International Journal of Engineering & Technology, 7 (3.34) (2018) 562-566

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

LPG Representation of the Reification of RDF

Zhanfang Zhao[1], Sung-Kook Han[2], Ju-Ri Kim[3*]

[1] *College of Information Engineering, Hebei GEO University, Hebei, China*
[2]*Department of Computer Engineering, WonKwang University, City Iksan, JeonBuk, Korea*
[3] *College of Convergence & Liberal Arts, Wonkwang University, City Iksan, JeonBuk, Korea*
**Corresponding author E-mail: cyanic@wku.ac.kr*

Abstract

Background/Objectives: It is still a challenging issue to represent the reification effectively since the reification representation of RDF standard has been revealed some drawbacks.
Methods/Statistical analysis: Currently, there are two main graph data models: RDF and LPG. LPG is a popular graph data model that is usually applied to NoSQL graph databases.This paper derives three types of the reification structures in terms of the structural and semantic relationships of the reification statements. The detailed representation of each type of the reification is presented with the extended LPG model.
Findings: This paper proposes a novel approach to represent the reification structure of RDF from the perspective of LPG. The paper explores the formal, conceptual properties of the conventional LPG models and proposes their extension to capture more complex knowledge structures efficiently. These augmentations of LPG can achieve more efficient and flexible resource modeling. This paper derives three types of the reification structures in terms of the structural and semantic relationships of the reification statements: assertion, quantification, and entailment.The proposed approach not only preserves the structure and semantics of the reification but also enables LPG modeling of the complex structural statements to be easy and intuitive.This can contribute to transfer RDF graphs into LPGs.
Improvements/Applications: The implementation of the extended LPG and the query processing of the reification remain future work.

Keywords: *LPG, RDF, key-value pairs, reification, triples, LOD.*

1. Introduction

Resource Description Framework (RDF) is a standard data model for opening, sharing and interchanging data on the Web. The data on the Web, by the way, reveal very unusual behaviors. The data on the Web dynamically evolve over time, and may only hold valid at a specific time point or only within a specific area or use. Some data are ambiguous, untrustworthy and become frequently obsolete or outdated. Besides, the data may not represent the true facts, but personal recommendations, speculations, or assertions [1, 2]. The provenance of data plays an important role in the assessments of data quality, reliability, and trustworthiness. These are curial issues to realize the Web of data or Linked Open Data (LOD). To solve these issues, RDF standard allows a way to define a set of RDF triples that provide additional information about individual triples, such as the source, the temporal or spatial information, or the certainty, called the reification [3]. The reification is imperative for Web of Data to represent the semantics of the statements precisely and formally. However, RDF model based on the triple structure <subject, predicate, object> has limitations to fully capture semantic relationships of the complex statements. So, the reification representation of RDF is insufficient to meet the requirement of the precise representation of knowledge [4].
Labeled Property Graph (LPG) model based on the key-value pairs has stronger expressiveness than comparable RDF [5]. This paper analyzes the reification structures from the perspective of LPG and shows appropriate representations of RDF reifications. Although LPG not only has a powerful capability to deal with Big Data but also can manage more complex data, LPG shows lack of the common semantics to represent knowledge. The core property structure based on the key-value pairs needs to be extended to capture both semantic and structural complexities of the data. This paper proposes an extension of the conventional LPG with enhanced functionalities.
Most of the approaches addressing the problems of the reification have focused on the plausible structural representation [4, 6, 7]. There was little effort to analyze the conceptual structures of the reification statements. This paper derives three types of the reification structures in terms of the structural and semantic relationships of the reification statements. The detailed representation of each type of the reification is presented with the extended LPG model. The proposed approach not only preserves the structure and semantics of the reification but also enables LPG modeling of the complex structural statements easy and intuitive.
The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the basic concepts of LPG and describes its extension for more robust expressiveness. In section 4, the representation of the reification with extended LPG model is presented. Section 5 summarizes the contributions and future works.

