288.
[27] Noor, Saba, et al. Agri Semantics: developments to improve data interoperability to support farm information management and decision support systems in agriculture. Smart farms: Improving data-driven decision making in agriculture. Burleigh Dodds Science Publishing Limited (2024). 75-96.
[28] D. Dosyn, V. Lytvyn, Models and methods for determining the usefulness of ontological knowledge: monograph, Novyy Svit 2000, L′viv ,2024.
[29] Institute of Oilseed Crops of the National Academy of Sciences of Ukraine, http://oilseed.org.ua/, last accessed 2024/11/21.
[30] J. De Bruijn, M. Ehrig, C. Feier, F. Martín-Recuerda, F. Scharffe, M. Weiten, Ontology mediation, merging and aligning, Semant. Web Technol. (2006) 95–113, doi:10.1002/047003033X.ch6
[31] S. Baker Effendi, B. van der Merwe, W-T.Balke, Suitability of Graph Database Technology for the Analysis of Spatio-Temporal Data. Future Internet 12(5):78, (2020) 1-31.
[32] Di Pierro, D., Ferilli, S.: An API for Ontology-driven LPG Graph DB Management. In: Proceedings of the 31st Symposium of Advanced Database Systems, Padua, Italy, 2023, pp. 303–316.
[33] Neo4j. URL: https://neo4j.com/