TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 2337 - 2344

Knowledge Graph Modeling based on Generalized Property Graphs

Zhanfang Zhao[1], Hee-Kyung Moon[2]

[1]*Department of Computer Engineering, Hebei GEO University, China, zzfsjz@gmail.com*

[2]*Center for Educational Innovation, Wonkwang University, Korea, ybnjcw@wku.ac.kr*

Article Info
Volume 81
Page Number: 2337 - 2344
Publication Issue:
November-December 2019

Article History
Article Received: *5 March 2019*
Revised: *18 May 2019*
Accepted: *24 September 2019*
Publication: *12 December 2019*

Abstract

As a popular emerging technology, knowledge graphs (KGs) have become a platform of knowledge applications and services. Two dominant graph-based knowledge models, RDF and LPG, are widely used to construct large-scale KGs. It is argued that these models have some limitations to cope with complicated knowledge structures. This paper proposes a noble generalized property graph model that can seamlessly realize the various types of knowledge structures with compact expressiveness and robust formalism. The paper also describes several structures of knowledge graphs to demonstrate the expressive capability of the proposed property graph model. Since the generalized graph model is compatible with RDF and LPG, it can be practically applied to the usual KGs with effective performance.

Keywords*: knowledge graph, generalized property graph, attribute-value matrices, frame, knowledge modeling.*

1. INTRODUCTION

Web technologies have become one of the most prominent sources of information and knowledge. A lot of the efforts have been focused on using the Web platform to realize large-scale knowledge bases. Especially, since graph-based formalisms not only provide an intuitive and effective way for knowledge modeling but also bring an appropriate application environment with knowledge from the diverse domains, a wide variety of graph-based approaches has been proposed to represent knowledge structures more effectively and practically. So knowledge graphs (KGs) have become a popular emerging technology with the rising of artificial intelligence (AI) [1], [2]. Several large-scale, distributed KGs such as DBpedia, YAGO, Freebase, and Wikidata containing a large number of plentiful datasets have been developed [3], [4]. Moreover, with the dissemination of Linked Open Data (LOD), KGs from the diverse domains have been rapidly grown to play an important role for knowledge sharing and the development of knowledge services [3]-[5]. However, despite enormous efforts spent to build large-scale KGs, major issues to realize practical KGs consistently remains. The most crucial issue is the appropriate modeling of knowledge with the heterogeneous features. At the moment, two dominant graph-based knowledge models, Resource Description Framework (RDF) and Labeled Property Graph (LPG), are widely used to construct large-scale KGs [6], [7]. RDF as a standard data model using simple triple structures provides the foundation of LOD. LPG model using the key-value pair model shows stronger expressiveness and good performance in query and storage for large-scaled KGs. However, these two popular graph models reveal the critical problems in describing the complicated relationships inherent in knowledge. The graph-based knowledge modeling should be

November-December 2019
ISSN: 0193-4120 Page No. 2337 - 2344

able to represent the essential knowledge structures such as N-ary relationships and reifications effectively.

This paper proposes a novel graph-based modeling method to represent knowledge on the Web. The proposed model generalizes property graph using the unique property list structures. The paper also describes several knowledge modeling examples to demonstrate the expressive capability of the proposed generalized property graph model.

This paper is structured as follows. Section 2 reviews related work and conduct an empirical analysis of existing approaches for graph-based knowledge modeling. Section 3 provides a theoretical definition of a generalized property graph and describes its unique modeling structures. Section 4 presents the typical examples of KGs to show the modeling capability of GPG. Section 5 discusses summaries and challenging issues.

2. RELATED WORKS

In the field of AI, several graph-based knowledge representation formalisms such as Semantic Networks and Conceptual Graphs have been developed, and a wide range of applications of graph-based methods such as the Entity-Relation model, UML and Topic Maps in the domain of database and information systems development have been developed[1],[8],[9]. Especially, various knowledge representation approaches such as frame and feature structures play an important role in linguistic knowledge representation in computational linguistics [8], [10], [11].

