A Graph Derivation Based Approach for Measuring and Comparing Structural Semantics of Ontologies

Yinglong Ma, Ling Liu, *Senior Member, IEEE,* Ke Lu, Beihong Jin, and Xiangjie Liu

Abstract—Ontology reuse offers great benefits by measuring and comparing ontologies. However, the state of art approaches for measuring ontologies neglects the problems of both the polymorphism of ontology representation and the addition of implicit semantic knowledge. One way to tackle these problems is to devise mechanism for ontology measurement that is stable, the basic criteria for automatic measurement. In this paper, we present a graph derivation representation based approach (GDR) for stable semantic measurement, which captures structural semantics of ontologies and addresses those problems that cause unstable measurement of ontologies. This paper makes three original contributions. First, we introduce and define the concept of semantic measurement and the concept of stable measurement. We present the GDR based approach, a three-phase process to transform an ontology to its GDR. Second, we formally analyze important properties of GDRs based on which stable semantic measurement and comparison can be achieved successfully. Third but not the least, we compare our GDR based approach with existing graph based methods using a dozen real world exemplar ontologies. Our experimental comparison is conducted based on nine ontology measurement entities and distance metric, which stably compares the similarity of two ontologies in terms of their GDRs.

Index Terms—Ontology, Ontology measures, Ontology comparison, Ontology reuse.

1 Introduction

We have witnessed continued and explosive growth in ontology-based applications over the last decade. Ontologies have been widely applied in many fields such as knowledge management [1]–[3], Semantic Web [4], information integration [5]–[7], and semantic search [8]–[10], etc. A well-known advantage of ontologies is that they provide a knowledge-sharing infrastructure that supports the representation and sharing of domain knowledge by formalizing meaning of content and information. As the size and the number of ontologies continue to increase [11], the reuse and the stable measurement of ontologies offer several important benefits. First, the effort of constructing new ontologies can be significantly reduced by reusing existing ontologies instead of starting from scratch [12]–[14]. Another benefit of ontology reuse is its potential to significantly facilitate the data interoperability in heterogeneous information systems by sharing a common ontology [15]–[20].

A fundamental operation in ontology reuse is to compute similarity and dissimilarity among ontological entities such that one can establish certain level of correlation between ontological entities used in different ontologies by predefined measures and semantic comparison. Most of the existing ontology measures to date are defined based on graphical models representing ontological structure, which have demonstrated potential and initial success in measurement and comparison of the semantics of ontologies [21]–[35].

However, some important open issues for measuring and comparing ontologies remains to be resolved. For example, the problem of *polymorphism of ontology representation* has been neglected in most of the literature to date, which refers to the fact that the same semantic knowledge can be explicitly represented using different semantic structures. Although most of the ontology languages, such as OWL, provide a shared vocabulary, how to use such vocabulary in modeling domain knowledge still depends on domain scientists and specific applications [36]. Often, the implicit kinship of components and structures is quite different and is derived from the original ontologies based on the complex concepts which are iteratively defined by other complex concepts. These and other semantic derivation cases are the common causes for the problem of polymorphism of ontology representation. Such problems make different semantic knowledge incomparable and often lead to meaningless ontology measurement and comparison. Furthermore, some other technical barriers, such as non-direct relations with transitive property and cycles of inheritance, can become the hidden causes of unstable semantic measurement, which can make the

- *Y. Ma and X. Liu are with the School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China (E-mail:yinglongma@gmail.com).*
- *L. Liu is with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA. E-mail:lingliu@cc.gatech.edu.*
- *K. Lu is with University of Chinese Academy of Sciences, Beijing 100049, China. (Email:luk@gucas.ac.cn)*
- *B. Jin is with the Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (Email:beihong@iscas.ac.cn)*

problem of polymorphism much harder to tackle. Third but not the least, we also need the basic criteria to avoid or minimize double counting in ontology measurement technology [37] in order to obtain meaningful and stable ontology measurement results.