2. Related Work

Representing and querying meta-knowledge for the statements including provenance, trust, certainty, time, and location have been emerging demands in creating and sharing Semantic Web

knowledge bases[1, 2]. The RDF standard introduces the notion of the reification as an approach to provide a set of RDF triples that describe some other RDF triple [3]. 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 node, leading to the difficulties in querying and searching. Although RDF reification has been withdrawn from the normative sections in the latest RDF Recommendation [3], the reification representing triples about triples is a basic requirement for the RDF data model but remains an unresolved problem.
A Named Graph is an RDF graph which is assigned a unique name to be referred in other triples. So named graphs use quads instead of triples [8, 9]. Named Graphs provide a high-value with a small and incremental change to RDF. However, the named graph is not intuitive since it is different from the original structure of RDF and requests its own storage mechanism.
The singleton property approach makes the predicates the object resources by assigning singleton property to the predicates [10, 11]. Therefore, the predicates with singleton properties can become subjects and objects of other meta-triples. This methodology can keep consistency with the syntax of RDF. Singleton property shows a good performance in the number of triples and query execution time since it preserves the triple structure of RDF. However, there remain some theoretic issues about making the predicates as the resources to be used as the subject or object. An additional issue of the singleton properties approach is that it introduces a large number of unique predicates, which is untypical for RDF data.
A syntactic extension of RDF called RDF* was proposed to deal with the reification by embedding triples in the subject or object position of another triple [12, 13]. RDF* simply emphasizes a syntactic extension of RDF that makes dealing with statement-level metadata more intuitive. At the same time, RDF* proposes a well-defined transformation of RDF* data back to standard RDF data and SPARQL extension to query RDF* [14, 15]. This approach seems to allow meta-resource since a triple can be a subject or object. The simple embedded structures increase the difficulties of a query as shown in SPARQL extension.
In addition to these, several other approaches like 4D-fluents and relational wrapper also have been proposed to tackle the reification, but these approaches adopted complex expression structures that bring the difficulties of the realization [16]. Moreover, only a few approaches using LPGs have been proposed [14].

3. Definition and Extension of LPG Model

Graphs are flexible and intuitive for modeling information resources, their relationships and the conceptual structure of their domain. In addition to the expressiveness, graphs can be stored efficiently and processed consistently with the well-known algorithms. LPGs are a popular graph data model that is usually applied to model data for NoSQL graph databases such as Neo4j[15]. This section explores the formal, conceptual properties of LPG model and proposes its extension to capture more complex knowledge structures efficiently.

3.1. Definition of LPG Model

LPG model is very popular in an emerged NoSQL database paradigm that is non-relational, non-ACID, schema-less database systems to handle a huge amount of diverse data generated on the Internet. LPG provides more compact, expressive representation of graph data modeling and efficiently store the key-value pairs with index-free adjacency that can allow for fast querying.
Definition (Labeled Property Graph): An LPG G is a directed, labeled, attributed, multi-relational graph consisting ofG = (V, E, P, L, δ, γ),where V is a set of vertices, E is a set of edges, P is a set of properties representing the relationships, L is a set of labels, δ is the labeling function mapping $(V \cup E) \rightarrow L$, and γ is the property assignment function mapping $(V \cup E) \rightarrow 2^{\mathcal{P}}$.
In LPG data model, entities or resources are represented as vertices and relationships as edges. The edges are directed, both vertices and edges are labeled with their roles and can have any number of properties and attributes, there can be multiple-edges between any two vertices [15].
An important aspect of LPG is that both vertices and edges can have labels and properties. This feature is essentially useful and flexible for providing the diverse informative metadata related to resources and relationships. So The LPG data model can be a ground model to generate various type of graph-based data model by simply adding or abandoning specific constraints on LPG.

- Features of Vertices: Vertices or nodes denote entities or resources of the domain. Vertices can contain any number of the properties consisting of the key-value pair. Although there are no restrictions to give the properties to vertices, the properties of a vertex are usually the intentional properties representing conceptual attributes inherent in the entity. The metadata or ontological vocabularies can be used for the key of the property. Vertices can have one or more labels. Vertex labels can play a vital part in specifying the roles of vertices. This makes it possible to form conceptual schema or hierarchy efficiently.
- Features of Edges: An edge, also known as a link, arc or relation, represents a relationship between two connected vertices to establish a conceptual context for each vertex. Edges have a direction to connect two vertices. Even though edges can be self-referencing or looping, they can never be dangling. As the mandatory feature of LPG model, every edge must have one and only one label to uniquely identify the edge. The edge label represents the relationships between two vertices while vertex labels represent the roles or categories of the vertex.

Much like vertices, edges also can have their properties. The properties usually describe the circumstantial or contextual attributes when the relationship is built between two vertices such as time, location and modality.

