Knowledge Big Graph Fusing Ontology with Property Graph: A Case Study of Financial Ownership Network†

Xiao-Bo Tang*, Wei-Gang Fu**, and Yan Liu***

*Wuhan University, Center for the Studies of Information Resources, No.16, Luojia Hill Road, Wuhan, Hubei Province, China, <xbtang2010@126.com>

**Wuhan University, Big Data Institute, No.16, Luojia Hill Road, Wuhan, Hubei Province, China, <fuweigang@whu.edu.cn>

***Wuhan University, Economic and Management School, No.16, Luojia Hill Road, Wuhan, Hubei Province, China, <yanliu.ems@whu.edu.cn>

Xiao-Bo Tang is a professor of information science at the Information Management School of Wuhan University in China. He holds a PhD in management science and engineering from Wuhan University and a master's degree in computer science and technology from Wuhan University of Hydraulic and Electrical Engineering. He is the director of the Information Systems Research Center in Wuhan University, member of the Association for Information Systems, the director of the China Branch of the Association for Information Systems, and the executive director of the Information Society of Hubei Province. His main research interests include knowledge organization and intelligence analysis.

Wei-Gang Fu (Corresponding author) is a PhD candidate at the Information Management School and a research assistant at the Big Data Institute of Wuhan University in China. His main research interests include knowledge organization, machine learning and network analysis of the financial domain.

Yan Liu is an associate professor in the Department of Finance at Economics and Management School of Wuhan University in China. He holds a PhD in economics from Stony Brook University in the USA. He is the deputy director of the Financial Big Data Research Center at Big Data Institute of Wuhan University. His main research fields include macroeconomics, data analytics, and machine learning.

Tang, Xiao-Bo, Wei-Gang Fu and Yan Liu. 2021. "Knowledge Big Graph Fusing Ontology with Property Graph: A Case Study of Financial Ownership Network." *Knowledge Organization* 48(1): 55-71. 36 references. DOI:10.5771/0943-7444-2021-1-55.

Abstract: The scale of knowledge is growing rapidly in the big data environment, and traditional knowledge organization and services have faced the dilemma of semantic inaccuracy and untimeliness. From a knowledge fusion perspective—combining the precise semantic superiority of traditional ontology with the large-scale graph processing power and the predicate attribute expression ability of property graph—this paper presents an ontology and property graph fusion framework (OPGFF). The fusion process is divided into content layer fusion and constraint layer fusion. The result of the fusion, that is, the knowledge representation model is called knowledge big graph. In addition, this paper applies the knowledge big graph model to the ownership network in the China's financial field and builds a financial ownership knowledge big graph. Furthermore, this paper designs and implements six consistency inference algorithms for finding contradictory data and filling in missing data in the financial ownership knowledge big graph, five of which are completely domain agnostic. The correctness and validity of the algorithms have been experimentally verified with actual data. The fusion OPGFF framework and the implementation method of the knowledge big graph could provide technical reference for big data knowledge organization and services.

Received: 1 June 2020; Revised: 13 September 2020; Accepted: 9 October 2020

Keywords: Ontology, property graph, knowledge fusion, consistent reasoning, financial ownership

† This study is part of the National Natural Science Foundation of China "Value Analysis, Discovery and Collaborative Creation Mechanism of Financial Big Data based on Knowledge Association" (No. 91646206),"Research on Intelligence Consulting Services Based on the Semantic Analysis of Text and Web" (No. 71673209) and the Fundamental Research Funds for the Central Universities of China "Wuhan University's independent scientific research project (humanities and social sciences)

1.0 Introduction

Knowledge organization is a process for conceptual representation of knowledge domains as an activity (Lígia et al. 2017; Bragato et al. 2019). According to Hjørland (2008), knowledge representation processes and knowledge representation systems can be used to characterize knowledge organization. Ontology, a typical method of traditional knowledge representation and knowledge organization systems, has garnered long-term attention (Stuart 2016). Its development context roughly spans from the semantic network to the ontology language represented by RDFS/OWL and then to the development of linked data (Bizer et al. 2009), which also accumulates many ontology knowledge bases. Knowledge graphs based on the property graph model have made great achievements in academia and industry in recent years, such as recommendation systems based on knowledge graph (Constantinov et al. 2016; Silva et al. 2010), the bioinformatics data management platform Bio4j (Pareja-Tobes et al. 2015), Facebook's OpenGraph (Ching et al. 2015), and Twitter's FlockDB (Hecht and Jablonski 2011). Knowledge graphs based on the property graph model are widely used to represent large amounts of heterogeneous knowledge from disparate sources.

The ontology model and property graph model are incompatible with each other, which directly hinders the continuity and inheritance of knowledge representation research. The ontology model usually defines the concepts of "things" and complex semantic relationships between concepts and can perform semantic reasoning. The model is generally expressed with the RDFS/OWL ontology language. The property graph model refers to a directed graph composed of nodes with multiple attributes and edges with multiple attributes (Rodriguez and Neubauer 2010; Hartig 2014; Tomaszuk and Dominik 2016). Different types of nodes and edges are identified by labels. On one hand, with the advent of the era of big data, traditional ontology is increasingly difficult to adapt to the rapid expansion of the knowledge scale. Due to the NP problem, semantic reasoning takes too long and crosses boundaries, and query responses are getting slower and slower, making it more difficult to land in actual application scenarios. This widens the gap between the research of ontology theory and knowledge services in industry (Cui et al. 2016; Gong et al. 2018). On the other hand, the knowledge contradiction rate is high in knowledge graphs based on the property graph model without semantic constraints and it is difficult to integrate domain knowledge from different experts using a property graph model (Miller 2013).

To effectively cope with the dual challenges of volume and semantic complexity brought by big data, this paper fuses ontology and property graph at the level of knowledge representation, and refers to the fused knowledge representation model as the "knowledge big graph." Finally, semantic consistent reasoning is performed to verify the effectiveness of the knowledge big graph.

2.0 Related research

The conceptual model of ontology is generally represented by the triple, which is significantly different from the storage model of ontology—plain text storage and variants of relational database storage. The difference between the conceptual model and the storage model not only causes low query efficiency but also brings obstruction to the release and utilization of the ontology. The graph database used to store the property graph adopts native graph processing and native graph storage technology. The resulting physical storage model formed is consistent with the conceptual model of a property graph, which can effectively match the characteristics of local data association in the big data environment, narrow the range of data traversal during data query and analysis, and improve the efficiency of formalization, storage, and utilization of large-scale knowledge. Therefore, many studies have begun to focus on how to transfer an ontology into a graph database, which are mainly divided into RDF level transformation and RDFS/OWL level transformation with complex semantic relationships.

2.1 RDF level transformation research

The ontology model is based on RDF triples and consists of more semantic vocabularies and primitives. Therefore, the ontology described by the mainstream language RDFS/OWL recommended by the W3C standard is also an RDF triple set, but the RDF triple set is not necessarily a standard ontology. The RDF triple form <subject, predict, object> naturally corresponds to the basic structure <node1, edge, node2> of a property graph. Therefore, many scholars started from the corresponding structure, designed the rules of ontology to property graph conversion, and implemented graph database storage of ontologies. For example, Gong et al. (2018) treated the attribute values in the RDF directed graph as nodes in the property graph model and dumped the oil domain ontology into the Neo4j graph database. Drakopoulos et al. (2017) converted the RDF triples corresponding to the instance layer of the ontology into edges in the property graph, where object property name and datatype property name were used as edge labels. Tomaszuk and Dominik (2016) proposed an algorithm, YARS, that translates RDF graphs into attribute graphs and serializes them into graph databases. The RDF-level transformation research only retains the data content of the ontology, which essentially degenerates the ontology into a set of RDF triples and then transforms it into a property graph model, losing the semantic constraints of the ontology itself.

2.2 RDFS/OWL level conversion research

Most of ontologies are described by the primitives provided by the RDFS/OWL ontology language, so some scholars mapped primitives in RDFS/OWL to the labels and attributes in the property graph model to represent the complex semantic relationship of the ontology. Krötzsch et al. (2016, 2017) used attribute logic to represent the ontology semantics in the property graph and explored the logical fit of the ontology into the property graph. Konno et al. (2017) constructed a two-layer property graph transformed from a retail ontology. Pham et al. (2018) built a computer science domain knowledge base and compared the query efficiency of the ontology version to the property graph version.

Related research mainly focused on the problem of dumping the ontology into a graph database at the physical level, rather than the fusion of the ontology and property graph at a higher level; that is, the semantic reasoning of the ontology and the predicate attribute of the property graph are not merged into a logical self-consistent whole. Therefore, from the perspective of knowledge fusion, this paper explores the fusion of the ontology and the property graph at the level of knowledge representation and constructs a knowledge big graph with both ontological reasoning and predicate attribute representation capabilities. Application scenarios of Chinese financial ownership network is used to verify the feasibility and practical value of the knowledge big graph proposed in this paper.

3.0 Knowledge fusion—the theoretical basis of knowledge big graph

The core function of knowledge fusion can be summarized as extracting knowledge elements from heterogeneous knowledge resources, adopting methods such as transformation, reasoning, merging, reorganization, and integration to establish a unified knowledge model. New knowledge can be obtained from the generated knowledge model to provide high-quality knowledge services (Smirnov et al. 2015; Preece et al. 2000). The ontology model can be regarded as knowledge A, and the property graph model can be regarded as knowledge B. Exploring the fusion of the ontology and the property graph is essential to establish the fusion logic of knowledge A and knowledge B. Therefore, it is possible to start from the conversion, merger, and reduction of knowledge elements and consider the mapping of the relationships among elements, that is, to investigate the integration of the ontology and the property graph from the perspective of knowledge fusion to ensure the logical consistency and coordination of the two knowledge representations in the knowledge big graph and to obtain new knowledge from the fusion results.