With these problems in mind, in this paper, we propose a graph derivation (GDR) based approach for stable semantic measurements of ontologies in three phases. In Phase 1, we generate a GDR for each concept in a given ontology by recursively applying a series of derivation. In Phase 2, we utilize the integration operations to merge multiple GDRs to produce an initial integrated GDR for the given ontology. In Phase 3, we generate the complete GDR representation for the given ontology by treating those relations that cause unstable semantic measurements, such as the problems of non-direct transitive relations, cycles of inheritance, and double counting. We argue that the GRD based approach will significantly facilitate reliable ontology measurement and comparison. This paper makes three original contributions.

- We formally define the concepts of semantic measurement and stable measurement and develop a graph derivation representation (GDR) based approach, which recursively transform an ontology to its GDR by a series of derivation rules.
- We introduce two classes of GDR treatments to polymorphism of ontology representation for automatic and reliable measurement and comparison of the structural semantics of ontologies.
- We provide the formal analysis of the important properties of GDR. We experimentally compare our GDR approach with existing graph based method in terms of nine ontology measurement entities over a dozen real world exemplar ontologies. A GDR based graph isomorphism approach is also used to stably compare the similarity of two ontologies.

The remainder of this paper is structured as follows. In Section 2, we review the related work. Section 3 presents the basic notions about ontology representation, definitions of semantic and stable measurements, and the definition of GDR. Section 4 describes gives an overview of the GDR based approach. Sections 5 and 6 discuss how to generate and treat the GDR of an ontology respectively. We formally analyze important properties of GDR in Section 7 and report our experimental comparison of our GDR approach with existing graph based approach for stable ontology measurement and comparison in Section 8. Section 9 concludes the paper with a summary and an outline of the future work.

2 Related Work

2.1 Measuring and Comparing Ontologies

An ontology measure is an indicator that is used to reflect some quality properties of ontologies. Ontology measurement is the process or the activities of measuring ontologies according to the definitions of ontology measures.

Many of the existing ontology measures consider only the explicit semantics of ontologies. They were used to compare similarity of ontological entities and structures explicitly expressed in ontologies. A cluster-based measure was proposed in [23], which combines the minimum path length and the taxonomical depth and defines clusters for each of the branches in the hierarchy w.r.t the root node. An ontology-based measure utilizing taxonomical features was proposed in [24] without using tuning parameters to weight the contribution of potentially scarce semantic features. In the context of computing semantic similarity, [25] adopted a similarity function to determine similar entity classes by a matching process based on synonym sets, semantic neighborhoods, and distinguishing features. [21] computed the similarity of two concepts by considering the relevant super-concepts and sub-concepts of the two concepts residing in their taxonomies. [22] presented several novel measures for computing the similarity of two gene products with graph-based ontology terms annotated by common taxonomy terms. Some quality measures [26]–[34] were also proposed to measure and evaluate certain ontology quality properties such as cohesion, complexity, and richness, and so forth.

The other measures consider the excavation of implicit semantic information residing in ontologies, and the complete semantic information of ontologies. [35] proposed some measures for evaluating ontology complexity by excavating implicit kinships between concepts. [38] proposed several measures for ontology cohesion measurement by using implicit semantic information. [39] also presented a set of semantic measures for measuring cohesion and coupling in modular ontologies, by which both explicitly asserted knowledge and the implied knowledge derived from the explicitly represented knowledge are considered for semantic measurement.

Most of the existing measures proposed are used to measure and compare the structural semantics of ontologies. Structural semantics is often described by graphs, where classes and relations are often modeled as nodes and edges, respectively. It is a natural way to compare ontologies in terms of comparing graphs that represent ontologies. However, semantic measurement neglects the polymorphism of ontology representation, which inevitably causes the problem, i.e., multiple graphs possibly exist for representing the same ontologies. Reliable ontology measurement is the precondition on which the meaningful and useful ontology comparison and evaluation can be made [40], [41]. However, no clear solution exists for handling polymorphism of ontology representation. In this paper, we define a solution of stable semantic measurement to handle polymorphism of ontology representation for ontology measurement and comparison.

