Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Ontology-Based Approach to Represent Knowledge of Graph Properties on Web

Halima Ben hebireche [1], Fatima Zohra Laallam[2], Khadra Bouanane[3] and Aicha Korichi[4]
[1]*Computer Science Department, University of Mohamed Khider, Biskra, ALGERIA*
benhbireche.halima@univ-ouargla.dz
2,4 Computer Science and Information Technology Department, University of Kasdi Merbah, Ouargla, ALGERIA.
3 Mathematics Department, University of Kasdi Merbah, Ouargla, ALGERIA.

ARTICLE INFO

Received: 28 Dec 2024
Revised: 18 Feb 2025
Accepted: 26 Feb 2025

ABSTRACT

Graph Theory is an active area of research that is growing very rapidly, owing to its successful applications in diverse fields. The main element in Graph Theory is the graph, each one possesses many properties and each one of these properties generates a class of graphs. On one hand, recognizing whether a given graph has or doesn't have certain properties could have significant importance because it may facilitate the resolution of a given problem in the represented system. On the other hand, a graph property recognition may not be easy; for the reason that an exponential algorithmic time is required to verify many of these properties.

In this paper, we propose an approach to representing knowledge from Graph Theory—specifically graph properties—within an ontology. The construction of this ontology consists of a single stage, namely the organization of knowledge using Protégé 5.2, in order to fully exploit the semantics of these concepts.

Keywords: Graph Theory, Graph Proprieties, knowledge, Protégé, OfGP, Ontology.

1. INTRODUCTION

Doubtless, Graph Theory is considered as one of the most growing areas of mathematics owing to its successful applications in diverse fields. In graph theory, a graph is known as a set of points (vertices) lied by lines (edges). It can be used as a mapping to represent concrete situations and problems visually. The graph permits to describe the operation of a system with a certain level of details to solve a real problem. Thus, concepts of graph theory are widely used in numerous fields such as Chemistry, Biology, Computer Science, Operations Research, Pure Mathematics, etc. [1–5].

Every graph could have many properties like connectedness, regularity, bipartition, triangle-free, even, order, planarity, hamiltonicity, possessing a k-factor, vertex-transitivity, etc. and each of these properties generates a class of graphs. Recognizing whether a graph has or does not have certain properties could have a significant importance. That is because it may facilitate the study of a given problem in the represented system. For example, the perfect matching property (1-factor) plays an important role in assignment problem or in detecting kekulé structures of a molecule graph [6], where vertex-transitive graphs are well known in Interconnection networks [7] and median graphs are related to location theory [8].

Actually, a recognition problem may not be easy since some of these problems such as hamiltonicity are known to be NP-complete. These problems may require an exponential algorithmic time to decide, for a given graph, if it has such property.

A natural way to avoid this issue is one has to use the anterior knowledge of properties s/he has on this graph and on the relations between them. For example, if is known that a given graph is 1-connected (vertex-connectivity=1), one can use this knowledge to deduce immediately that this graph is not hamiltonian. S/he can also deduce that a graph

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

has a 1-factor if s/he knows that it is regular and bipartite.

Therefore, constructing an ontology to represent graph properties and relations between these properties may be very useful for graph theory community, since it allows to deduce rapidly properties from other ones, and exploit fully the semantics and provide a common understanding of knowledge of this field.

In this paper, a domain ontology is presented and baptised Ontology for Graph Properties (OfGP). This ontology deals with graph theory field and, more specifically, deals with properties of graphs. The checked literature, shows that our is the first one in this area.

The remainder of this paper is structured as follows: Section 2 briefly deals with previous related work. Section 3 presents our contribution. In section 4, Ontology engineering and conceptual modelling paradigms, and then a detailed description of our ontology are provided. In Section 5, an experimental evaluation of the performance of our ontology is presented. Finally, the conclusions are also provided.

2. RELATED WORKS

Mathematical Knowledge Management (MKM)[1][9, 13, 14] is an active area of research, which is growing very rapidly. Michael Kohlhase, the author of [12] considers MKM as a new area of research that belongs to the intersection of many domains such as Mathematics, Artificial Intelligence, Computer Science, Library Science and Scientific Publishing.

In the literatures [9-12], principal challenges of Mathematical Knowledge Management (MKM) are discussed. Among these challenges one cites:

- Implementation of Digital Mathematics Libraries [15];
- Creation of systems for modern symbolic computation;
- Mathematical search and retrieval [16] ;
- Ontologies and languages for mathematical knowledge representation.