- Features of Properties: The property is the foundational mechanism of LPG model to describe the attributes of vertices (entities) and edges (relationships). Since the attributes are the intentional characteristics of an entity, object or relation, the property usually represents intrinsic or conceptual features such as color, weight, and size for vertices, time and location for edges. Since the property is essentially for expressing non-relational data, it should be distinguished from the associative features that are generally represented by the edges.
- Features of Labels: Labels are one of the foundational elements of LPG data model. Both vertices and edges have labels, however, their applications are different. The vertices labels are a way to assign the roles to vertices and to categorize vertices by means of their semantic features. The vertex labels are similar to rdf:type of RDF, but more efficient and powerful. The vertex labels can be used for many different purposes such as sub-graph creation, efficient LPG data store, and schema generation. On the other hand, every edge has mandatorily one and only one label that represents the relationship between two connected vertices. The edge label plays a role of the unique identifier of the edge and constructs the domain graph structure.

3.2. Extension of LPG Model

LPG model is especially used for managing data that are intrinsically graph-based such as social networks and Internet of Things (IoT). The key-value pair properties of LPG model not only has a powerful capability to deal with Big Data but also can model complex data structures efficiently. However, LPG model shows lack of semantic expressiveness and a foundational ontology to realize semantic interoperability of the data in the open and shared environment of the Web. The core property structure needs to be extended to capture both semantic and structural complexities of the data.

■ Vertex as a Resource**:** The vertices of LPG, playing a similar role of the resources of RDF, are placeholders for data properties consisting of the key-value pairs. This nature of the vertices is different from those of the resources of RDF uniquely identifiable by IRI. In LPG, labels used to indicate the roles and categorize the vertices are essentially important than the vertex identifiers. The vertex identifier is mainly used to define the edge relationships. Some NoSQL systems based on LPG internally assign the unique identifier to each vertex for efficient management of graph operations. Such a method may be enough for the localized data objects. However, data objects should be semantically complete and uniquely identifiable to be published and shared in the open environment of the Web. Thus, the vertices of LPG should have resource properties of RDF to harmonize two graph models. This also provides an identical view of data objects.

■ Ontology Vocabularies and Namespaces**:** The conventional LPGs use localized vocabularies within a specific system for labels and keys of the property. It is strongly required to use ontology vocabularies and namespace so that LPGs can be shared in the open environment like RDF graphs.

The ontology vocabularies and namespaces used in the labels can generate a conceptual schema of the domains. The conceptual schema makes LPG an abstract knowledge model and provides the substantial basis for high-level knowledge processing. And besides, the property also use ontology vocabularies with namespaces. The property efficiently consists of key-value pairs similar to tagging data values. The keys of the property can be regarded as metadata for vertices and edges and feasibly represented by ontology vocabularies with the namespaces. Ontology vocabularies with namespace for the key can provide the commonly shared vocabularies and the coherent semantic interoperability to LPG as RDF graphs.

■ Nested Property**:** The original definition of key-value does not give a strict constraint for the datatype of value. So the value of the key-value pair is usually an opaque string of bytes of arbitrary length. However, different systems expand the datatype for the convenient data modeling and graph traversal, for example, lists for heterogeneous, ordered collections of values and maps for heterogeneous, unordered collections of key-value pairs. LPG needs adequate value types than such the expansion for conceptual modeling.

The key-value datatype has been widely used in feature-based systems to provide more clear and understandable conceptualization for compound attributes [17]. The nested key-value datatype also has complete theoretic basis and application use cases. As an example of the key-value datatype shown in Figure 1, it provides a preferable conceptualization of resources and relationships.

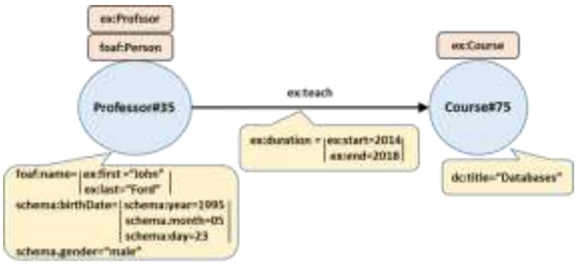

Figure 1: Example of the nested key-value data types

■ Public Identifiable Resources**:** In the practical domain modeling, the intangible, vague resources are usually presented. Since such the resources similar to blank nodes of RDF do not have the complete reality as a resource, it is difficult to define them as the usual resource. However, LPG model different from RDF that defines only blank nodes can assign the key-value pairs to the arbitrary vertices. Thus, LPG model can define this kind of resources as the generic resource that can be accepted in LPG modeling domain. The resource called public identifiable resource (PIR) has the predefined label such as za:Meta and an arbitrary vertex name. The label za:Mata declares the resource is intended to be used as an abstract resource that has only properties of the unreal entity.