2.2 Graphical Ontology Representation

Recently, a few graphical models for ontologies were proposed. Unified Modeling Language (UML) together with its associated Object Constraint Language (OCL) is sometimes used as the graphical model of ontologies [42]. UML is suitable for representing explicit taxonomical information instead of implicit non-taxonomic relationship. The emerging semantic link network (SLN) [43] is a description of semantic relations among objective existences. The main idea in using SLN is to pursue semantic richness instead of semantic correctness. [44] presented an interesting work that uses the notion of description graph model to address the problem of insufficient expressivity of describing explicit structured objects in ontological knowledge bases. However, this approach to semantic measurement fails to explicitly express implicit semantic constraints. [45] proposed an ordered binary decision diagram (OBDD) method by generalization of binary decision trees, in which every ontology concept is converted into its negation normal form (NNF), and ontological knowledge base needs to be further flattened. However, OBDDs fail to support some important features of ontologies, such as fanouts/fanins of concepts, etc., thus many of the existing ontology measures are not applicable to OBDDs.

A graphical model for semantic measurement should have the following features. 1) It can explicitly express semantic knowledge including the implicit kinship of concepts and non-taxonomic relationships. 2) The existing ontology measures must be still applicable to the model. 3) There is no the problem of polymorphism of ontology representation in the model. 4) It can satisfy basic criteria in measurement technology so that semantic measurement can be automatically made based on the model. However, to the best of our knowledge, existing graphical models for ontologies fail to satisfy all these requirements. In this paper, we present a novel graph derivation representation (GDR) based approach, which is the first one meeting all the four requirements for stable semantic measurements.

3 Definitions and Notations

3.1 Ontology Languages

Different ontology languages have different levels of expressivity provided with different constructors. As for the OWL language [46], it provides three increasingly expressive sublanguages: OWL Lite, OWL DL and OWL Full (ordered by increasing expressivity). Description Logics (DL) [44] are the dialects of OWL, and have a formal underpinning. Ontologies typically consist of a number of classes, properties, individuals, and axioms. Classes and properties in OWL respectively correspond to concepts and roles in DL. Axioms in both OWL and DL have the same meaning. Individuals are the instances of concepts/classes.

Ontology languages based on DL represent ontologies as 'TBoxes' (terminological boxes) containing inclusions between complex concepts over the vocabulary. The repositories consisting of data sets of instances of concepts and relations are typically modeled as 'ABoxes' (assertion boxes). In TBoxes, the basic descriptions are atomic concepts and atomic roles. Complex concepts are built by applying the DL constructors such as intersection ($\sqcap$), union ($\sqcup$), negation ($\neg$), value restriction ($\forall R.C$) and existential quantification ($\exists R.C$), etc. An axiom is of the form $C \sqsubseteq D$ or $C \equiv D$, where C and D are concepts. $C \sqsubseteq D$ iff C is subsumed by D. $C \equiv D$ iff $C \sqsubseteq D$ and $D \sqsubseteq C$. An ABox represents the facts by instance assertions of the form $C(a)$ or $R(a,b)$, where C is a concept, R is a role, and a and b are instances.

To illustrate polymorphism of ontology representation, we give the descriptions of two example ontologies in DL language (i.e., Examples 1 and 2), where all the axioms/assertions are labeled by different numbers for referring them. They in fact represent the same semantic ontology knowledge in the same domain. The difference between them is the use of different combinations of subsumption and constructors because of the powerful expressivity of the OWL DL language.

Example 1. *TBox*={*1.Researcher* $\sqsubseteq$ *People*, *2.Researcher* $\equiv$ *Professor* $\sqcup$ *PhD*, *3.Prof_with_PhD* $\equiv$ *Professor* $\sqcap$ *PhD*, *4.PhDStudent* $\sqsubseteq$ *Student*, *5.Student* $\equiv$ $\exists$ *register.Dept* $\sqcap$ $\exists$*take.Course*, *6.PhDStudent* $\sqsubseteq$ $\forall$*advisedBy.Professor*, *7.Researcher* $\sqsubseteq$ *ScientificPersonnel*, *8. Prof_with_PhD* $\sqsubseteq$ *Researcher*, *9. ScientificPersonnel* $\sqsubseteq$ *Researcher*}.

ABox={*10.register*$(John, CS)$, *11.take*$(John, Java)$, *12.Dept*(CS), *13.Course* $(Java)$}.