In recent years a number of research into KGs have been conducted to realize Web-scale KGs containing the large volume of facts. DBpedia, YAGO, LOD Cloud, Freebase, Wikidata, and Cyc are the typical examples of the well-known KGs [3], [4], [12]. To implement KGs, two dominant graph data models, RDF and LPG, are widely applied for the materialization of Web-scale KGs [6], [7]. Although RDF and LPG both pursue graph-based knowledge modeling, they adopt different strategies to model knowledge representation.

RDF has been widely used as the core data model for Linked Open Data (LOD) that enables the Web of Data. Several comprehensive LOD products containing plentiful RDF datasets have emerged, such as DBpedia and YAGO [3], [4].

However, RDF modeling has been widely criticized for its awkward structures and semantic interpretations. Notably, the blank nodes and the reification provoke the serious difficulties in querying and searching [13], [14]. Although RDF reification has been withdrawn from the normative sections in the latest RDF Recommendation, the expressive capabilities of RDF remain an unresolved problem [15].

Recently, LPG model emerged from NoSQL database paradigm has received significant attention on account of the good performance in dealing with a considerable amount of diverse data generated on the Internet [6], [9]. LPG model owns distinctive features, using any number of key-value pairs to describe the semantic properties of knowledge. However, the conventional LPG model shows a lack of semantic expressiveness to realize semantic interoperability of the knowledge in the open and shared environment of the Web. The value types of the properties used in LPG model are also restricted so that it cannot represent knowledge properties having the compound or structured values.

Since RDF and LPG are popular graph modeling that has similar objectives, it is more reasonable to harmonize the distinguishable features of two graph formalisms in order to implement robust KGs effectively.

November-December 2019
ISSN: 0193-4120 Page No. 2337 - 2344

3. GENERALIZED PROPERTY GRAPH MODEL FOR KNOWLEDGE GRAPHS

This section describes the definition of a noble knowledge modeling method called generalized property graph (GPG). The characteristics of the property list that is the core structure of GPG are also described.

3.1 Definition of Generalized Property Graphs Final Stage

Although the generalized property graph (GPG) model adopts the approaches of typed feature-structured formalisms and LPG modeling, GPG embraces the open-world features of RDF and NoSQL conceptions of the key-value pair so that it can realize the common sharable KGs. GPG provides powerful expressiveness of knowledge structures and efficient store of property lists with index-free adjacency that can allow for fast querying.

[Definition: Generalized Property Graph] A GPG $\mathcal{G}$ is a directed, labeled, attributed, multi-relational graph consisting of $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{K}, \mathcal{P}, \mathcal{L}, \theta, \alpha, \beta)$, where $\mathcal{E}$ is a set of entities, $\mathcal{R}$ is a set of relations, $\mathcal{P}$ is a set of property lists, $\mathcal{L}$ is a set of labels, θ: $(\mathcal{E} \cup \mathcal{R}) \rightarrow \mathcal{L}$ is a label assignment, and α: $(\mathcal{E}$ x $\mathcal{P}) \rightarrow \mathcal{E}$ and β: $(\mathcal{R}$ x $\mathcal{P}) \rightarrow \mathcal{R}$ is a conceptual function for entities and relations, respectively.

In GPG, entities are represented as nodes and relations as edges, respectively. Both nodes and edges are labeled with their semantic functions. The edges are directed and can be multiple-edges between any two nodes.

[Definition: Entity] Entity $\mathcal{E}$ as a primitive conceptual element of a domain is an atomic building block of knowledge, consisted of a property list that specifies semantic constituents of the entity.

Entities denoted by nodes have a property list representing conceptual attributes inherent in the entities. Entities can have one or more labels. Entity labels can play a vital part in specifying ontological classes of entities. This makes it possible to form conceptual schema or hierarchy of ontological classes efficiently.

[Definition: Relationship] Relationship $\mathcal{R}$ as a semantic association between entities is a binary construct $\mathcal{R}(\mathcal{E}_i, \mathcal{E}_j)$, where $\mathcal{E}_i$ and $\mathcal{E}_j$ are different entities.

Relations are usually represented as directional links between two connected entities. So the elementary knowledge constructed by a single relation $\mathcal{R}(\mathcal{E}_i, \mathcal{E}_j)$ is usually represented by <$E_1 \rightarrow R \rightarrow E_j$>, called triple.