Ontology generally consists of a concept layer and an instance layer. Its description logic is <TBox, ABox>, where TBox corresponds to the concept layer of the ontology and ABox corresponds to the instance layer of the ontology (Dutta et al. 2014). The specific composition is as follows:

> TBox: a finite set of axioms such as term inclusion relationship $C \sqsubseteq D$, term equivalent relationship $C \equiv D$, term mutual exclusion relationship $C \sqcap D = \emptyset$;
> ABox: a finite set of conceptual assertions $C(a)$, role assertions $R(a, b)$, negative role assertions $\neg R(a, b)$, identity assertion statements $a \approx b$, and negative assertion statements $a \not\approx b$.

Property graph uses graph theory in mathematics as their logical basis, with a relatively simple structure, mainly consisting of nodes and edges. The property graph can be represented by a quaternion <V, E, P, L>, and the corresponding mapping functions $\langle h_e, t_e, p_v, p_e, l_e, l_v \rangle$ are attached. The description of each structure is as follows:

1. V represents the set of nodes in the property graph;
2. E represents the set of edges in the property graph;
3. P represents the set of attributes in the property graph, generally represented by the key-value pair <attribute name, attribute value>;
4. L represents the set of labels in the property graph;
5. h_e represents the bijective function of E to the head node V_h;
6. t_e represents the bijective function of E to the tail node V_t;
7. p_v represents the injective function of P to V;
8. p_e represents the injective function of P to E;
9. l_v represents the injective function of L to V;
10. l_e represents the bijective function of L to E.

In the above formal description, the bijective function represents a one-to-one mapping relationship, and the injective function generally represents a many-to-one mapping relationship. In a property graph, a node can belong to multiple types at the same time, that is, it can possess multiple labels, so l_v is an injective function. An edge only expresses a relationship; that is, there is only one label, so l_e is an injective function.

The ontology model and the property graph model are heterogeneous in syntax and semantics. Through a knowledge fusion lens, the fusion of ontology and property graph is mainly divided into transformation fusion, recombination fusion, and mapping fusion. Transformation fusion refers to the syntax conversion that converts the grammatical format of heterogeneous knowledge resources to the same syntax type. For example, the primitives described by RDFS/OWL in an ontology are converted into attribute key-value pairs and labels in a property graph. Knowledge resources are also semantically heterogeneous, which brings the need to merge and fuse the same element parts in an ontology and a property graph at the granularity of the knowledge elements and recombine different parts. Mapping fusion refers to mapping the <TBox, ABox> structure of an ontology and the four-tuple representation of a property graph to generate the skeleton of the knowledge big graph.

Obtaining new knowledge is one of the goals of knowledge fusion, and it should also be the goal of fusing ontology and property graph. The way to get new knowledge for an ontology is ontology reasoning, and for a property graph, it is graph mining based on path traversal and query. We fuse the two methods of obtaining new knowledge to generate a new reasoning mode. The new mode of inference is not only the new knowledge obtained after the knowledge fusion but also the means to obtain the new knowledge from the knowledge big graph after fusing ontology and property graph.

4.0 Ontology and property graph fusion framework—the logical realization of knowledge big graph

The explicit semantics of structured data and knowledge bases are generally described by an explicit or implicit vocabulary. The vocabulary is the data that describes the data, that is, metadata. Metadata describing the structure of the ontology is called primitive in ontology. This paper maps main primitives in the traditional ontology language RDFS/OWL recommended by the W3C into property graph model and develops the ontology and property graph fusion framework (OPGFF) as shown in Figure 1.

The content layer fusion in OPGFF mainly involves combining the primitives of the TBox layer and the ABox layer in the ontology with the elements in the property graph model. The constraint layer fusion combines the consistent reasoning of the ontology with the path traversal and predicate attribute of the property graph on the basis of the content layer fusion to realize the correctness of the knowledge and the derivation of new knowledge. A knowledge big graph is generated through the two-layer fusions, and the subsequent semantic query and semantic inference rules based on the knowledge big graph can be highly decoupled and completely domain agnostic.

Explanation	RDFS/OWL primitive	mapped property graph element
a concept or class	*owl:Class*	*Node(:Class)*
role that associates one instance with another	*owl:ObjectProperty*	*Node(:ObjectProperty)*
transitive role	*owl:TransitiveProperty*	*Node(:TransitiveProperty)*
antisymmetric role	*owl: AsymmetricProperty*	*Node(:AsymmetricProperty)*
role that associates instance with literal	*owl:DatatypeProperty*	*Node(:DatatypeProperty)*
role-to-concept domain relationship	*rdfs:domain*	*Edge(:Domain)*
role-to-concept range relationship	*rdfs:range*	*Edge(:Range)*
role-to-role inverse relationship	*owl:inverseOf*	*Edge(:InverseOf)*
role-to-role equivalent relationship	*owl:equivalentProperty*	*Edge(:EquivalentProperty)*
role-to-role disjoint relationship	*owl:disjointProperty*	*Edge(:DisjointProperty)*
role-to-role sub-property relationship	*rdfs:subPropertyOf*	*Edge(:SubPropertyOf)*
concept-to-concept subclassOf relationship	*rdfs:subClassOf*	*Edge(:SubClassOf)*
concept-to-concept equivalent relationship	*owl:equivalentClass*	*Edge(:EquivalentClass)*
concept-to-concept disjoint relationship	*owl:disjointWith*	*Edge(:DisjointWith)*

Table 1. Terminology mapping in the TBox layer fusion

Box 1a

$$\text{ClassNodes(PG)} = \{\text{tc : "Class"} < \text{"cname"=tco.ConceptName} > \mid \text{tco} \in \text{Classes(Ontology)}\}. \quad (1)$$

$$\text{RoleNodes(PG)} = \left\{\text{tp : RoleType} < \text{"ename"=tpo.RoleName} > \middle| \begin{array}{l}\text{tpo} \in \text{Roles(Ontology)},\\ \text{RoleType} \neq \text{DatatypeProperty}\end{array}\right\}$$
$$\cup \quad \{\text{tp : DatatypeProperty} < \text{"pname"=tpo.RoleName} > \mid \text{tpo} \in \text{Roles(Ontology)}\}. \quad (2)$$

Box 1b

$$\text{TEdges(PG)} = \{\text{tr : RelationName} \mid \text{RelationName} \in \text{OntologyRel}\}, \text{where OntologyRel}$$
$$= \{\text{SubClassOf, SubPropertyOf, DisjointWith, Domain, Range}\}. \quad (3)$$

Box 2

$$h_e(\text{tr}) = \text{tp}; (\text{tr}) = \text{tp}; h_e(\text{tr}) = \text{tc}; t_e(\text{tr}) = \text{tc}, \text{where tr} \in \text{TEdges(PG)}, \text{tp} \in \text{RoleNodes(PG)}, \text{tc} \in \text{ClassNodes(PG)}. \quad (4)$$

4.1 Content layer fusion

Content layer fusion refers to redefining the TBox layer and the ABox layer of an ontology in a property graph model. The result of the fusion corresponds to the schema layer and instance layer of the knowledge big graph so that the knowledge big graph can express the basic semantics of the ontology.

4.1.1 TBox layer fusion

We represent the semantics of the ontology's TBox layer by converting the concepts, roles, and relationships between them into nodes and edges in the property graph as shown in Table 1. First, the concepts and roles in the ontology are converted into nodes of the property graph so that the basic skeletal elements corresponding to the ontology TBox layer are constructed in the knowledge big graph. The transformation fusion formula is shown in Box 1a.

ClassNodes(PG) is the set of nodes representing all concepts of the ontology in the knowledge big graph, and RoleNodes(PG) is the set of nodes representing all roles of the ontology in the knowledge big graph. (:"Class"、: RoleType、: "DatatypeProperty") are used as the labels of nodes in the knowledge big graph, and ("cname"、"ename"、"pname") are used as the properties of nodes in the knowledge big graph. The meaning of tco ∈ Classes(Ontology) is that the symbol "tco" represents the conceptual elements of the ontology.

The edges in the property graph model are used to represent the vertical and horizontal relationships between the elements forming the skeleton structure of the knowledge big graph. The transformation fusion formula is shown in Box 1b.

TEdges(PG) is a set of edges representing all the relationships between concepts and concepts, roles and roles, and concepts and roles. Finally, the bijective functions h_e and t_e are used to associate the above nodes and edges together to form specific semantics. The recombination fusion formula is shown in Box 2.

Through the various combinations of these four fusion equations (1)~(4), the complex semantic relationships formed between the concepts and roles in the ontology TBox layer can be represented in the knowledge big graph. For example, h_e(tr(:Domain))=tp, t_e(tr(:Domain))=tc_1, h_e(tr(:Range))=tp, t_e(tr(:Range))=tc_2 jointly express the semantics of role tp, that is, role tp associates an instance of concept tc_1 with an instance of concept tc_2. For example,

$$\text{Person(: Class)} \xleftarrow{\text{:Domain}} \text{AuthorOf(: ObjectProperty)} \xrightarrow{\text{:Range}} \text{Book(: Class)}$$

expresses the authorOf role relationship between the instance belonging to Person and the instance belonging to Book.