Example 2. *TBox*={*1.Researcher* $\sqsubseteq$ *People*, *2.Researcher* $\sqsubseteq$ *Professor* $\sqcup$ *PhD*, *3.PhD* $\sqsubseteq$ *Researcher*, *4.Professor* $\sqsubseteq$ *Researcher*, *5.Prof_with_PhD* $\sqsubseteq$ *Professor*, *6.Prof_with_PhD*$\sqsubseteq$*PhD*, *7.Professor*$\sqcap$*PhD*$\sqsubseteq$ *Prof_with_PhD*, *8.PhDStudent* $\sqsubseteq$ *Student*, *9.Student* $\equiv$ $\exists$ *register.Dept* $\sqcap$ $\exists$*take.Course*, *10.PhDStudent* $\sqsubseteq$ $\forall$*advisedBy.Professor*, *11.Researcher* $\equiv$ *ScientificPersonnel*, *12. Prof_with_PhD* $\sqsubseteq$ *Researcher*}.

ABox={*13.register*$(John, CS)$, *14.take*$(John, Java)$, *15.Dept*(CS), *16. Course*$(Java)$}.

3.2 Definitions of Measurements

A semantic description of Ontology $\mathcal{O}$ includes not only the explicitly expressed information in $\mathcal{O}$ but also some implicit information derived from the explicitly represented knowledge. Due to polymorphism of ontology representation, the same semantic knowledge residing in $\mathcal{O}$ can be explicitly expressed by multiple semantic descriptions. Let $Sem(\mathcal{O})$ be the set of semantic descriptions of Ontology $\mathcal{O}$.

Definition 1. ***(Semantic Measurement)*** *A semantic measurement* SEM_m *of Ontology* $\mathcal{O}$ *w.r.t a measure* m *is a mapping* SEM_m: $Sem(\mathcal{O}) \rightarrow \mathbf{R}$, *where* $\mathbf{R}$ *is a nonempty set of real numbers.*

Definition 2. ***(Stable Measurement)*** *A semantic measurement* SEM_m *of Ontology* $\mathcal{O}$ *is a stable measurement if and only if all the* $SEM_m(s)$ *are equal for any* $s \in Sem(\mathcal{O})$.

3.3 Graph Derivation Representation

The main cause of polymorphism of ontology representation is that there is a lack of uniform ontology representation model for stable ontology measurement and comparison. Existing ontology languages and models focus more on expressiveness rather than stable ontology measurement. We develop the graph derivation representation (GDR) based approach for measuring and comparing structural semantics of ontologies, aiming towards the goal of uniform ontology representation.

Definition 3. ***(Graph Derivation Representation, GDR)*** *A graph derivation representation of Ontology $\mathcal{O}$, denoted as $G_{\mathcal{O}}=(V_{\mathcal{O}}, E_{\mathcal{O}}, \rho, \lambda, \eta)$, is a directed labeled graph, where*
— $V_{\mathcal{O}}$ is a finite set of vertices, where each vertex is a unique positive integer.
— $E_{\mathcal{O}} \subseteq V_{\mathcal{O}} \times V_{\mathcal{O}}$ is a set of edges.
— $\rho : \mathcal{C} \rightarrow V_{\mathcal{O}}$ is a mapping function, where $\mathcal{C}$ is the set of the defined concepts and individual instances in $\mathcal{O}$.
— $\lambda : \mathcal{A} \rightarrow E_{\mathcal{O}} \cup V_{\mathcal{O}}$ is a mapping function, where $\mathcal{A}$ is the set of axioms/assertions in $\mathcal{O}$.
— η is a labeling function that assigns a set of literal names $\eta(i) \subseteq N_L$ to each vertex $i \in V_{\mathcal{O}}$, and a set of literal names $\eta(i,j) \subseteq N_P$ to each edge $(i,j) \in E_{\mathcal{O}}$, where $N_L = N_C \cup N_I$, and N_C, N_I and N_P are the sets of literal names of concepts, individual instances and roles, respectively.

We use integers as indices of vertices and refer to vertices by their indices, which provide us with compactness in reference in operation and storage. Concepts here refer to both atomic concepts, which have basic definitions of their own, and complex concepts, which are defined recursively in terms of other concepts. We use the three mapping functions ρ, λ and η to ensure that each concept defined in $\mathcal{O}$ can be mapped to exactly one labeled vertex or one labeled edge in $G_{\mathcal{O}}$. The sets N_C, N_I and N_P are disjointed with each other.