The most relevant previous works to this last challenge were summarized by Lange in [11]. This includes services, ontological models and languages for mathematical knowledge management on the Semantic Web and beyond.

However, we can notice that this knowledge base cannot represent the semantic aspect of these classes and relationships between them.

We can overcome this limit by using an ontology in addition to a knowledge base system, since the former can integrate semantic data and communication between heterogeneous systems. For more details about the difference between knowledge base and ontology, we refer the reader to this papers [24-26].

With regard to the field of Graph theory, we can notice that the existing ontologies in the literature review presents some of its concepts. However, It has no interest in graph properties since it presents only some generalities like Graph and Connected graph, which have been considered as subclasses of Set.

It is in this context that the current work is part of. The fact that in the literature, too little attention has been devoted to the field of Graph theory and more specifically, the automatic representation of graph properties or graph classes, encourages us to develop the domain Ontology for Graph Properties.

3. OUR CONTRIBUTION

The importance and the difficulty to recognize whether a graph has or not certain properties motivate us to propose another approach for addressing this challenge. The main contributions of the present work is to build a Graph properties ontology.

As we have previously mentioned in the introduction and to the best of our knowledge, there is no serious efforts to build an ontology for Graph Properties(a detailed description of our ontology that we called it OfGP is presented in

[1] http://www.mkm-ig.org/

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

section 4. The process of building this ontology is a long-term undertaking that requires concerted and multidisciplinary efforts.

The semantics of our concepts are formed by using both classical mathematical publications and electronic resources such as the survey of Andreas Brabdstadt et al [40] and ISGC.

ENGINEERING THE "OFGP" ONTOLOGY

Building an ontology is not something easy, it requires a thorough jobs analysis and understanding of the field and domain users. To build the good ontology, we must carefully choose the methodology we follow in the ontology development process. In the literature [41-44], we can find a lot of methodologies such as the On-To-Knowledge methodology [45], Waterfall methodology [46], Gruninger's methodology [43], the SENSUS methodology [47], Ontology Development 101 [48], and an METHONTOLOGY methodology [49], however, these methodologies are limited to case studies of the building of a specific ontology, or limited to a particular project. The authors of [47, 50] summarized and compared the main useful ontology design methodologies.

We choose the METHONTOLOGY method to develop OfGP because it depends on its application to apply on "logical descriptive" criteria for the concepts under study and due to its extensibility and high modularity [49].

This section describes briefly the steps for building our ontology.

4.1 Specification of the ontology OfGP

In this phase, we produce an informal ontology specification document, It is written in natural language. METHONTOLOGY proposes that at least some of the following information must be included in this document: the purpose of the ontology, level of formality of the implemented ontology and the scope, etc. Figure 1 illustrates a part from this specification.

Ontology requirement specification document
Domain: Graph Theory **Name**: OfGP(Ontology for Graph Properties) **Date**: May, 15th 2017 **Purpose**: Ontology for Graph theory substances to be used when information about graph properties is required in teaching, manufacturing, analysis, etc. This ontology could be used to ascertain, e.g., the complete graph is a connected graph, and we can use OfGP by ontology-based Graph Editor. **Level of Formality**: Semi-formal. **Scope**: The ontology has to cover a set of core concepts from the domain of Graph Theory (List of 390 elements: Multigraph, Simple, Bipartite, Hypercube, Tree, Leaf, Root, Vertices, Cycle...). **Implementation language**: The ontology is implemented in OWL2 using Protégé and the Pellet reasoner. **Sources of Knowledge**: Book, internet, and Graph Theory experts.

Figure 1. Ontology requirement specification in the domain of graph

4.2 Conceptualization of our 'OfGP' ontology

In this step, the Graph Theory(graph properties) knowledge is structured in a conceptual model that describes the problem and its solutions in term of a domain vocabulary which is identified in the ontology specification activity. The first step is to build a complete Glossary of Terms, it includes concepts, instances, verbs, and properties.

4.2.1 Construction of Glossary Terms and Extraction of relations for Graph Classes

This glossary contains definitions of the most used terms related to Graph Theory domain (concepts, attributes, relations, etc.) (see Table 1) that will be represented in the final ontology. Basic terms are extracted from these books:

- "Graph Theory (Graduate Texts in Mathematics)" by Bondy and Murty [53];

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

- "*Graph Theory*" (Graduate Texts in Mathematics) by Reinhard Diestel [54]. A free version of the book is available at http://diestel-graph-theory.com;
- "*Graph Theory: Penn State Math 485 Lecture Notes*" by Griffin [55];
- "*Introduction to graph theory*" by West et all [56];
- "*Introduction to graph theory*" by Chartrand [57];
- "*Graph theory and its applications*" by Gross and Yellen [58];
- "*Graph classes: a survey*" survey of Andreas Brabdstadt et al [40];
- "*The Interval Function of a Graph*" by Mulder [59].