4.1.2 A Box layer fusion

We map the concept assertions and role assertions in the ontology to the nodes, edges, and attributes of the nodes in the knowledge big graph and connect the corresponding edges and nodes to represent the semantic relationships among instance objects and semantic constraints of the TBox layer on the ABox layer of the ontology. The mapping fusion formula is shown in Box 3.

Box 3

$$\text{InNodes(PG)} = \left\{\text{ni : tc."cname"} < \text{tp."pname"} = \text{specialValue}, \ldots > \middle| \begin{array}{l}\text{tc} \in \text{ClassNodes(PG)},\\ \text{tp} \in \text{RoleNodes(PG)}\end{array}\right\} \quad (5)$$

$$\text{AEdges(PG)} = \{\text{ar : tp."ename"} \mid \text{tp} \in \text{RoleNodes(PG)}\}. \quad (6)$$

$$h_e(\text{ar}) = \text{ni}; t_e(\text{ar}) = \text{ni}, \text{where ni} \in \text{InNodes(PG)}, \text{ar} \in \text{AEdges(PG)}. \quad (7)$$

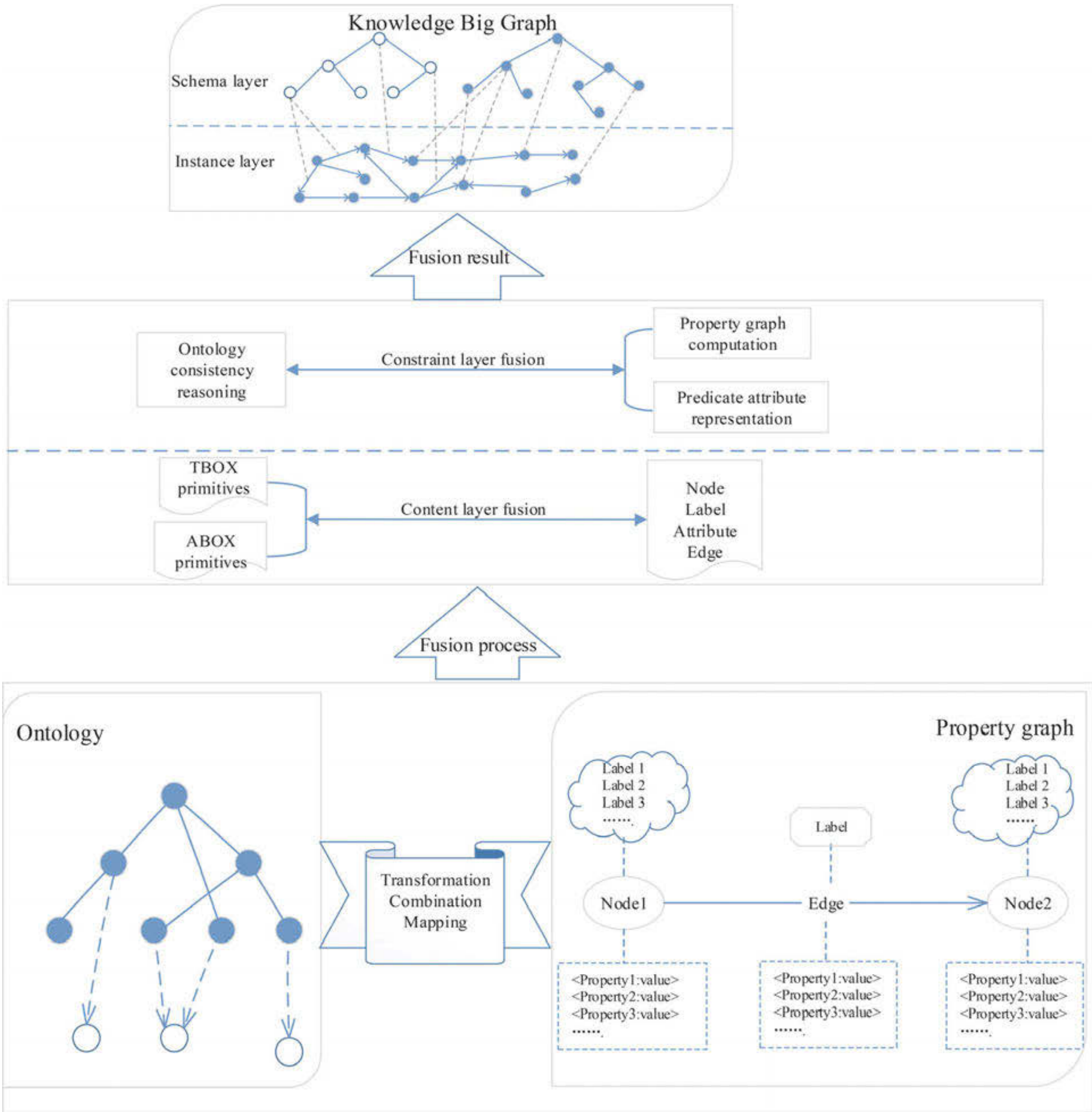

Figure 1. Ontology and property graph fusion framework(OPGFF).

InNodes(PG) is the set of nodes representing all the concept assertions of the ontology in the knowledge big graph, tc."cname" represents the specific name of a concept; AEdges(PG)is the set of edges representing all the non-DatatypeProperty role assertions of the ontology in the knowledge big graph. (tp."ename," tp."pname") represents the specific name of a role; the bijective functions h_e and t_e associate nodes with edges. The result of the fusion corresponds to the instance layer of the knowledge big graph as shown in Figure 2.

Using the above mapping fusion formula (5)~(7), the semantic constraints of the TBox layer on the ABox layer in the ontology can be mapped to the semantic constraints of the schema layer on the instance layer in the knowledge big graph. All concept names in the ontology are used as labels of nodes in the instance layer of the knowledge big graph. Thus, the association between the schema layer and the instance layer is naturally established in the knowledge big graph. Although additional edges can be used to connect the instance layer and the schema layer in knowledge big graph, it will likely cause tens of millions of nodes in the in-

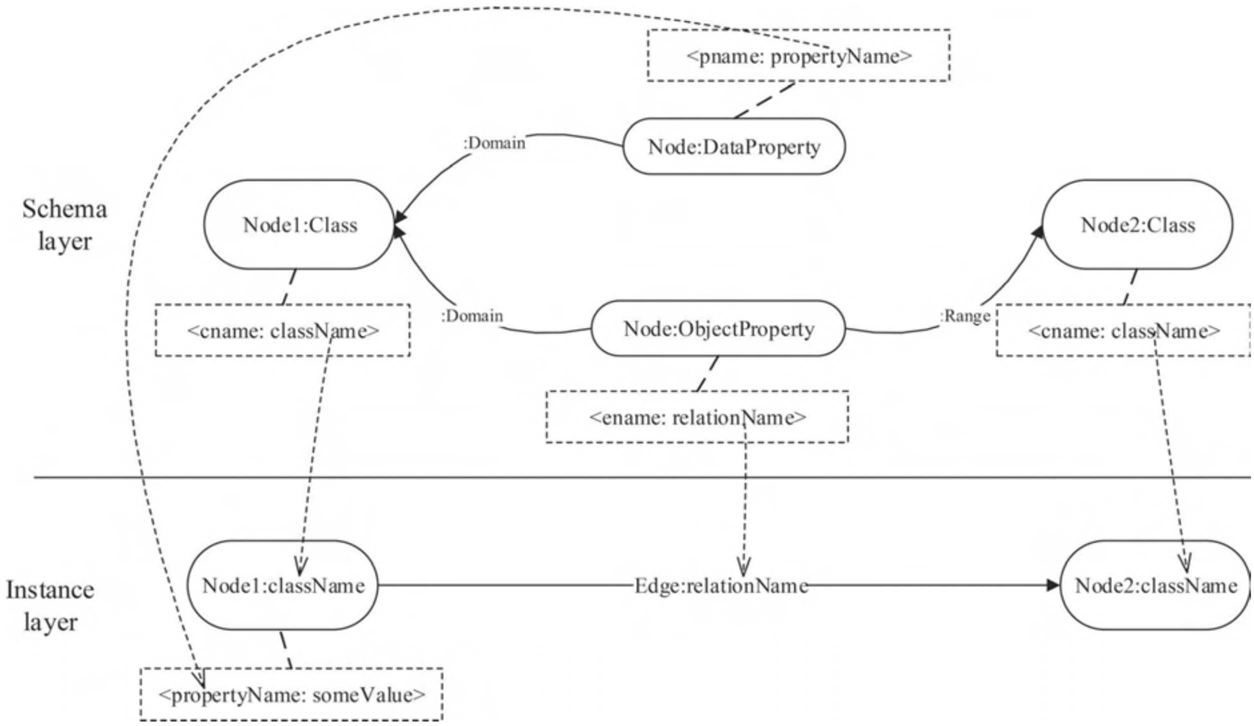

Figure 2. ABox layer fusion.

stance layer to be connected with one node in the schema layer, making the structure of the entire knowledge big graph sparse and unbalanced, which is not conducive to a rapid response to the query. It also does not take advantage of the characteristics of the native storage and the adjacency query of the property graph model and increases storage and query costs. We use the attribute value of the "cname" attribute of the concept node as the label of instance nodes and do not add extra edges to the entire knowledge big graph so that when querying an instance of a concept node, only the subgraph is traversed whose node label is the name of the concept node without having to traverse the entire knowledge big graph.