Relations denoted by edges have a direction to connect two entities. As the mandatory feature of GPG model, similar to LPG, every relation must have one and only one label to represent the edge uniquely. Much like entities, relations also can have their property list. The property list of relations usually describes the eventual or contextual attributes such as time, location and modality when the relationship is built between two entities.

[Definition: Property] A property $\mathcal{P}$ is a unary function $\mathcal{P}(\mathcal{X})$ to represent intensional features of the entity and relation, where $\mathcal{X}$ is either an entity or a relation. The property value can be literal, integer, date, and identifiable data.

Properties are special relationships where the ranges of a relationship are values of a data type (e.g., dates, age) while they are used to denote a binary relation in RDF(S) and OWL. Properties are intrinsic characteristics, for examples, age(John)=33, capital(German)=Berlin, and height(Everest) =8848. The adjunctual meanings also can be properties, for examples, mother(Tom)=Mary,

November-December 2019
ISSN: 0193-4120 Page No. 2337 - 2344

birthdate(Susan)=2003-05-25, and nickname(Hawaii)='The Aloha State'.

In additions, the contextual and functional features of relations are represented by means of the property structure. For example, the temporal and special property of relation M = marry(David, Jane) can be specified as date(M) = 1988-04-17 and place(M)=Paris *over* M.

Sometimes the distinction between relationships and properties are vague and misunderstood. In GPG, note that relationships are a binary construct to represent semantic association of two entities while properties are a unary functions to describe the conceptual features of an entity. So the properties have a value but relationships can't.

3.2 Property Values as Knowledge Structures Lists

The property list has a flexible structure that can be extended to capture the comprehensive range of complex knowledge structures. The extension of the property list structure maintains the original formalism of the property graphs without any principal variants. Considering the expressive power and Web environment of KGs, GPG accepts three types of extensions: container value structures, hierarchical property structures and disjunctive value structures.

Container Value Structures. Since the property graphs are based on the property list of property-value pairs, the diverse value structures are crucial to specify the properties effectively. In addition, the diverse value structures support the flexible definition of the properties and localize the related information in KGs. It is very common that some properties of real-world domain knowledge have multiple values, for examples, blood_type = ['A', 'B', 'O', 'AB'] and planet = < :Mercury, :Venus, :Earth, :Mars>.

In general, there are two types of container value structures: set that is an unordered collection of distinct values and list that is ordered collection of objects. GPG uses square brackets [ ] to denote the set value structures and angle brackets < > for the list structures, respectively. The co-author of Figure 2 and the flight route of Figure 3 show an example of container value structures.

Hierarchical Value Structures. The hierarchical property structures have been widely used in feature-based systems to provide more concise and understandable conceptualization for compound property[10], [16], [17]. The hierarchical property structures can localize the structured data of the properties and provide a preferable conceptualization of entities and relationships. As discussed in feature-based systems, the hierarchical property structures also have a complete theoretical basis and application use cases.

The effectiveness of hierarchical property structures can be observed in Figure 1, 2, and 4. The hierarchical property structures provide a compact and concise representation of structured knowledge

Disjunctive Value Structures. The disjunctive value structure is also a well-known essential structure to enumerate the possible feature values in feature-based systems[11], [18], [19]. The value disjunction provides an efficient way for grouping and localizing property values. The value disjunction has a very efficient structure to specify alternative property values, for examples, payment = cash ∨ visa ∨ mastercard *or* meal = hamburger ∨ sandwich. The disjunctive value structure enables compact structural representation and enhances semantic expressiveness.

As the value disjunction is the common value structure of the properties, this can be applied to modeling embarrassing knowledge structures in

TEST Engineering & Management
November-December 2019
ISSN: 0193-4120 Page No. 2337 - 2344

the various domains[11], [19]. Especially, the value disjunction coupled with hierarchical property structures shows efficient representative power to describe the spatial, temporal and thematic context. One of the prominent applications of the value disjunction with hierarchical property structures is time-dependent knowledge modeling. Although several approaches have been proposed to resolve this controversial constructs, RDF reification, named graphs and other extensions of RDF do not suggest plausible clarification[20]. However, the value disjunction with hierarchical property structures can efficiently describe time-dependent contextual data. This paper uses curly brackets { } to denote the value disjunction, i.e., payment={cash, visa, mastercard}.