Note that, many other references [59-76] have been used to enrich this ontology.

Concept	Definition
Graph	A graph G is an ordered pair (V, E) consisting of a set $V(G)$ of vertices and a set $E(G)$, disjoint from $V(G)$, of edges [53].
Multigraph	A graph $G = (V, E)$ is a multigraph if there are two edges e_1 and e_2 in E such that $e_1 = e_2 = \{v_1, v_2\}$ [55].
Simple	A graph that is not a multigraph and doesn't contain any loops.
Complete	A complete graph is a simple graph in which any two vertices are adjacent (joined by an edge) [53].
Connected	A graph is connected if, for every partition of its vertex set into two non empty sets X and Y, there is an edge with one end in X and one end in Y [53].
Regular	A graph G is k−regular if $d(v) = k \forall v \in V(G)$; a regular graph is one that is k−regular for some k [53].
Bipartite	A graph $G = (V, E)$ is bipartite if its vertex set V can be partitioned into 2 independent sets V_1 and V_2, such that each edge of E has one end in V_1 and one end in V_2.
(0,2)-graph	A (0,2)-graph is a connected graph such that any two vertices u and v, have either 0 or 2 common neighbours [65, 77].
Rectagraph	A triangle free (0,2)-graph.
Hypercube	A graph $G = (V, E)$ is a hypercube if the node set V consists of the 2^n n-dimensional boolean vectors, i.e., vectors with binary coordinates 0 or 1, where two nodes are adjacent whenever they differ in exactly one coordinate. [65, 78].
Median	A graph $G = (V, E)$ is median if for all vertices x, y, z: $I(x, y) \cap I(x, z) \cap I(y, z) = 1$ [72, 79, 80].
Antipodal	A graph $G = (V, E)$ such that, for each vertex u there exist a unique vertex $\bar{u}$ such that $I(u, v) = V$.
Diametral	A graph $G = (V, E)$ such that, for each vertex u there exist a unique vertex $\bar{u}$ such that $d(u, v) = Diam(G)$.
Connectd	A d-regular graph G such that $\kappa(G) = d$
Tree	A connected graph that doesn't contain any cycles.
Diamd	A regular graph such that the diameter is equal to the degree.

Spherical	A graph such that in each interval $I(u,v)$ and fi=or each $w \in I(u,v)$, there exist a unique vertex $\overline{w}$ in $I(u,v)$ such that $d(w,\overline{w}) = d(u,v)$.
Class6	A graph such that $\forall u,v \in V(G), \lvert N_1(u,v)\rvert \geq d(u,v)$.
Dist-Regular	Distance-Regular graph (see [?] for the definition).
...	...

Table 1. An excerpt of the Glossary of Terms of the OfGP ontology

Where

- $G = (V,E)$ is a graph;
- $n = |V|$ is its number of vertices or its order;
- $m = |E|$ is its number of edges or its size;
- $d(u)$ or degree of the vertex u denotes the number of edges that have u as extremity;
- the distance $d(u,v)$, denotes the length of the shortest path between two vertices u and v;
- the diameter $Diam(G) = \max d(u,v)$
- the interval $I(u,v)$ denotes the set of vertices that belong to a shortest (u,v)-path;
- $N_1(u,v)$ denotes the number of neighbours of the vertex u in the interval $I(u,v)$;
- the connectivity of a connected graph $\kappa(G)$ is the minimum number of vertices that deletion deconnect G.

4.2.2 Construction of taxonomy

We create a taxonomy of OfGP ontology, in order to structure the Knowledge of Graph Theory domain. This is a classification of information entities in the form of a hierarchy according to the presumed relationships of the Graph Theory concepts that are presented in glossary terms. Furthermore, the classification is based on the similarities of the information entities called concepts. This part of taxonomy represents the structure of the OfGP ontology in the next figure.