4.1.3 Fusion of predicate attribute and edge attribute

The attributes of the predicates cannot be directly expressed in RDF triples. As a result, the ABox layer of the ontology based on the RDF triples cannot directly represent the attributes of the relationship. For example, <Jack, marry, Rose> can indicate the marriage relationship between Jack and Rose but cannot directly indicate when Jack and Rose married. The mainstream solution is to treat the marriage relationship as an intermediate entity, such as: <Marriage, bridegroom, Jack>, <Marriage, bride, Rose>, <Marriage, date, 1990>. However, this solution increases the cognitive complexity, the storage costs of the computer, and the complexity of the semantic query. In contrast, we can directly set key-value pairs for edges to represent the attributes of a relationship in a property graph. By combining the predicate attributes of the ontology with the edge attributes in the property graph model, a large number of intermediate entities in an ontology are eliminated and the expressiveness of the ontology model is improved. The recombination fusion formula shown in Box 4.

$\mathrm{AEdges\_pa(PG)}$ is the edge set with predicate attributes in knowledge big graph, and $\mathrm{tp}_2.$"pname" represents the specific role name of the DatatypeProperty role. DatatypeProperty role nodes can be associated with concept nodes or other role nodes through the "Domain" edge in the schema layer of the knowledge big graph. DatatypeProperty role nodes represent the attributes of the other role nodes when they are associated with other role nodes. In the instance layer of the knowledge big graph, the edge attribute name is the attribute value of the "pname" attribute of a DatatypeProperty role node. As shown in Figure 3, the DatatypeProperty role node

Box 4

$$\mathrm{AEdges}_{\mathrm{pa(PG)}} = \{\mathrm{ar} : \mathrm{tp}_1.\mathrm{ename}"<\mathrm{tp}_2.\mathrm{pname}" = \mathrm{someValue} > |\mathrm{tp}_1, \mathrm{tp}_2 \in \mathrm{RoleNodes(PG)}\}. \quad (8)$$

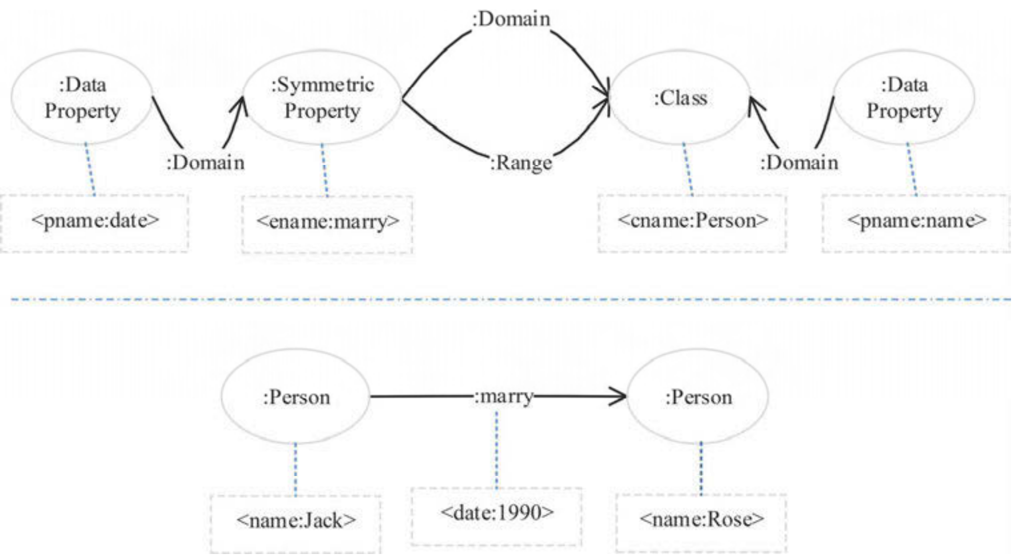

Figure 3. Example of fusion of predicate attribute and edge attribute.

expressing time (the attribute value of the "pname" attribute is "date") associates with the SymmetricProperty role node representing the marriage relationship (the attribute value of the "ename" attribute is "marry") through the "Domain" edge. It means that the marriage relationship has the "date" attribute and the marriage relationship is bidirectional. The "Jack" node and the "Rose" node are associated through the edge with the "marry" label, and this edge has a "date" attribute of which the value is "1990."

4.2 Constraint layer fusion

The function of the ontology is reflected in two aspects. One is the interchangeability of knowledge. By integrating the ontology into the knowledge big graph, the semantics are placed in the data. The description information about the knowledge big graph's structure can be obtained easily by querying the schema layer of the knowledge big graph, making the knowledge big graph self-descriptive, which facilitates its subsequent transfer and utilization. The other aspect is the interoperability between knowledge elements, which is reflected in the semantic application of ontology reasoning. However, the property graph model lacks strong semantic constraints. It is likely to produce incorrect or contradictory data during construction and evolution, which in turn leads to knowledge graphs based on the property graph model being prone to providing incorrect answers in subsequent knowledge services. Therefore, it is necessary to integrate the consistent reasoning function of the ontology into the knowledge big graph to improve the self-checking ability of knowledge big graph, which has important practical significance for the construction and update of the knowledge base.

This paper also focuses on the fusion of knowledge interoperability in ontologies with the predicate attributes of the property graph model to achieve semantically consistent reasoning in the knowledge big graph. The constraint layer fusion consists of two parts: 1) mapping the traditional consistent reasoning of ontology to the path traversal query of the property graph model; and, 2)recombining the ontology semantics and the predicate attribute of the property graph model to make the knowledge big graph capable of consistent reasoning of predicate attributes that the ontology cannot.

4.2.1 Traditional consistent reasoning of ontology

Traditional ontology's consistent reasoning mainly relies on an external reasoning machine, modifying internal algorithms of inference engines or directly calling external reasoning machines to determine whether and where inconsistency problems exist in the ontology knowledge base by finding the minimal unsatisfactory maintaining subset (Liu et al. 2012, Parsia et al. 2005). These methods have insufficient efficiency and stability and cannot deal with large-scale ontology knowledge bases. Existing studies have shown that the methods based on graph traversal is superior to the calculation methods based on inference engines (Fu et al. 2016; Qi et al. 2015; Fu et al. 2014). Taking comprehensive consideration of efficiency and usefulness, we achieve completely domain-independent consistent reasoning by fusing the primitives of the ontology and the path traversal query of the graph. Specifically, it is divided into two levels: 1) consistent reasoning at the schema level of the knowledge big graph, focusing on whether logical contradiction exists in the semantic relationships between conceptual nodes; and, 2) consistent reasoning at the instance level

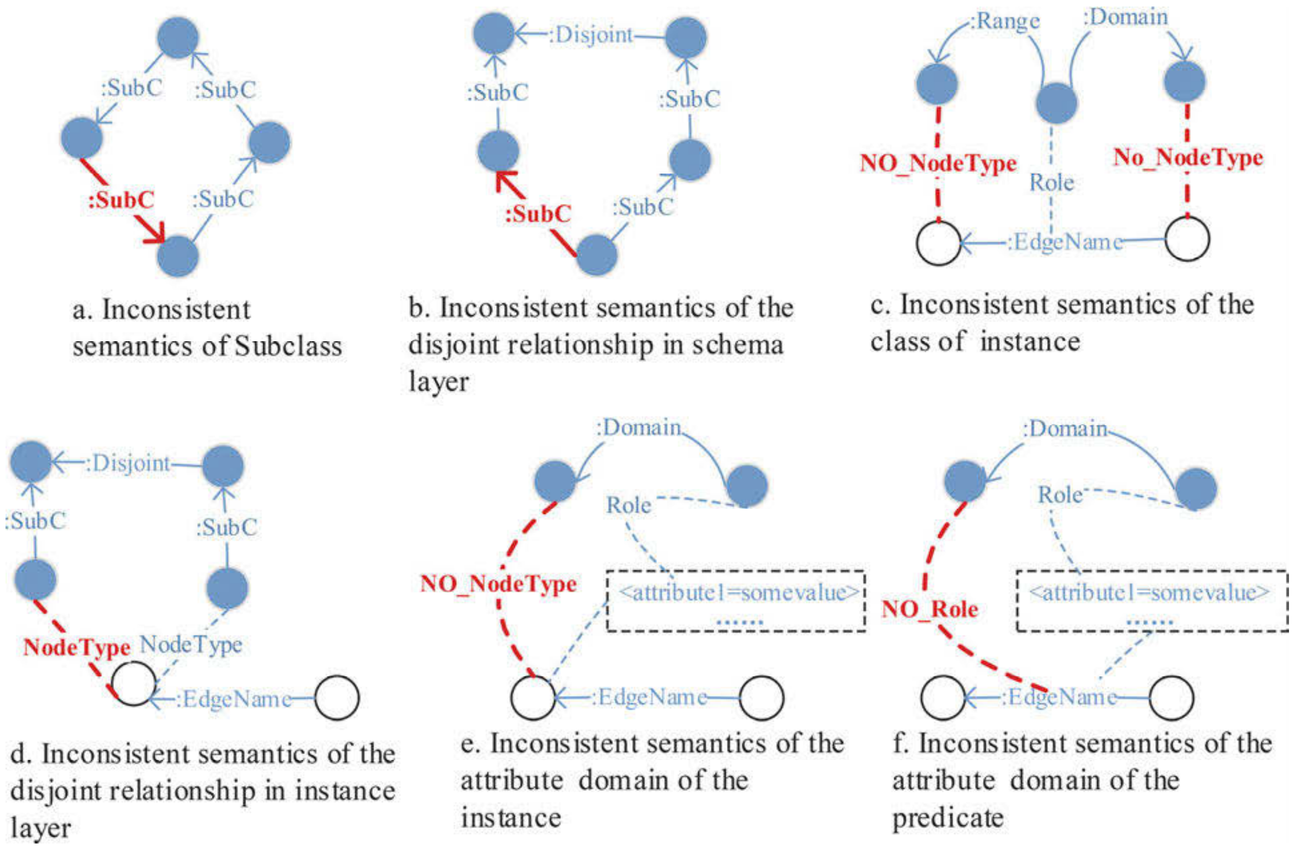

Figure 4. Inconsistent semantics in the knowledge big graph.

of knowledge big graph, focusing on whether the attributes of the instance nodes and the relationship between the instance nodes are consistent with the semantics defined by the schema layer.