4 Generating GDRs: An Overview

Graph derivation representations (GDR) are the graphical semantic descriptions of ontologies. The goal of generating GDRs of ontologies is to measure and compare ontologies based on their GDRs for stable semantic measurement. It needs to consider the following two basic problems. The first is how to establish the structural semantics of an original ontology's GDR under the condition of preserving the semantics of the ontology. On one hand, the explicit semantic information in the ontology should be preserved in its structural semantics of GDR. On the other hand, the implicit semantic knowledge in the ontology should also be explicitly expressed by its structural semantics of GDR. Considering that concepts are the core of ontological semantics, we have to make explicit all the defined complex concepts, and excavate the taxonomic and non-taxonomic relations between them. Especially for the complex concepts that are defined by other complex concepts in an iteratively nested fashion, a recursive process needs to be adopted to find out all the defined complex concepts and the implicit relations between them. As such, our approach can establish the complete structural semantics of GDR for an ontology.

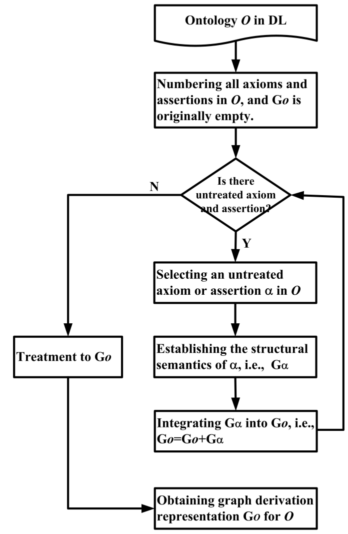

Fig. 1. Overview of Generating GDR for Ontology $\mathcal{O}$

The second problem is how to treat an ontology's GDR to satisfy both the basic measurement criteria and the requirements of automatic measurement. The goal of treating an GDR is to normalize the GDR so that it can be measured correctly and effectively by using the existing ontology measures. For examples, some ontology measures related to fanins/fanouts are concerned about *direct inheritance* between classes instead of non-direct inheritance between them. By eliminating non-direct inheritance, we comply with the definition of fanins/fanouts, and also avoid the double counting of entities. Moreover, cycles of class inheritance will make interminable the computation of path and depth of ontologies, so eliminating such cycles will facilitate automatic ontology measurement supported by ontology management software tools.

The procedure for generating the GDR (denoted as $G_{\mathcal{O}}$) of an ontology $\mathcal{O}$ is outlined in Figure 1. The graph derivation process is conducted in three phases based on the three mapping functions ρ, λ and η. In the first phase, each axiom and assertion in $\mathcal{O}$ is indexed with positive integers, as shown in Examples 1 and 2. The $G_{\mathcal{O}}$ is originally set to empty, and has no vertex and relation. Then, each axiom/assertion α in $\mathcal{O}$ is examined and the GDR (denoted as G_{α}) is generated for each α. Some derivation rules for axioms/assertions can be first applied to generate the structural semantics of α by the derivation based process, which will be discussed in Section 5. If Axiom α includes complex concepts, then the derivation rules for complex concepts in α will be applied in a recursive manner until there is no applicable rule to concepts in α. Once the GDR for each axiom/assertion is generated, we start the second phase, which integrates each GDR into $G_{\mathcal{O}}$ by the integration operation. We obtain an integrated (but untreated) GDR for the given ontology at the end of the second phase. In the third phase, $G_{\mathcal{O}}$ is treated by eliminating cycles of class inheritance and non-direct relations with transitive property. We provide a detailed discussion on the phase 1 in the next section and present the integration and treatment of GDR in Section 6. The process of generating GDRs will be not terminated until there is no derivation rule that is applicable to all the axioms/assertions and concepts in $\mathcal{O}$. The final complete GDR $G_{\mathcal{O}}$ for ontology $\mathcal{O}$ will be obtained.