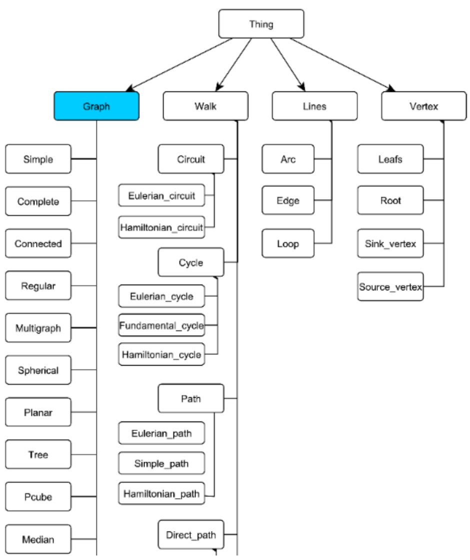

Figure 2. Part of knowledge Taxonomy of OfGP ontology

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4.2.3 Construction of the concept dictionary

A dictionary of concepts contains all domain concepts. For each concept, we will define the attributes, parents and relationships. The following table shows an excerpt from the dictionary of the concepts of OfGP ontology.

Concept	ConceptID	ParentID	attributes	Relation
Graph	Graph	Thing	name ordre size nb chromatique index chromatique radius ...	Has vertices Has edges Has path Has cycle ....
Simple	Simple	Proprieties	nb of loop coloration ...	
Hamiltonian	Hamiltonian	Proprieties		Has Hamilton cycle
...	...	....	...	...

Table 2. Extract from the original concepts table

4.2.4 Construction of the binary relations table

For each relationship used in the OfGP ontology, we define the relation name, name of potential relations (Relation)(Table 3), the concepts they link (Domain and Range), and a natural language definition of the notion behind the relation to try to capture their intension (Natural Language Definition).

Relation	**RelationID**	**Domain**	**Range**
Has Hamilton cycle	Has_Hamilton _cycle	Hamilton	Hamilton_cycle
Has arc	Has_arc	Directed_Graph	Arcs
...	...	...	...

Table 3. Extract from the original relations table

Currently, the OfGP ontology has 390 concepts and 32 relation-types. It grows as we reuse knowledge since we need to specify new relations.

4.2.5 Construction of the table of attributes

Attributes are properties that take their values in the predefined types (String, Integer, Boolean, Date ...). For each attribute we specify: name, type, cardinality, and default value (see Table 4).

Attribute	**Type of value**	**Card(min/max)**	**default value**
Name	String	1...n	-
Size	Integer	1...1	-
Ordre	Integer	1...1	-
Chromatic number	Integer	1...1	-
Orientation	Boolean	1...1	-
....	....	....	....

Table 4. Extract from the original attributes table

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

4.2.7 Construction of instances table

The instances or individuals are the "ground-level" components of an ontology, which represents a concrete concept occurrence of a class. For each instance in OfGP Must be specified each name, its attributes and values that are associated to it. Table 6 illustrates some instances created in this OfGP ontology. As an example in this ontology, we have an Tutte 12-cage instance, it is specifc type of graph classes. The Tutte 12-cage is Regular, Cubic, Cage, Hamiltonian, Connected and Bipartite graph with 126 vertices and 189 edges.

Name instances	Concept	Attributes	Values
12-cage_of_Tutte	Regular Cage Bipartite Hamiltonian Regular Connected	name size order orientation chromatic number chromatic index diameter radius automorphism girth k _regular k_connected	12-cage of Tutte 189 126 False 2 3 6 6 12096 12 3 3 .....
54-graph_of_Ellingham_Horton	Cubic Regular	name size order orientation chromatic number chromatic index diameter radius automorphism girth k _regular k_connected	54-graph of Ellingham Horton 81 54 False 2 3 10 9 32 6 3 3
Isolated-vertex	Vertex	name degree coloration	Isolated vertex 0 -
....	....	....	....

Table 5. Extract from the original instances table

In this ontology we have defined 100 instances.

4.3 Implementation of OfGP with the Web Semantic Technologies

We have implemented the Graph Properties and their relationships (OfGP ontology) with Protégé [52] in order to: visualize, validate and build our ontology in the OWL language in conformity with the W3C recommendations.

OfGP provides an integrated conceptual model for sharing information related to Graph Theory concepts. An OWL property is a binary relation to relate an OWL Class (Concept in OfGP) to another one or to RDF literals and XML Schema data types. For example, the Has_cycle Object property relates the Graph class to the Properties class. Described by these formal, explicit and rich semantics, the domain concept of Properties, its properties and relationships with other concepts can be queried, reasoned or mapped to support the Knowledge sharing across the Graph Editor. The following figures(Figure 3, Figure 4) represent the implementation of our ontology in Protégé.