(1) Consistent reasoning at the schema level of the knowledge big graph. It mainly includes subclassOf consistency reasoning (see Figure 4a) and disjoint relationship consistency reasoning at the schema level (see Figure 4b). The schema layer of the knowledge big graph is mainly composed of concepts and subclassOf relationships between concepts. There may be inconsistent semantics between two concepts with a subclassOf relationship; that is, it is necessary to check whether the path formed by the subclassOf relationship is looped to complete the subclassOf consistency inference. Disjoint consistency inference at the schema level refers to finding whether two concepts with disjoint relationships have the same subclasses.

(2) Consistent reasoning at the instance level of the knowledge big graph. It mainly includes instance relationship consistency reasoning (see Figure 4c), consistency reasoning of the attribute domain of the instance (see Figure 4e), and consistency reasoning of disjoint relationships at the instance level (see Figure 4d). Instance relationship consistency reasoning is also called object property consistency reasoning. It refers to checking whether the label of the edge in the instance layer and the labels of the two nodes associated with the edge are correctly mapped one by one in the schema layer. Consistency reasoning of the attribute domain of the instance refers to checking whether the attribute domain of the instance node is consistent with the label of the instance node; that is, finding whether the two nodes corresponding to the label of the instance node and the attribute of the instance node are connected through the "Domain" edge in the schema layer. Consistency reasoning of disjoint relationships at the instance level refers to checking whether there is an instance node belonging to two disjoint classes.

4.2.2 Consistent reasoning for predicate attribute

The predicate attribute cannot be directly expressed in the traditional ontology model, and consistent reasoning for predicate attribute is impossible to be performed for the ontology. The attributes of the edge can be used to directly represent the attributes of the predicate in knowledge big graph, which enriches the expression ability of the ontology.

As a result, we implement the consistent reasoning of the domain of predicate attribute and the consistent reasoning of custom domain constraints in the knowledge big graph. Similar to the consistency reasoning of the attribute domain of instance, the consistent reasoning of the domain of predicate attribute refers to checking whether the two nodes corresponding to the label of the edge and the attribute of the edge at the instance layer are connected through the "Domain" edge at the schema layer of the knowledge big graph (see Figure 4f). The consistent reasoning of custom domain constraints refers to checking whether the predicate attribute meets the domain constraint defined by domain experts, such as the sum of shareholding ratios that the same company held by all shareholders cannot exceed one hundred percent in the financial domain.

5.0 Application of the knowledge big graph in financial shareholding structure

The shareholding structure is the foundation of the governance system of financial institutions in the financial field, which determines governance mechanisms of financial institutions and related companies (Lemmon and Lins 2003). The financial ownership network formed by the equity relationship of financial institutions profoundly affects the stability of the entire financial system, such as the occurrence of financial risks (Saunders et al. 1990; Fichtner et al. 2017) and the spread of financial risks (Elliott et al. 2014). Furthermore, analysis of the financial ownership network is the basis of the supervision and risk control of typical financial problems—capital groups, cross-shareholding, actual controller, etc.

We are concerned with China's financial ownership network. The financial ownership network consists of financial institutions and the direct or indirect shareholders' holding relationships of financial institutions. In general, conflicting data is prone to appear during the construction process due to the need to extract data from multiple data sources to build a complete financial ownership network. Moreover, the shareholding structure of enterprises changes frequently over time, which easily leads to data inconsistency in financial ownership network. Therefore, we convert the China's financial ownership network into the knowledge big graph to lay the foundation for the accuracy and efficiency of the subsequent analysis of the financial shareholding structure.

5.1 Financial ownership knowledge big graph

The schema layer of the financial ownership knowledge big graph explained here explicitly describes the semantics of the hierarchical classes and relationships of entities in China's financial shareholding structure while the instance layer mainly shows the entities and relationships in the financial shareholding structure.

According to the TBox layer fusion rules, the schema layer of the financial ownership knowledge big graph constructed is shown in Figure 5. The schema layer stipulates the main categories of financial institutions and the hierarchical categories of civil subjects that are shareholders of the financial institutions. The shareholding property as the object property stipulates that all social subjects can be shareholders in the shareholding structure while the entity held can only be an entity of the type of enterprise or financial institution and their subclasses. The control property is a sub-property of the shareholding and is an asymmetric role; that is, entity B cannot control entity A when entity A controls entity B. The event is the key element of financial risk identification and control. Social subjects as participants in events that contain financial risks are likely to become the media for the spread of financial risks (Petrone and Latora 2018; Poledna et al. 2015).

According to the ABox layer fusion rules, the fragment of the instance layer of the financial ownership knowledge big graph is shown in Figure 6, using the shareholding structure of the China Development Bank as an example. The direct shareholders of the China Development Bank are the Chinese Ministry of Finance, Wutongshu Investment Platform Co., Ltd., Central Huijin Investment Ltd., and the National Council for Social Security Fund. The sum of the shares that they hold in China Development Bank is one. The share and start time are attributes of shareholding edges and control edges, corresponding to the role nodes of "share" and "start time" in the schema layer of the financial ownership knowledge big graph. The labels of the China Development Bank node correspond to the "state policy bank" and "state-owned business" nodes in the schema layer of the financial ownership knowledge big graph.

5.2 Consistent reasoning algorithms for the financial ownership knowledge big graph

We use Cypher query language to describe the consistent inference algorithms for the financial ownership knowledge big graph. Cypher is a user-friendly, declarative property graph query language. The Cypher query used in this paper has the following structure:

```
MATCH <pattern1>
[ WITH <result1> ]
[MATCH <pattern2> ]
[ WHERE <constraint> ]
RETURN <result2> [ as <expression>].
```

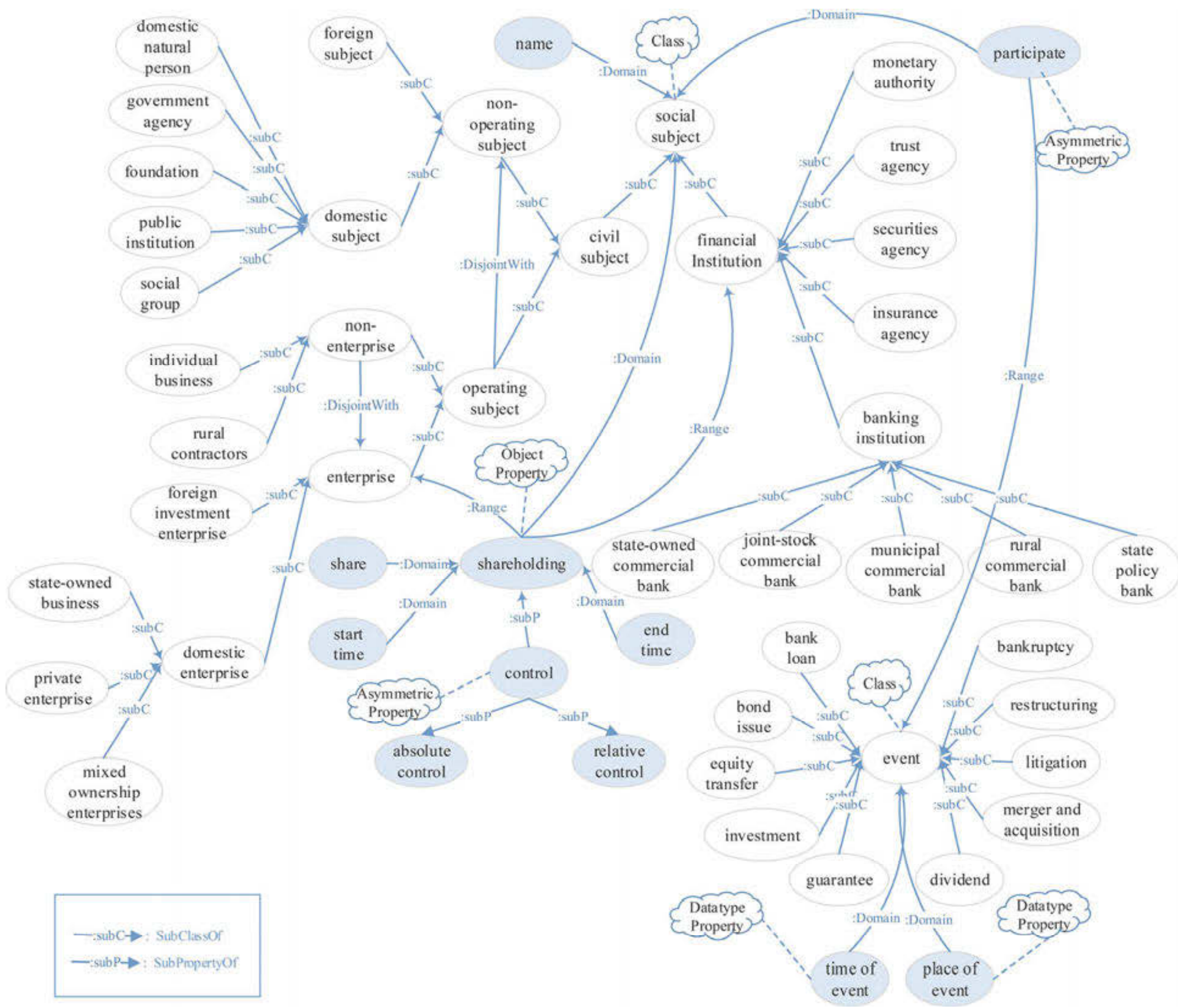

Figure 5. The schema layer of the financial ownership knowledge big graph.