4. KNOWLEDGE GRAPHS USING GENERALIZED PROPERTY GRAPHS

The conventional KGs adopt their knowledge modeling methods. However, much of modeling methods are criticized for the lack of expressiveness and clearness to represent the intricate knowledge structures such as N-ary relationships, temporal and spatial information, and reification. Although they extend their base framework to resolve these cumbersome problems, they still suffer from the complexity of knowledge representations. GPG modeling can transform the complicated relationships of these problems into the property list.

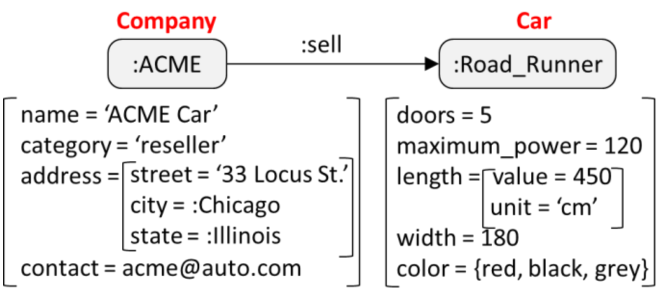

Figure 1: Simple knowledge graph of GPG

4.1 Simple GPG-based knowledge graphs

The simple knowledge graph using GPG is shown in Figure 1. The hierarchical value structures of the property address and length show the clear description of knowledge and its structure. The conventional modeling approaches and their extensions are difficult to achieve the expressive power of GPG. It is very clear that the property list of GPG using property hierarchy and container value, as shown in Figure 1, gives the compact and comprehensive representation of knowledge.

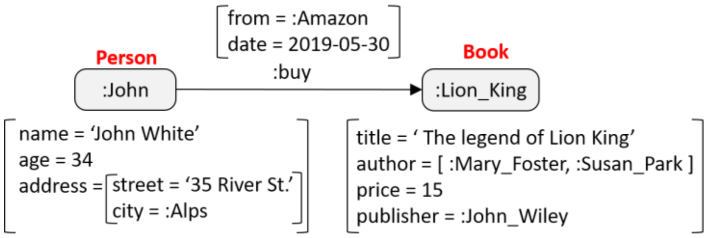

Figure 2: N-ary structure representation of GPG

4.2 Representations of N-ary structures in GPG

The essential relationship used in most of knowledge modeling is binary, namely a relationship between two entities. However, many knowledge structures of the real domain consist of more than 2 entities. So one of the critical issues in knowledge modeling is how to represent N-ary relationships within the binary framework of knowledge modeling. Although many approaches have been proposed to address the modeling N-ary relationships, the complexity of representation and interpretation is escalated by the rigid extension of the base framework[13], [21].

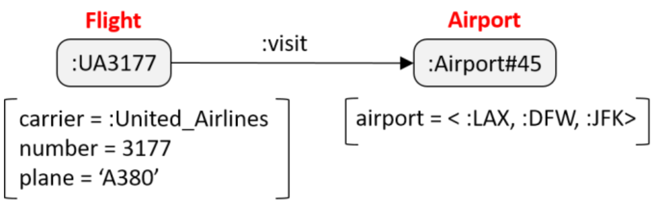

Figure 3: N-ary value representation of GP

In GPG, some subordinate attributes of knowledge that are usually dealt with part of N-ary relationship can be effectively represented

November-December 2019
ISSN: 0193-4120 Page No. 2337 - 2344

within the property list. Figure 2 shows knowledge representation of N-ary relationships in GPG. The properties from and date are generally regarded as relationships and cannot be reduced into attributes. However, GPG can represent the subordinate attributes of knowledge such as temporal and spatial information within the property list. This gives a more explicit representation of knowledge than other knowledge modeling approaches [22-25].

Since N-ary relationships are the usual structure in knowledge, a lot of research has investigated the aspects of N-ary. There are several well-known patterns in N-ary relationships[11].

Some patterns related to temporal and spatial knowledge are simply represented with the property list of the relationship. The following example shown in Figure 3 is one of the N-ary patterns. However, on account of the lack of expressive capabilities, the conventional models use the intricate structures. Actually, this pattern is related to property values rather than structural relationships. GPG simply resolve this N-ary pattern with the container values as shown in Airport of Figure 3.