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

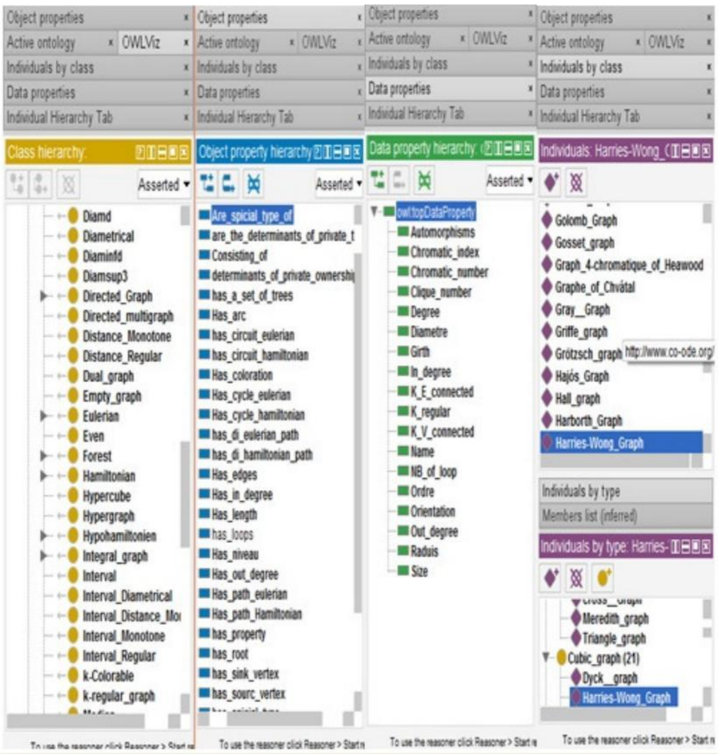

Figure 3. The implementation of our Graph Properties ontology in Protege 5

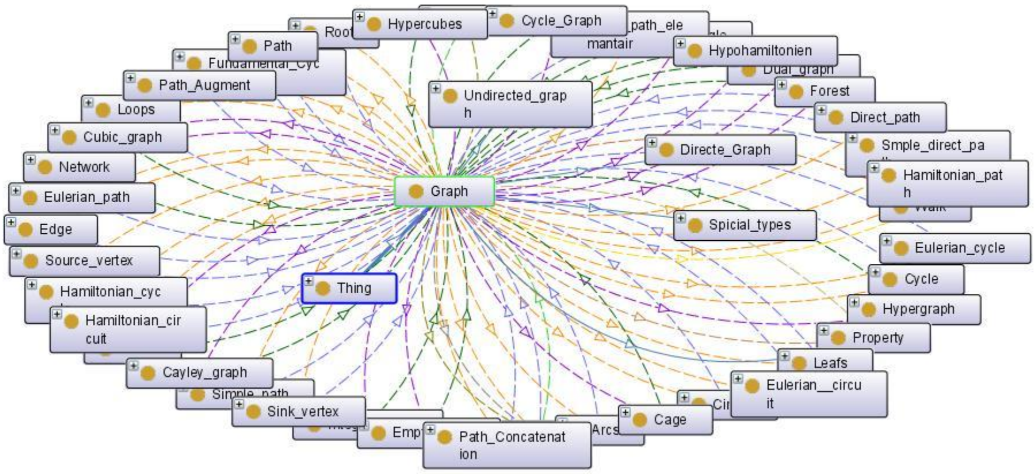

Figure 4. OfGP view in Protégé

Ontology in Protégé can be exported to different formats. We used protégé 5 to export the ontology to OWL that file called 'OfGP.owl'. This file contains the full description of our ontology.

4. EVALUATION

After finishing the implementation of our Graph Properties ontology, this section evaluates OfGP regarding all (no–functional requirements and functional requirements) and the ontology requirements specification document. Ontology evaluation means taking into consideration that guarantees the stability and accuracy of the ontology. Evaluation of the ontology avoids concept duplication, excessiveness and inconsistent relationships to create a better understanding. We execute the verification using protégé debugger. This debugger uses many reasoners to verify the ontology like HermiT, Pellet, Fact++, Racer, and Ontop. The ontology can be sent to the reasoner to automatically

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

compute the classification hierarchy, and also to check the logical consistency of the ontology. In Protégé 5, the manually constructed class hierarchy is called the asserted hierarchy. The class hierarchy that is automatically computed by the reasoner is called the inferred hierarchy. To automatically classify the ontology (and check for inconsistencies) the 'Classify' action should be used. The verification of our OfGP ontology presents positive result, we note that no suggestion is produced by reasoning and that "hierarchy Class" and "Class hierarchy inferred" are identical; that is to say, OfGP is consistent and coherent. Figure 5 illustrates the obtained results.