The MATCH clause refers to the path or subgraph structure being queried. The WITH clause uses the result of the previous MATCH clause query as the input of the next MATCH clause query. The WHERE clause is used to conditionally restrict the query process or filter the query result, and the RETURN clause returns the final query result. The MATCH clause in this paper involves indefinite-length path queries. For example, "(n1:Class)-[:SubClassOf*0..]->(n2:Class)" means that the n1 concept node is a direct child of the n2 concept node Class (path length is 1), indirect subclass (path length is greater than 1), or n2 concept node itself (path length is 0). "*0.." means that the path length is at least zero.

The descriptions of the consistent reasoning algorithms implemented with Cypher are shown in Table 2. If the result returned by the consistent reasoning is null, it means there are no inconsistent semantics. Otherwise, the result returned by the consistent reasoning represents the inconsistent semantics, which needs to be subsequently corrected manually. Taking "consistency reasoning of subclassOf" as an example (Table 2a), if n1 is a(n) (indirect) subclass of n2, n2 is a(n) (indirect) subclass of n3, and n1 and n3 are the same conceptual node, then the path will be returned, which contains inconsistent semantics. The Cypher descriptions of the five consistent reasoning algorithms do not involve specific domain vocabularies and are completely domain-independent except for the consistent reasoning algorithm of custom domain constraints.

5.3 Experimental results

The financial ownership knowledge big graph we constructed contains the ownership of 1,432 financial institutions and all enterprises and other entities of the National

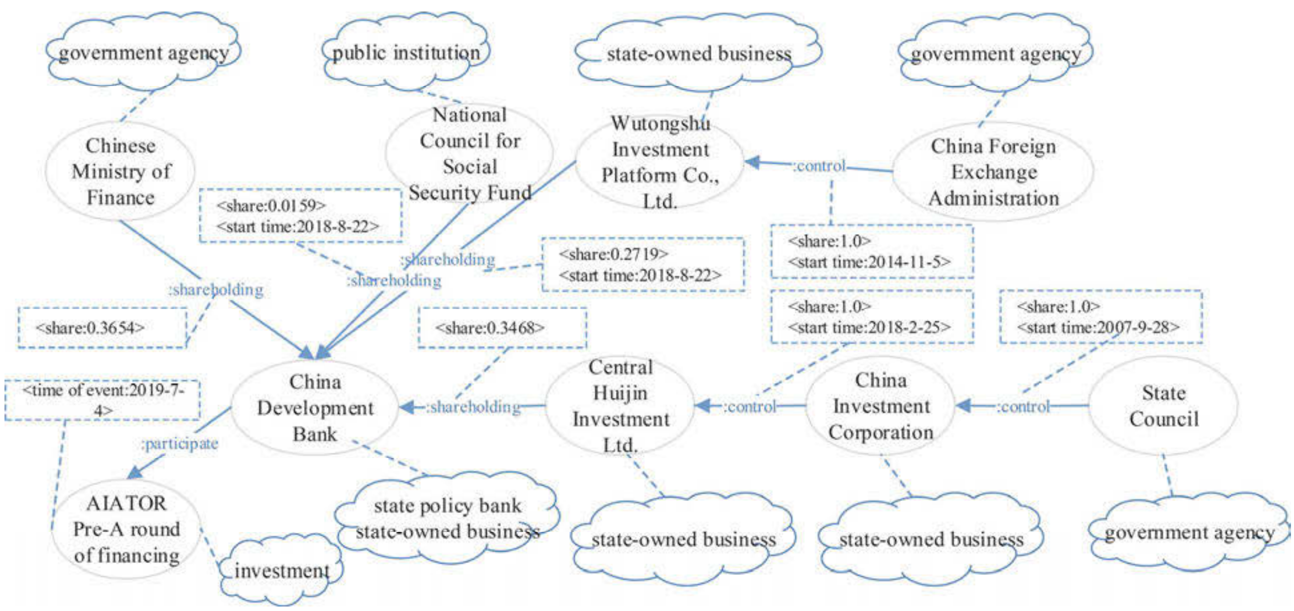

Figure 6. The fragment in the instance layer of the financial ownership knowledge big graph.

Enterprise Credit Information Publicity System of China. There are currently more than eighty million entities and nearly 100 million shareholding edges, covering almost all Chinese enterprises and their shareholders. The original data comes from the National Enterprise Credit Information Database of China and annual reports of Chinese financial institutions. The Neo4j graph database is used to store, query, and reason about the financial ownership knowledge big graph. The query results of the schema layer and instance layer of the financial ownership knowledge big graph in the Neo4j database are shown in Figure 7.

In order to test the accuracy of the consistent inference algorithm, the following inconsistent semantics is added to the financial ownership knowledge big graph in advance:

1. Add
   "civil subject $\xrightarrow{\text{:SubClassof}}$ social group."
   Figure 5 shows that the social group is an indirect subclass of the civil subject, which results in semantic inconsistency of the subclassOf relationship.
2. Add
   "state_owned business $\xrightarrow{\text{:SubClassof}}$ public institution"
   and set the instance node "China Investment Corporation" to have the labels "state-owned enterprise" and "government agency." Figure 5 also shows that state-owned enterprise is an indirect subclass of non-operating subject and government agency and public institutions are indirect subclasses of operating subject. There is a disjoint relationship between non-operating subject and non-operating subject. This results in inconsistent semantics of disjoint classes based on common subclass and inconsistent semantics of disjoint classes based on common instance.
3. Remove all labels of the "Chinese Ministry of Finance" node and the "China Development Bank" node. Figure 6 shows that the Chinese Ministry of Finance is the direct shareholder of the China Development Bank. The domains of shareholding relationships are social subject and its subclasses, and the ranges of shareholding relationships are enterprise and financial institution as well as their subclasses (Figure 5), resulting in inconsistent semantics of instance relationship (shareholding). The Chinese Ministry of Finance node and the China Development Bank node have datatype property ("name"), and the domains of datatype property ("name") are social subject and its subclasses. Therefore, inconsistent semantics of attribute domain of instance are also generated.
4. Change the label of the shareholding edge of the "Central Huijin Investment Ltd." node to the "China Development Bank" node from "shareholding" to "participate." The edge has a datatype property ("share"), and the domains of the datatype property ("share") are the object property ("shareholding") and its sub-properties (see Figure 5), resulting in inconsistent semantics of the domain of predicate attribute.

a. Consistency reasoning of subclassOf

```
MATCH path=(n1)-[:SubClassOf*1..]->(n2:Class)-[:SubClassOf*1..]->(n3)
WHERE n1=n3
RETURN path as SubClassOf_Inconsistency
```

b. Consistency reasoning of disjoint relationship between classes

```
//① check for the inconsistent semantics of the disjoint relationship in schema layer
MATCH path=(n3)-[:SubClassOf*1..]-> (n1:Class)<-[:DisjointWith]-(n2:Class)<-[:SubClassOf*1..]-(n4)
WHERE n3=n4
RETURN n3.cname as SubClass_DisjointWith_Inconsistency,n1.cname as DisClass1,n2.cname as DisClass2
//② check for the inconsistent semantics of the disjoint relationship in instance layer
MATCH (n3)-[:SubClassOf*1..]-> (n1:Class)<-[:DisjointWith]-(n2:Class)<-[:SubClassOf*1..]-(n4)
WITH n3, n4
MATCH (iNode)
WHERE n3.cname in labels(iNode) and n4.cname in labels(iNode)
RETURN iNode.name as Instance_DisjointWith_Inconsistency, n3.cname as DisClass1,n4.cname as DisClass2
```

c. Consistency reasoning of class of instance

```
// ① check for the inconsistent semantics of object property domain
MATCH (n)-[:SubClassOf*0..]->(n1:Class)<-[:Domain]-(r1:ObjectProperty)<-[:SubPropertyOf*0..]-(r)
WITH collect(distinct n.cname) as classLables, r.ename as edgeName
MATCH (iNode)-[edge]->()
WHERE type(edge)=edgeName and none( iLable in labels(iNode) where iLable in classLables )
RETURN iNode.name as instance_node,edgeName as ObjectProperty_DomainInconsistency,classLables as TrueClasses
// ② check for the inconsistent semantics of object property range
MATCH (n)-[:SubClassOf*0..]->(n1:Class)<-[:Range]-(r1:ObjectProperty)<-[:SubPropertyOf*0..]-(r)
WITH collect(distinct n.cname) as classLables,r.ename as edgeName
MATCH ()-[edge]->(iNode)
WHERE type(edge)=edgeName and none(iLable in labels(iNode) where iLable in classLables )
RETURN iNode.name as instance_node, edgeName as ObjectProperty_RangeInconsistency,classLables as TrueClasses
```