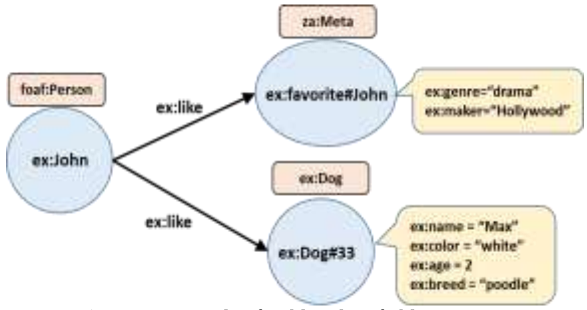

Figure 2: Example of Public Identifiable Resource

For example, in Figure 2, the vertex ex:favorite#John is PIR defined by the label za:Meta. It can be assumed that this vertex represents a resource related to the movie by means of its properties. PIR can create an identifiable resource by assigning its temporal vertex name. In other words, PIR provides the convenient way to make a collection of the properties into a definite resource.

■ Formalized Data Type Definition**:** Many graph database systems based on LPG use the diverse datatypes for the effective modeling and management. However, there are no common specifications to define the datatypes consistently. Moreover, the specifications of the aggregation datatypes such as containers and collections, especially, consisting of vertices or resources as the primitive elements have not been definitely addressed yet. This paper uses the key-value pair specification to define the complex datatypes involving vertices. This approach can keep methodical consistency in LPG modeling based on the key-value property.

The predefined key vocabularies are used to specify the datatypes of data aggregation related to vertices as follows:

● za:Construct declare the aggregated data structure.
● za:tag gives the data aggregation a literal name that can be used as the reference.
● za:datatype specifies the type of the aggregated structure such as rdf:Bag and rdf:List.
● za:order represents a sequential number of the resource in the aggregated structure.

Assuming that the interpretation of the za:datatype can be accomplished in the given system, this method can specify the diverse datatypes of arbitrary data structures consisting of vertices.

4. Representation of the Reification with Extended LPGs

The reification is a powerful mechanism to make statements about statements. RDF defines specialized vocabularies and unique structures to describe the reification owing to the expressive restrictions of its simple triple structures. The following is a typical example of RDF representation of the reification.

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix dc: <http://purl.org/dc/elements/1.1/#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix ex: <http://example.org/test#>.
ex:John rdf:type foaf:person;
foaf:name "John Ford";
foaf:age "33"^^xsd:integer;
ex:believe _:z1023.

z1023 rdf:type rdf:Statement ;
rdf:subject ex:Jane ;
rdf:predicate ex:wrote ;
rdf:object "Legend of Jedi" .
ex:Jane rdf: type foaf:person;
foaf:name "Jane Maple";
foaf:gender "female".

RDF model decomposes the statement into rdf:subject, rdf:predicate and rdf:object. Such the representation of RDF reification has been criticized for the awkward syntax and the obscure semantic interpretation. So, RDF reification has been withdrawn from the normative sections in RDF 1.1 Recommendation [10]. Nevertheless, the diverse approaches have been proposed to resolve this cumbersome problem such as named graphs, singleton properties and RDF* [14, 15]. While these approaches show concrete theoretical justification, they have tried to suggest a new formalism within RDF triple framework. The proposed approaches would reveal certain limitations to capture syntactic and semantic aspects of the reification.

According to the formal definition of RDF reification structures, there are 3 possible structures [14, 15]. However, this definition does not consider semantic aspects of the reification even though the reification is semantic relationships to make statements about statements. On the other hand, the key-value pairs of LPG can declaratively express the diverse semantic relationships with the resource [14, 15]. From the perspective of LPG, the reification structures can be categorized into 3 use cases.

■ Use case 1: Assertions: As the most typical use case of RDF reification, this case is widely used for the recommendation, specification of sources or provenance, presumption or assertion. As the assertion binds semantic relationships between two statements, it is the most important for the correlation of the predicates typifying the statements to be clearly represented in graph models. LPG can efficiently achieve the perplexed task by means of defining key-value pairs that show the correlation between two predicates as shown in Figure 3. The representation of the correlations can be archived by many different ways in LPG.