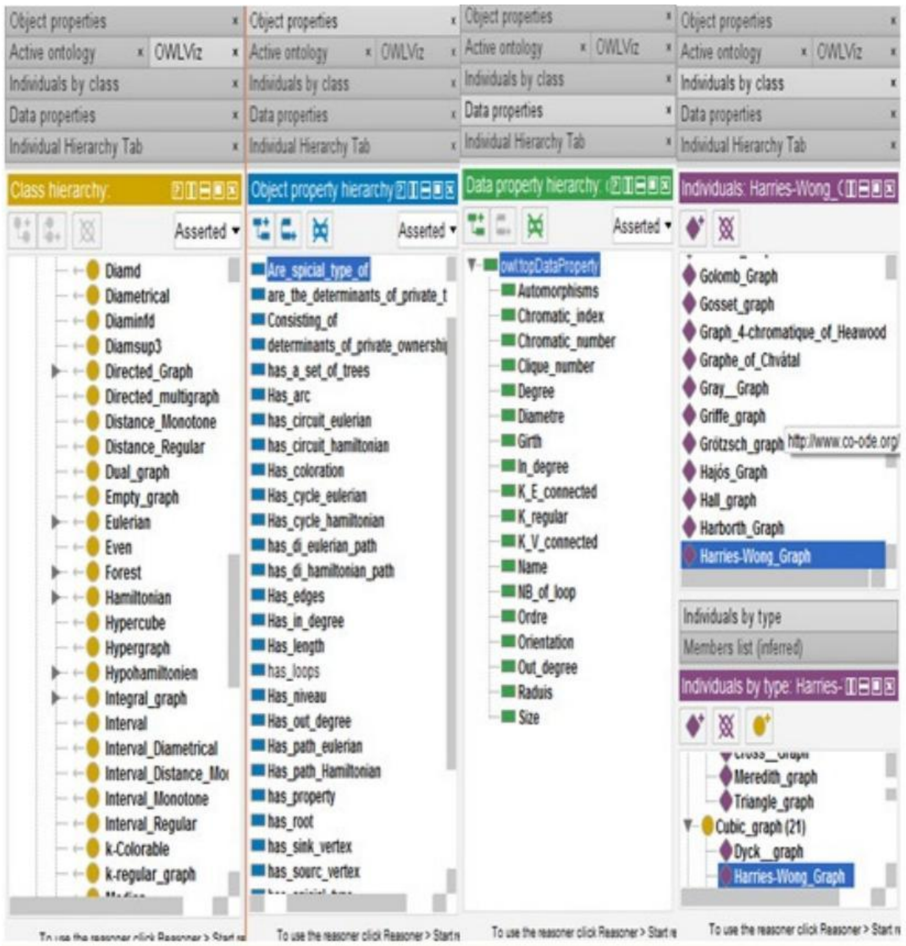

Figure 5. Checking the classification of the OfGP otology

5. CONCLUSION

In this article, we have presented our approach for classification of knowledge to build an ontology for Graph Properties (Classes) domain. The first step in the process is about the knowledge identification and specification of concepts, attributes and their relations in order to build the ontology called OfGP. This knowledge representation must be validated by graph theoretic experts before the creation of a typology and a taxonomy.

The OfGP ontology allows users to better understand the properties of a given graph, which favors interoperability of data, research, information retrieval, and automated inference of properties. It also favors the formalization and the sharing of knowledge by using semantic web technologies.

REFRENCES

[1] Che, Z., & Chen, Z. (2011). Forcing on perfect matchings: A survey. *MATCH Communications in Mathematical and in Computer Chemistry, 66*, 93–136.

[2] Brouwer, A. E. (2006). Classification of small (0, 2)-graphs. *Journal of Combinatorial Theory, Series A, 113*(8), 1636–1645.

2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

[3] Lavault, C. (1999). Interconnection networks: Graph-and group theoretic modelling. In *12th International Conference on Control Systems and Computer Science (CSCS12)* (Vol. 2, pp. 207). Romanian Society of Control Engineering and Technical Informatics.