4.3 Representations of reifications in GPG

The reification is also another cumbersome issues in KGs. Reification is a general purpose technique used in RDF and OWL for representing additional information about statements used for provenance, trust, time, space and certainty of meta triples. However, RDF reification has been widely criticized for its awkward structures and semantic interpretations. Additionally, the existing reification usually expresses structure by means of the blank nodes, leading to the difficulties in querying and searching.

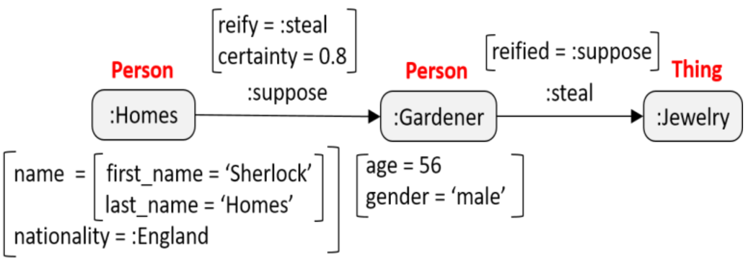

Figure 4: Reification representation of GPG

GPG can specify the reified statement by the co-referential property values as shown in Figure 4. As opposed to other approaches toward the reification, GPG can describe the reification more intuitively by the property list, since the entities and relations in GPG can have arbitrary property-value pairs to describe properties information. This method can also be applied to describe the contextual or situational knowledge.

5. CONCLUSION

In recent years a number of sizable KGs have been developed, the largest ones containing more than 100 billion facts. However, to disseminate KGs to the real-domain applications, many practical issues such as knowledge expressiveness, completeness, and performance should be solved.

This paper addresses a new knowledge modeling approach to achieve the goals of knowledge graphs for intelligent knowledge services. A GPG model from the perspective of conceptual modeling of knowledge is proposed. The proposed GPG has unique structures such as hierarchical property structures and disjunctive value structures to model real-domain knowledge. The paper also shows several knowledge graphs to demonstrate the expressive capability of GPG. As shown in the examples, GPG provides compact and comprehensive knowledge modeling and solves some intractable problems such as N-ary relations and reifications. Since GPG can be compatible with RDF and LPG, the current well-known KGs based on RDF and LPG can be efficiently transformed into or federated with GPG-based KGs. The sound implementation of GPG will be investigated as a further research project.