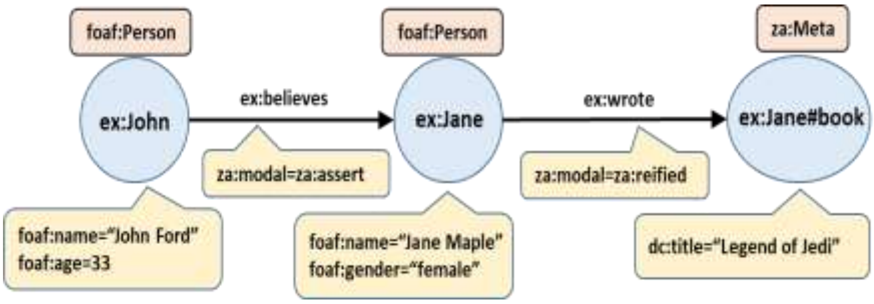

Figure 3: Representation of the Assertions of the Reification

The correlation representation of the reification is a new approach to describe structural and semantic relations of the reification. The property mechanism of LPGs is suitable for the correlated relations. Since the correlations of the reification statement are localized, query processing of LPG based on graph traversal can efficiently handle these correlated relations.

■ Use case 2: Quantifications: One of the main uses of the reification is to quantify the predicate. Temporal and spatial information is the typical example of this kind of reification. Since the simple triple structure of RDF has limitation to represent the quantification of the predicate, RDF has tried to resolve this problem under the reification. However, it is obvious that the quantification is different from the reification in syntactic and semantic structure.

// Turtle RDF
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix ex: <http://example.org/test#>.
ex:Alice rdf:type foaf:person;
ex:knows ex:Bob.
ex:Bob rdf:type foaf:person;
ex:workFor#1 ex:Univ.A.
ex:workFor#1 singletonPropertyOf ex:workFor;
ex:from "2002";
ex:to "2017".
ex:Bob ex:livedin#2 ex:Boston.
ex:livedin#2 singletonPropertyOf ex:livedin;
ex:certainty "0.8".

LPG can efficiently represent all kinds of the quantifications with the properties of the edges, including temporal and spatial information. So LPGs are not necessary to stick to the reification structure for the quantification of the predicates. The following shows LPG representation of the above reification statements of RDF.

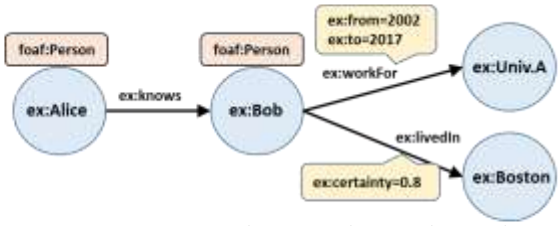

Figure 4: Representation of the Quantification of the Reification

The statement metadata such as certainty, provenance, and reliability, in general, can be regarded as the quantification in a broad sense. This approach is more appropriate than the reification of RDF.

■ Use Case 3: Entailments: Entailments are the conventional way to draw or deduce the certain facts from other statements. In the perspective of logical relationships, entailments are similar to the assertions since they are related to two statements. However, the necessary conclusions in the entailment are complete statements, not parts of the reification. The following is an example of the entailments mentioned as the reification in RDF.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix ex: <http://example.org/test#>.
ex:John rdf:type foaf:person;
ex:holdPosition#1 ex:President.
ex:holdPosition#1 singletonPropertyOf ex:holdPosition;
ex:hasSuccessor ex:Tom.
ex:John ex:holdPosition#2 ex:Chairman.
ex:holdPosition#2 singletonPropertyOf ex:holdPosition;
ex:hasSuccessor ex:Bill.

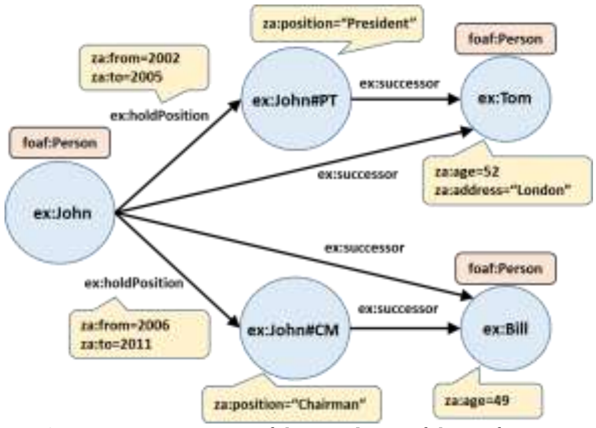

Figure 5: Representation of the Entailment of the Reification

In this example, the key-value pairs are used to form the correlation implicitly. Although these are similar to those of the assertions, the representation is different in that each statement is represented independently. The representation of Figure 5 is simple and provides more plentiful information than RDF representations.