[4] Pavlopoulos, G. A., Kontou, P. I., Pavlopoulou, A., Bouyioukos, C., Markou, E., & Bagos, P. G. (2018). Bipartite graphs in systems biology and medicine: A survey of methods and applications. *GigaScience, 7*(4), giy014.

[5] Pirzada, S. (2007). Applications of graph theory. In *PAMM: Proceedings in Applied Mathematics and Mechanics* (Vol. 7, No. 1, pp. 2070013). Wiley Online Library.

[6] Prasad, M. (2015). *Advanced graph theory and its applications*.

[7] Yegnanarayanan, V. (2005). Graph theory to pure mathematics: Some illustrative examples. *Resonance, 10*(1), 50–59.

[8] Klavžar, S., & Mulder, H. M. (1999). Median graphs: Characterizations, location theory and related structures. *Journal of Combinatorial Mathematics and Combinatorial Computing, 30*, 103–128.

[9] Shirinivas, S. G., Vetrivel, S., & Elango, N. M. (2010). Applications of graph theory in computer science: An overview. *International Journal of Engineering Science and Technology, 2*(9), 4610–4621.

[10] Sir, M., Bradac, Z., & Fiedler, P. (2015). Ontology versus database. *IFAC-PapersOnLine, 48*(4), 220–225.

[11] Giaretta, P., & Guarino, N. (1995). Ontologies and knowledge bases: Towards a terminological clarification. *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing, 25*(32), 307–317.

[12] Fonseca, F., & Martin, J. (2007). Learning the differences between ontologies and conceptual schemas through ontology-driven information systems. *Journal of the Association for Information Systems, 8*(2), Article 4.

[13] Ridder, H. N. (n.d.). Information System on Graph Classes and their Inclusions (ISGCI). Retrieved from http://www.graphclasses.org

[14] Carette, J., & Farmer, W. M. (2009). A review of mathematical knowledge management. In *International Conference on Intelligent Computer Mathematics* (pp. 233–246). Springer.

[15] David, C., Kohlhase, M., Lange, C., Rabe, F., Zhiltsov, N., & Zholudev, V. (2010). Publishing math lecture notes as linked data. In *Extended Semantic Web Conference* (pp. 370–375). Springer.

[16] Kohlhase, M. (2014). Mathematical knowledge management: Transcending the one-brain barrier with theory graphs. *European Mathematical Society (EMS) Newsletter, 92*, 22–27.

[17] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., & Bizer, C. (2015). DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web, 6*(2), 167–195.

[18] Sloane, N. J. A., et al. (2008). The On-Line Encyclopedia of Integer Sequences.

[19] Lange, C. (2011). *Enabling collaboration on semiformal mathematical knowledge by semantic web integration* (Vol. 11). IOS Press.

[20] Lange, C., & Kohlhase, M. (2009). A mathematical approach to ontology authoring and documentation. In *Intelligent Computer Mathematics* (pp. 389–404). Springer.

[21] Nevzorova, O., Zhiltsov, N., Zaikin, D., Zhibrik, O., Kirillovich, A., Nevzorov, V., & Biryaltsev, E. (2013). Bringing math to LOD: A semantic publishing platform prototype for scientific collections in mathematics. In *International Semantic Web Conference* (pp. 379–394). Springer.

[22] Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. *Proceedings of the AAAI Spring Symposium*, 33–40.

[23] Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management, 2*(1), 26–30.

[24] Sure, Y., Staab, S., & Studer, R. (2004). On-To-Knowledge Methodology (OTKM). In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (pp. 117–132). Springer.

[25] Jones, D., Bench-Capon, T., & Visser, P. (1998). Methodologies for ontology development. *Proceedings of IT&KNOWS Conference*.

[26] Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05*.

[27] Buitelaar, P., Cimiano, P., & Magnini, B. (2005). *Ontology learning from text: Methods, evaluation and applications* (Vol. 123). IOS Press.