REFERENCES

1. X. Huang, J. Zhang, D. Li and P. Li, **Knowledge graph embedding based question answering**, in *Proc. 12th ACM International Conference on Web Search and Data Mining (WSDM2019)*, Melbourne Australia, 2019, pp. 105-113.
2. L. Ehrlinger and W. Wöß, **Towards a Definition of Knowledge Graphs**, in Proc. *12th International Conference on Semantic Systems (SEMANTiCS2016) and 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS16)*, Leipzig, Germany, 2016, pp. 1-4.
3. H. Paulheim. **Knowledge graph refinement: A survey of approaches and evaluation methods**, Semantic web, vol. 8, no. 3, pp. 489-508, Dec. 2016.
4. T. Heath and C. Bizer. **Linked data: Evolving the web into a global data space**, Synthesis lectures on the semantic web: theory and technology, vol. 1, no. 1, pp. 1-136, Feb. 2011.
5. M Pavlić, A Meštrović and A Jakupović, **Graph-based formalisms for knowledge representation**, in *Proc. 17th world multi-conference on systemics cybernetics and informatics (WMSCI 2013)*, vol. 2, Orlando, FL, 2013, pp. 200-204.
6. S. Das, J. Srinivasan, M, Perry, E. I. Chong, and J. Banerjee, **A Tale of Two Graphs: Property Graphs as RDF in Oracle**, in *Proc. 17th International Conference on Extending Database Technology(EDBT)*, Athens, Greece, 2014, pp. 762-773.
7. M. Margitus, G. Tauer and M. Sudit, **RDF versus attributed graphs: The war for the best graph representation**, in *Proc. 18th International Conference on Information Fusion (Fusion)*, Washington, DC, 2015, pp. 200-206.
8. J. F. Sowa. **Conceptual Graphs, Conceptual Structures: Information Processing in Mind and Machine**, *Addison-Wesley*, 1983.
9. M.A. Rodriguez and P. Neubauer. **Constructions from dots and lines**, *Bulletin of the American Society for Information Science and Technology*, vol. 36, no. 6, pp. 35-41, Sep. 2010,
10. M. Johnson, **Logic and Feature Structures**, in *Proc. 12th International Joint Conference on Artificial Intelligence (IJCAI1991)*, Sydney, Australia, 1991, pp. 992-996.
11. J. Véronis and N. Ide, **A feature-based model for lexical databases**, in *Proc. 14th conference on Computational linguistics*, vol. 2, Nantes, France, 1992, pp. 588-594.
12. J. Hoffart, F.M. Suchanek, K. Berberich and G. Weikum. **YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia**, *Artificial Intelligence*, vol. 213, pp. 28-61, Jan. 2013.
13. S. Batsakis and EGM. Petrakis, **Representing and reasoning over spatio-temporal information in owl 2.0**. in *Proc. 6th Workshop on Semantic Web Applications and Perspectives (SWAP 2010)*, Sydney, Australia, 2010, pp. 1-16.
14. W3C, **W3C Working Group Note, Defining N-ary Relations on the Semantic Web**, Available: https://www.w3.org/TR/swbp-n-aryRelations/
15. W3C, W3C working group note, **RDF 1.1 Concepts and Abstract Syntax.** Available: https://www.w3.org/TR/rdf11-concepts/.
16. B. Carpenter. **The logic of typed feature structures: with applications to unification grammars, logic programs and constraint resolution**, *Cambridge University Press*, 2005.
17. M. C. Emele and R. Zajac, **Typed unification grammars**, in *Proc. 13th conference on Computational linguistics (COLING90)*, vol. 3, Helsinki, Finland, 1990, pp. 293-298.

November-December 2019
ISSN: 0193-4120 Page No. 2337 - 2344

18. R.T. Kasper, **A Unification Method for Disjunctive Feature Descriptions**, in *Proc. 25th Annual Meeting of the Association for Computational Linguistics*, Stanford, CA, 1987, pp. 235-242.
19. A. Eisele and J. Dörre, **Unification of disjunctive feature descriptions**, in *Proc. 26th annual meeting on Association for Computational Linguistics. Association for Computational Linguistics*, Buffalo, NY,
20. A.G. Salguero, C. Delgado and F. Araque, **Easing the Definition of N–Ary Relations for Supporting Spatio-Temporal Models in OWL**, in *Proc. International Conference on Computer Aided Systems Theory (EUROCAST 2009)*, Springer, Berlin, Heidelberg, 2009, pp. 271-278.
21. AC. Kanmani, T. Chockalingam and N. Guruprasad, **RDF data model and its multi reification approaches: A comprehensive comparative analysis**, in Proc. *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 1, Coimbatore, Tamilnadu, India, 2016, pp. 1-5.
22. Hussain, H.I., Kamarudin, F., Thaker, H.M.T. & Salem, M.A. (2019) Artificial Neural Network to Model Managerial Timing Decision: Non-Linear Evidence of Deviation from Target Leverage, International Journal of Computational Intelligence Systems, 12 (2), 1282-1294.
23. Sasongko, G., Hunga, I.A.R., Julana, A.S.N. Wahyudi, Y., Paulus Leliak, P., Huruta, A.D. 2019. Public policy for housing development: a case study on housing development in Semarang Regency – Indonesia. Insights into Regional Development, 1(3), 280-300. https://doi.org/10.9770/ird.2019.1.4(1)
24. Reinhold, K.; Järvis, M.; Prause, G. 2019. Occupational health and safety aspects of green shipping in the Baltic Sea, Entrepreneurship and Sustainability Issues 7(1): 10-24. http://doi.org/10.9770/jesi.2019.7.1(1)
25. Shi, X., Dini, A., Shao, Z., Jabarullah, N.H. & Liu, Z. (2019) Impacts of photovoltaic/ wind turbine/ microgrid turbine and energy storage system for binding model in power system, Journal of Cleaner Production, 226, 845-857.