5. Conclusion

With the increasing number of Web resources, the need to validate the available resources becomes essential. The reification to make assertions about statements represented by RDF triples is an important structure to provide meta-knowledge including provenance, validity, reliability, and trustworthiness of the information. However, RDF reification has been widely criticized for its cumbersome structures and revealed critical limitations in representing and querying the reified triples.

This paper presents a novel approach for the reification under the framework of LPG based on the key-value property. Since the conventional LPG models show lack of semantic correlations among the concepts. The functionalities to enhance LPG modeling such as the nested key-value type and public identifiable resource are proposed in this paper. These augmentations of LPG can achieve more efficient and flexible resource modeling. This paper derives three types of the reification structures in terms of the structural and semantic relationships of the reification statements: assertion, quantification, and entailment. The detailed representation of each type of the reification is presented with the extended LPG model. The proposed approach not only preserves the structure and semantics of the reification but also enables LPG modeling of the complex structural statements easy and intuitive. The LPG representation of the reification is succinct and easy to realize. The implementation of the extended LPG and the query processing of the reification remain future work.

Acknowledgment

This paper was supported by Wonkwang University in 2018

References

[1] Zhao, J., Bizer, C., Gil, Y., Missier, P., & Sahoo, S. (2010, June). Provenance requirements for the next version of rdf. In W3C Workshop RDF Next Steps.

[2] Hernández, D., Hogan, A., & Krötzsch, M. (2015). Reifying RDF: What works well with wikidata?. SSWS@ ISWC, 1457, 32-47.

[3] Cyganiak, R., et al, (2014). RDF 1.1 concepts and abstract syntax,Retrieved April 15, 2018, fromhttps://www.w3.org/TR/rdf11-concepts/

[4] Kanmani, A. C., Chockalingam, T., & Guruprasad, N. (2016). RDF data model and its multi reification approaches: A comprehensive comparative analysis. 2016 International Conference on Inventive Computation Technologies (ICICT), doi:10.1109/inventive.2016.7823190

[5] Tomaszuk, D. (2016, November). RDF data in property graph model. In Research Conference on Metadata and Semantics Research,Springer, Cham,104-115.

[6] Hoffart, J., Suchanek, F. M., Berberich, K., & Weikum, G. (2013). YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 3161-3165.

[7] Rula, A., Palmonari, M., Harth, A., Stadtmüller, S., & Maurino, A. (2012). On the Diversity and Availability of Temporal Information in Linked Open Data. The Semantic Web – ISWC 2012,492-507.

[8] Carroll, J. J., Bizer, C., Hayes, P., & Stickler, P. (2005). Named graphs, provenance and trust. Proceedings of the 14th International Conference on World Wide Web, 613-622.

[9] Watkins, E. R., & Nicole, D. A. (2006). Named Graphs as a Mechanism for Reasoning About Provenance. Frontiers of WWW Research and Development - APWeb 2006, 943-948

[10] Nguyen, V., Bodenreider, O., & Sheth, A. (2014). Don't like RDF reification?: making statements about statements using singleton property. Proceedings of the 23rd International Conference on World Wide Web - WWW 14,759-770.

[11] Nguyen, V., Leeka, J., Bodenreider, O., & Sheth, A. (2016). A formal graph model for RDF and its implementation. arXiv preprint arXiv:1606.00480.

[12] Hartig, O. (2014). Reconciliation of RDF* and property graphs. arXiv preprint arXiv:1409.3288.

[13] Hartig, O. (2017). Foundations of RDF* and SPARQL*:(An Alternative Approach to Statement-Level Metadata in RDF). In AMW 2017 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017. 1912

[14] Das, S., Srinivasan, J., Perry, M., Chong, E. I., & Banerjee, J. (2014, March). A Tale of Two Graphs: Property Graphs as RDF in Oracle. In EDBT, 762-773.

[15] Rodriguez, M. A., & Neubauer, P. (2010). Constructions from dots and lines. Bulletin of the Association for Information Science and Technology, 36(6), 35-41.

[16] Ramanujam, S., Gupta, A., Khan, L., Seida, S., & Thuraisingham, B. (2009). A Relational Wrapper for RDF Reification. Trust Management III IFIP Advances in Information and Communication Technology,196-214.

[17] Shieber, S. M. (2003). An introduction to unification-based approaches to grammar. Microtome Publishing.