Journal of Information Systems Engineering and Management
2025, 10(58s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[28] Gruninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. *Proceedings of IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*.
[29] Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The Knowledge Engineering Review, 11*(2), 93–136.
[30] Uschold, M. (1996). Building ontologies: Towards a unified methodology. In *Proceedings of the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems*. Citeseer.
[31] Fernández-López, M. (1999). Overview of methodologies for building ontologies. *CEUR Workshop Proceedings*.
[32] Cristani, M., & Cuel, R. (2005). A survey on ontology creation methodologies. *International Journal on Semantic Web and Information Systems, 1*(2), 49–69.
[33] Spyns, P., Meersman, R., & Jarrar, M. (2002). Data modelling versus ontology engineering. *ACM SIGMOD Record, 31*(4), 12–17.
[34] Gross, J. L., & Yellen, J. (2005). *Graph theory and its applications*. Chapman and Hall/CRC.
[35] Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies, 43*(5–6), 907–928.
[36] Aïder, M., & Aouchiche, M. (2002). Distance monotonicity and a new characterization of hypercubes. *Discrete Mathematics, 245*(1–3), 55–62.
[37] Aïder, M., & Aouchiche, M. (2005). Distance monotonicity and a new characterization of Hamming graphs. *Information Processing Letters, 96*(6), 207–213. https://doi.org/10.1016/j.ipl.2005.08.009
[38] Bondy, J. A., & Murty, U. S. R. (2008). *Graph theory (Graduate texts in mathematics)*. Springer.
[39] Stinson, D. R. (2007). *Combinatorial designs: Constructions and analysis*. Springer Science & Business Media.
[40] Paulin, F. (2014). *Groupes et géométries*. Lecture notes.
[41] Bouanane, K., & Berrachedi, A. (2017). 4-cycle properties for characterizing rectagraphs and hypercubes. *Czechoslovak Mathematical Journal, 67*(1), 29–36.
[42] Bouanane, K., & Berrachedi, A. (2016). On the Hamiltonicity of (0,2)-graphs. *Soumis*.
[43] Benhebireche, H., Laallam, F. Z., Bouanane, K., & Babatunde, A. A. (2016). Developing a domain ontology for graph properties. *Soumis*.
[44] Bouanane, K., & Berrachedi, A. (2014, March). Some properties of (0,2)-graphs. In *International Conference on Artificial Intelligence and Information Technology*.
[45] Lucas, É. (1882). *Récréations mathématiques* (Vol. 1). Gauthier-Villars.
[46] Avann, S. P. (1961). Metric ternary distributive semi-lattices. *Proceedings of the American Mathematical Society, 12*(3), 407–414.
[47] Bandelt, H.-J., & Mulder, H. M. (1983). Infinite median graphs, (0,2)-graphs, and hypercubes. *Journal of Graph Theory, 7*(4), 487–497.
[48] Harary, F., Hayes, J. P., & Wu, H.-J. (1988). A survey of the theory of hypercube graphs. *Computers & Mathematics with Applications, 15*(4), 277–289.
[49] Bandelt, H.-J., & Barthélemy, J.-P. (1984). Medians in median graphs. *Discrete Applied Mathematics, 8*(2), 131–142.
[50] Bandelt, H.-J., & Mulder, H. M. (1991). Pseudo-median graphs: Decomposition via amalgamation and Cartesian multiplication. *Discrete Mathematics, 94*(3), 161–180.
[51] Benmeziane, Z., & Bekkai, S. (n.d.). Gathering results on hamiltonicity and hamiltonicity of the join graph of a (0,2)-graph. *U.S.T.H.B, Faculté de Mathématiques*.
[52] Broersma, H. J. (2002). On some intriguing problems in Hamiltonian graph theory—A survey. *Discrete Mathematics, 251*(1), 47–69. https://doi.org/10.1016/S0012-365X(01)00325-9
[53] Broersma, H. J., Ryjáček, Z., & Vrána, P. (2012). How many conjectures can you stand? A survey. *Graphs and Combinatorics, 28*(1), 57–75.
[54] Berrachedi, A., & Mollard, M. (1999). Median graphs and hypercubes: Some new characterizations. *Discrete Mathematics, 208–209*, 71–75.
[55] Berrachedi, A. (1997). *Sur quelques propriétés métriques de graphes de type Hypercube* [Doctoral dissertation, U.S.T.H.B].
[56] Berrachedi, A., Havel, I., & Mulder, H. M. (2003). Spherical and clockwise spherical graphs. *Czechoslovak Mathematical Journal, 53*(2), 295–309.

[57] Berrachedi, A., & Mollard, M. (1998). On two problems about (0,2)-graphs and interval-regular graphs. *Ars Combinatoria, 49*, 303–309.

[58] Berrachedi, A. (1994). A new characterization of median graphs. *Discrete Mathematics, 128*(1), 385–387. https://doi.org/10.1016/0012-365X(94)90128-7

[59] Fink, J. (2007). Perfect matchings extend to Hamilton cycles in hypercubes. *Journal of Combinatorial Theory, Series B, 97*(6), 1074–1076. https://doi.org/10.1016/j.jctb.2007.02.007