5 Structural Semantics Based on GDR

5.1 Structural Semantics of Axioms in GDR

In description logic (DL), there are three types of axiom representations of the form $C_1 \sqsubseteq C_2$, $C_1 \equiv C_2$ and $C_1 \sqsubseteq \neg C_2$, where C_1 and C_2 are atomic or complex concepts. Table 1 gives five axioms/assertions (A_1 to A_5), the correspondence between axioms/assertions and its structural semantics of GDR by using the functions ρ, λ and η. Taking the example of using λ to transform $C_1 \sqsubseteq C_2$, the concepts C_1 and C_2 will be mapped onto two vertices by ρ. Moreover, the edge name between them is labeled as `subClassOf`, and further added into the set of the edge by η. There are two types of assertions of the form $C(a)$ and $R(a,b)$, where C is a concept, a and b are instances, and R is a binary relation. For example, if we use λ to transform $R(a,b)$, then $R(a,b)$ is mapped onto an edge, the instances a and b are respectively mapped onto two vertices by ρ, and their names will be stored by η.

We would like to note that by derivation rule **A_2**, multiple concept names share the same vertex in GDR if these concepts are equivalent. This rule amounts to say that we preserve the semantics of class equivalence, i.e., the concepts belonging to the same vertex in GDR are semantically equivalent. This rule also enables us to avoid the problem of cyclic inheritance between equivalent classes, and ensure terminability of automatic ontology measurement process.

5.2 Structural Semantics of Concepts in GDR

If a concept C is atomic, then C will be mapped onto a unique vertex i in GDR such that $\rho(C) = i$ and $C \in \eta(i)$. A similar process applies to an instance of a concept. In contrast, the process of mapping complex concepts to vertices is more complex. Once we detect a complex concept, we need to map it onto a vertex. At the same time, we have to continue to detect whether the complex concept contains other complex concepts defined in a nested fashion. If it does, then all of the iteratively defined concepts should be also mapped so that we can excavate the implicit structural semantics until all the nested complex concepts are mapped. As a result, all the *defined* concepts in ontologies will be measured no matter whether the concepts are atomic or complex.

In general, positive integers of vertices begin with 1. If a vertex is labeled the positive integer i, then the next one is labeled i+1. By using different positive integers, we can differentiate different vertices in a GDR.

For finding out all the complex concepts in an ontology, we need to review the ways in which complex concepts are constructed. In DL, a complex concept can be constructed by the following ways, e.g, $C \sqcap D$, $C \sqcup D$, $\forall R.C$, $\exists R.C$, $\exists R.\{a\}$, $\{a_1, a_2, \cdots, a_n\}$, $\geq nR.C$ and $\leq nR.C$. Table 2 lists the structural semantics of GDRs of different types of complex concepts in DL. Obviously, mapping a complex concept onto a vertex of GDR is a *recursive* process according to Table 2.

It is worth noting that in Table 2 we need to normalize the naming of vertices when complex concepts are mapped onto vertices because complex concepts have no specific names unlike atomic concepts. An atomic concept X can be mapped onto a vertex i with the literal name X. We assign the literal names of complex concepts in terms of the semantic meanings based on which they are defined. For examples, the vertex name of the complex concept $\forall R.C$ is named as forall_R_C. The vertex of the complex concept $C_1 \sqcap C_2$ has the name C_1_and_C_2.

In Table 2, the notion $[info]$ represents that $info$ is the additional knowledge from other knowledge in the ontology being measured. If additional knowledge is available, it can be helpful in finding additional structural semantics of concepts. For example, according to the derivation rule **C_5** in Table 2, if there is only concept $\exists R.C$, then its structural semantics is only $\rho(\exists R.C) = i_5 \in V_{\mathcal{O}} \wedge \rho(C)$. However, if the knowledge $C(b), R(a,b)$ are also available, its additional structural semantics in considering the additional knowledge, $\wedge \lambda(R(a,b)) \wedge \lambda(C(b)) \wedge (\rho(a), i) = e \in E_{\mathcal{O}} \wedge \mathtt{type} \in \eta(e)$, should be appended to the structural semantics of the concept, and thus generates a complete structural semantics. An illustrative example is provided in Section 6.4.

6 Integration and Treatment of GDRs

6.1 Integration of Segmental GDRs

When we establish the structural semantic descriptions for each of concepts and axioms/assertions in an ontology, they should be integrated to generate the integrated GDR of this ontology.