d. Consistency reasoning of datatype property domain of instance

```
MATCH (n)-[:SubClassOf*0..]->(n1:Class)<-[:Domain]-(p1:DatatypeProperty)<-[:SubPropertyOf*0..]-(p)
WITH collect(distinct n.cname) as classLables, p.pname as propertyName
MATCH (iNode) WHERE propertyName in keys(iNode) and none(iLable in labels(iNode) where iLable in classLables)
RETURN iNode.name as instance_node, propertyName as Instance_DataProperty_DomainInconsistency,classLables as
TrueClasses
```

e. Consistency reasoning of the datatype property domain of predicate

```
MATCH (r)-[:SubPropertyOf*0..]->(r1:ObjectProperty)<-[:Domain]-(p1:DatatypeProperty)<-[:SubPropertyOf*0..]-(p)
WITH collect(distinct r.ename) as edgeNames,p.pname as propertyName
MATCH ()-[edge]->() WHERE propertyName in keys(edge) and not type(edge) in edgeNames
RETURN id(edge) as edgeid, propertyName as Edge_DataProperty_DomainInconsistency,edgeNames as TrueClasses
```

f. Consistency reasoning of custom domain constraints about the attribute of predicate

```
// check for companies with more than 1 sum of shares held or controlled
MATCH (n1)-[r1:shareholding|:control]->(n2)
WITH n2, sum(toFloat(r1.share)) as sumshare WHERE sumshare > 1 WITH n2
MATCH path=(n1)-[r1:shareholding|:control]->(n2)
RETURN n2.name as Firm_with_EquityOver1_Inconsistency, sumshare
```

Table 2. The Cypher description of the consistent reasoning algorithms.

5. Add
“Ministry of Finance $\xrightarrow{:control}$ Chinese Ministry of Finance”
and set the value of the attribute (share) of the control edge to 0.3654. This results in the China Development Bank being controlled or held by a ratio of more than one, which generates inconsistent semantics of predicate attribute custom domain constraints.

The results of the implementation in the Neo4j database using the consistent reasoning algorithms designed in this paper are shown in Figure 8. The consistent reasoning algo-

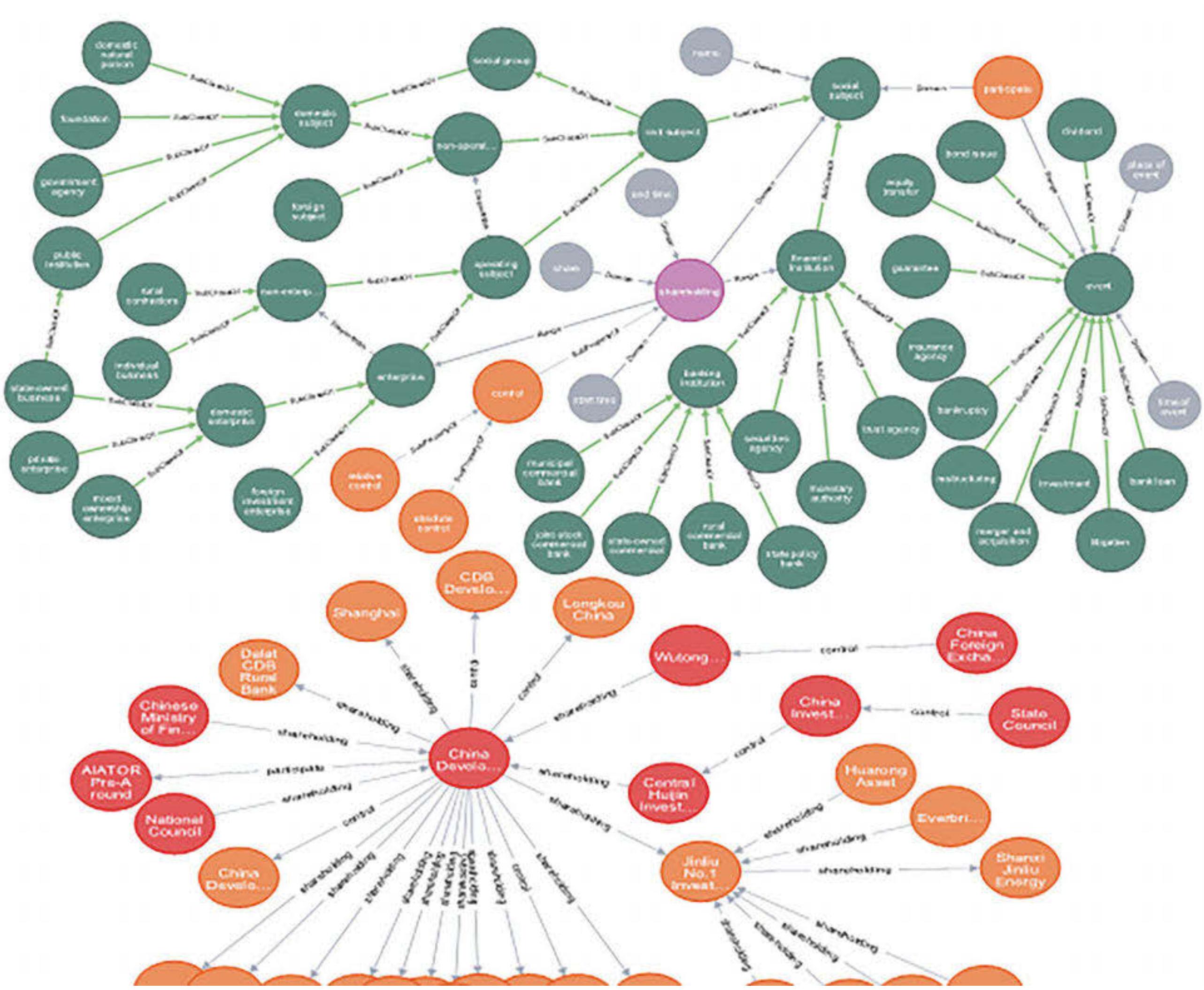

Figure 7. Visual query results of financial ownership knowledge big graph.

rithms reasoned out all the pre-added inconsistent semantics, which proves the logical correctness of the knowledge big graph and the validity of the consistent reasoning algorithms based on the knowledge big graph. According to the results of the inconsistent reasoning shown in Figure 8, further correction can be performed to remove inconsistent semantics, ensure the logical consistency of the knowledge big graph, and lay a quality foundation for subsequent knowledge services based on the knowledge big graph.

6.0 Conclusion

The huge amount of knowledge, the complexity of knowledge semantics, and the frequent updates of knowledge have brought new problems to knowledge representation, organization, storage, and utilization in big data environment. These facts seriously hinder the timeliness and accuracy of subsequent knowledge services. To this end, the paper first proposes the ontology and property graph fusion framework (OPGFF), combining the precise description of the ontology model and the native graph characteristics of the property graph model from the perspective of knowledge fusion. Then, the construction of the financial ownership knowledge big graph and semantic reasoning are performed. The OPGFF framework proposed in this paper can be applied to general, large-scale knowledge organizations, such as application in the field of medical biology, and can be new a perspective for the construction and utilization of knowledge bases. In addition, the schema layer of the financial ownership knowledge big graph and the consistent reasoning algorithms provide semantic analysis tool and semantic data quality constraints for subsequent financial risk discovery.

Generic ontology model contains other semantic relationships in addition to the basic semantic relationships selected in the paper. Therefore, the knowledge big graph

"SubClassOf_InconsistencyPath"
"social group->domestic subject->non-operating subject->civil subject->social group"

"SubClass_DisjointWith_Inconsistency"	"DisClass1"	"DisClass2"
"state-owned business"	"non-operating subject"	"operating subject"

"Instance_DisjointWith_Inconsistency"	"DisClass1"	"DisClass2"
"China Investment Corporation"	"government agency"	"state-owned business"

"instance_node"	"ObjectProperty_DomainInconsistency"	"TrueClasses"
"Chinese Ministry of Finance"	"shareholding"	["social subject","civil subject","operating subject","enterprise",

"instance_node"	"ObjectProperty_RangeInconsistency"	"TrueClasses"
"China Development Bank"	"shareholding"	["financial Institution","banking institution","state policy bank",

"instance_node"	"Instance_DataProperty_DomainInconsistency"	"TrueClasses"
"China Development Bank"	"name"	["social subject","civil subject","operating subject","enterprise",
"Chinese Ministry of Finance"	"name"	["social subject","civil subject","operating subject","enterprise",

"edgeid"	"Edge_DataProperty_DomainInconsistency"	"TrueClasses"
774111	"share"	["shareholding","control","absolute control","relative control"]

"Firm_with_EquityOver1_Inconsistency"	"sumshare"
"China Development Bank"	1.0186

Figure 8. Experimental results of consistency reasoning.

does not contain all the ontology semantics (such as equivalent anonymous classes or disjoint attributes). Subsequent research can add more ontology semantics to the knowledge big graph according to actual needs. The consistent reasoning algorithms we proposed can be further expanded or improved in the future, such as by combining more efficient graph mining algorithms and cognitive computing technologies (Chen et al. 2019).

References

Bizer, Christian, Tom Heath and Tim Berners-Lee. 2009. "Linked Data-The Story So Far." *International Journal on Semantic Web and Information Systems* 5: 1-22.

Bragato Barros, Thiago Henrique and Renato Tarciso Barbosa de Sousa. 2019. "Archival Science and Knowledge Organization: Relationships and Challenges in a Digital World." *Knowledge Organization* 46: 493-501.

Chen Min, Wei Li, Giancarlo Fortino, Yixue Hao, Long Hu and Iztok Humar. 2019. "A Dynamic Service Migration Mechanism in Edge Cognitive Computing." *ACM Transactions on Internet Technology* 19: 1-15.

Ching, Avery, Sergey Edunov, Maja Kabiljo, Dionysios Logothetis and Sambavi Muthukrishnan. 2015. "One Trillion Edges: Graph Processing at Facebook-scale." In *Proceedings of the 41st International Conference on Very Large Data Bases Kohala Coast Hawaii*, ed, Chen Li and Volker Markl. Proceedings of the VLDB Endowment 8. VLDB Endowment, 1804-15. http://www.vldb.org/pvldb/vol8/p1804-ching.pdf

Constantinov, Calin, Cosmin M. Poteras and Mihai L. Mocanu. 2016. "Performing Real-time Social Recommendations on a Highly-available Graph Database Cluster." In *17th International Carpathian Control Conference May 2016 Tatranska Lomnica, Slovakia*. Piscataway, NJ: IEEE, 116-21. doi:10.1109/CarpathianCC.2016.7501078

Cui, Licong, Shiqiang Tao and Guo-Qiang Zhang. 2016. "Biomedical Ontology Quality Assurance Using a Big Data Approach." *ACM Transactions on Knowledge Discovery from Data* 10: 1-28.

Drakopoulos, Georgios, Andreas Kanavos, Dimitrios Tsolis, Phivos Mylonas and Spyros Sioutas. 2017. "Towards a Framework for Tensor Ontologies over Neo4j: Representations and Operations." In *8th International Conference on Information, Intelligence, Systems & Applications August 2017 Larnaca, Cyprus*. Piscataway, NJ: IEEE, 1-6. doi:10.1109/IISA.2017.8316441

Dutta, Biswanath, Fausto Giunchiglia and Vincenzo Maltese. 2014. "From Knowledge Organization to Knowledge Representation." *Knowledge Organization* 41: 44-56.

Elliott, Matthew, Benjamin Golub and Matthew O. Jackson. 2014. "Financial Networks and Contagion." *The American Economic Review* 104: 3115-53.

Fichtner, Jan, Eelke M. Heemskerk and Javier Garcia-Bernardo. 2017. "Hidden Power of the Big Three? Passive Index Funds, Re-Concentration of Corporate Ownership and New Financial Risk." *Business and Politics* 19: 298-326.

Fu, Xuefeng, Guilin Qi, Yong Zhang and Zhangquan Zhou. 2016. "Graph-based Approaches to Debugging and Revision of Terminologies in DL-Lite." *Knowledge Based Systems* 100: 1-12.

Fu, Xuefeng, Yong Zhang and Guilin Qi. 2014. "GrOD: Graph-based Ontology Debugging System." In *The Semantic Web and Web Science: 8th Chinese Conference, CSWS 2014, Wuhan, China, August 8-12, 2014, Revised Selected Papers*, ed. Dongyan Zhao, Jianfeng Du, Haofen Wang, Peng Wang, Donghong Ji and Jeff Z. Pan. Communications in Computer and Information Science 480. Berlin: Springer, 87-94. doi:10.1007/978-3-662-45495-4_8

Gong, Faming, Yuhui Ma, Wenjuan Gong, Xiaoran Li, Chantao Li and Xianbing Yuan. 2018. "Neo4j Graph Database Realizes Efficient Storage Performance of Oilfield Ontology." *PLOS ONE* 13: 1-16.

Hartig, Olaf. 2014. "Reconciliation of RDF* and Property Graphs." *arXiv*:1409.3288

Hecht, Robin and Stefan Jablonski. 2011. "NoSQL Evaluation: A Use Case Oriented Survey." In *Proceedings 2011 International Conference on Cloud and Service Computing Hong Kong, China 12 - 14 December 2011*. Piscataway, NJ: IEEE, 336-41. doi:10.1109/CSC.2011.6138544

Hjørland, Birger. 2008. "What is Knowledge Organization (KO)?" *Knowledge Organization* 35: 86-101.

Konno, Takahiro, Runhe Huang, Tao Ban and Chuanhe Huang. 2017. "Goods Recommendation Based on Retail Knowledge in A Neo4j Graph Database Combined with an Inference Mechanism Implemented in Jess." In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation June 2018 San Francisco, CA*. Piscataway, NJ: IEEE, 1-8. doi:10.1109/UIC-ATC.2017.8397433

Krötzsch, Markus, Maximilian Marx, Ana Ozaki and Veronika Thost. 2017. "Attributed Description Logics: Ontologies for Knowledge Graphs." In *The Semantic Web – ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I,* ed. Claudia d'Amato, Miriam Fernandez, Valentina Tamma, Freddy Lecue, Philippe Cudré-Mauroux, Juan Sequeda, Christoph Lange and Jeff Heflin. Lecture Notes in Computer Science 10587. Cham: Springer, 418-35.

Krötzsch, Markus and Veronika Thost. 2016. "Ontologies for Knowledge Graphs: Breaking the Rules." In *The Semantic Web - ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I,* ed. Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck and Yolanda Gil. Lecture Notes in Computer Science 9981. Cham: Springer, 376-92. doi:10.1007/978-3-319-46523-4_23

Lemmon, Michael L. and Karl V. Lins. 2003. "Ownership Structure, Corporate Governance and Firm Value: Evidence from the East Asian Financial Crisis." *Journal of Finance* 58: 1445-68.

Liu, Bo, Jianquiang Li and Yu Zhao. 2012. "Repairing and Reasoning with Inconsistentand Uncertain Ontologies." *Advances in Engineering Software* 45: 380-90.

Lígia Maria Arruda Café and Renato Rocha Souza. 2017. "Sentiment Analysis and Knowledge Organization: An Overview of the International Literature." *Knowledge Organization* 44: 199-214.

Miller, Justin J. 2013. "Graph Database Applications and Concepts with Neo4j." In *2013 Proceedings of the Southern Association for Information Systems Conference, May 2013 Atlanta, GA*. AIS eLibrary, 135-40.

Pareja-Tobes Pablo, Raquel Tobes, Marina Manrique, Eduardo Pareja and Eduardo Pareja-Tobes. 2015. "Bio4j: A High-performance Cloud-enabled Graph-based Data Platform." *BioRxiv.* doi:10.1101/016758

Parsia, Bijan, Evren Sirin and Aditya Kalyanpur. 2005. "Debugging OWL Ontologies." In *Proceedings of the 14th International Conference on World Wide Web May 2005 New York, USA*. Association for Computing Machinery, 633-40.

Petrone, Daniele and Vito Latora. 2018. "A Dynamic Approach Merging Network Theory and Credit Risk Techniques to Assess Systemic Risk in Financial Networks." *Scientific Reports* 8: 1-14.

Pham, Phu, Thuc Nguyen and Phuc Do. 2018. "Computing Domain Ontology Knowledge Representation and Reasoning on Graph Database." *Advances in Intelligent Systems and Computing* 672: 765-75.

Poledna, Sebastian, José Luis Molina-Borboa, Serafín Martínez-Jaramillo, Marco van er Leij and Stefan Thurner. 2015. "The Multi-Layer Network Nature of Systemic Risk and Its Implications for the Costs of Financial Crises." *Journal of Financial Stability* 20: 70-81.

Preece, Alun D., Kit-ying Hui, W. A. Gray, Philippe Marti, Trevor Bench-Capon, Dean Jones and Zhan Cui. 2000. "The KRAFT Architecture for Knowledge Fusion and Transformation." *Knowledge Based Systems* 13: 113-20.

Qi, Guilin, Zhe Wang, Kewen Wang, Xuefeng Fu and Zhiquiang Zhuang. 2010. "Approximating Model-based A Box Revision in DL-Lite: Theory and Practice." In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence January 2015 Austin, Texas*. AAAI Press, 254-60.

Rodriguez, Marko A. and Peter Neubauer. 2010. "Constructions from Dots and Lines." *Bulletin of the American Society for Information Science and Technology* 36: 35-41.

Saunders, Anthony, Elizabeth Strock and Nickolaos G.Travlos. 1990. "Ownership Structure, Deregulation and Bank Risk Taking." *Journal of Finance* 45: 643-654.

Silva, Nitai B., Ing-Ren Tsang, George D. Cavalcanti and Ing-Jyh Tsang 2010. "A Graph-based Friend Recommendation System Using Genetic Algorithm." *In IEEE Congress on Evolutionary Computation. Congress on Evolutionary Computation*, 1-7. doi:10.1109/CEC.2010.5586144

Smirnov, Alexander, Tatiana Levashova and Nikolay Shilov. 2015. "Patterns for Context-based Knowledge Fusion in Decision Support Systems." *Information Fusion* 21: 114-29.

Stuart, David. 2016. *Practical Ontologies for Information Professionals*. London: Facet.

Tomaszuk, Dominik. 2016. "RDF Data in Property Graph Model." In *Metadata and Semantics Research: 10th International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings*, ed. Emmanouel Garoufallou, Imma Subirats Coll, Armando Stellato and Jane Greenberg. Communications in Computer and Information Science 672. Cham: Springer, 104-15.