Definition 4. ***(Integrated GDR)*** *Let $\mathcal{O} = \{\alpha_i | 1 \leq i \leq n\}$ be an ontology, where α_i is an axiom or assertion, n is the total number of axioms and assertions in $\mathcal{O}$. For any $\alpha_i \in \mathcal{O}$, its structural semantic description is denoted as $G_{\alpha_i} = (V_{\alpha_i}, E_{\alpha_i}, \rho, \lambda, \eta)$. For any $\alpha_i, \alpha_j \in \mathcal{O}$ ($i \neq j$), the integration operation between them is denoted as $G_{\alpha_i} + G_{\alpha_j} = (V_{\alpha_i} \cup V_{\alpha_j}, E_{\alpha_i} \cup E_{\alpha_j}, \rho, \lambda, \eta)$. The GDR $G_{\mathcal{O}}$ of $\mathcal{O}$ can be further denoted as $G_{\mathcal{O}} = \overset{n}{\underset{i=1}{+}} G_{\alpha_i}$.*

TABLE 1
Structural Semantics of Axioms/Assertions

No.	Axiom/Assertion	Structural Semantics of GDR
A_1	$C_1 \sqsubseteq C_2$	$(\rho(C_1), \rho(C_2)) = e \in E_{\mathcal{O}} \wedge \texttt{subClassOf} \in \eta(e)$
A_2	$C_1 \equiv C_2$	$\rho(C_1) = \rho(C_2) = i \in V_{\mathcal{O}} \wedge \{C_1, C_2\} \subseteq \eta(i)$
A_3	$C_1 \sqsubseteq \neg C_2$	$(\rho(C_1), \rho(C_2)) = e \in E_{\mathcal{O}} \wedge \texttt{disjointWith} \in \eta(e)$
A_4	$C(a)$	$(\rho(a), \rho(C)) = e \in E_{\mathcal{O}} \wedge \texttt{type} \in \eta(e)$
A_5	$R(a_1, a_2)$	$(\rho(a_1), \rho(a_2)) = e \in E_{\mathcal{O}} \wedge R \in \eta(e)$

TABLE 2
Structural Semantics of Concepts/Instances

No.	Concepts	Structural Semantics of GDR	$\eta(i)$
C_1	X is an atomic concept name or an instance name	$\rho(X) = i_1 \in V_{\mathcal{O}} \wedge X \in \eta(i_1)$	$X \in \eta(i_1)$
C_2	$C_1 \sqcap C_2\, [, C_3 \sqsubseteq C_1, C_3 \sqsubseteq C_2]$	$\rho(C_1 \sqcap C_2) = i_2 \in V_{\mathcal{O}} \wedge (i_2, \rho(C_k)) = e_k \in E_{\mathcal{O}} \wedge \texttt{subClassOf} \in \eta(e_k)\, [\wedge (\rho(C_3), i_2) = e \in E_{\mathcal{O}} \wedge \texttt{subClassOf} \in \eta(e)]$, where $k = 1, 2$.	C_1_and_$C_2 \in \eta(i_2)$
C_3	$C_1 \sqcup C_2\, [, C_1 \sqsubseteq C_3, C_2 \sqsubseteq C_3]$	$\rho(C_1 \sqcup C_2) = i_3 \in V_{\mathcal{O}} \wedge (\rho(C_k), i_3) = e_k \in E_{\mathcal{O}} \wedge \texttt{subClassOf} \in \eta(e_k)\, [\wedge (i_3, \rho(C_3)) = e \in E_{\mathcal{O}} \wedge \texttt{subClassOf} \in \eta(e)]$, where $k = 1, 2$.	C_1_or_$C_2 \in \eta(i_3)$
C_4	$\forall R.C$	$\rho(\forall R.C) = i_4 \in V_{\mathcal{O}} \wedge (i_4, \rho(C)) = e \in E_{\mathcal{O}} \wedge R \in \eta(e)$	forall_R_$C \in \eta(i_4)$
C_5	$\exists R.C\, [, C(b), R(a, b)]$	$\rho(\exists R.C) = i_5 \in V_{\mathcal{O}} \wedge \rho(C)\, [\wedge \lambda(R(a, b)) \wedge \lambda(C(b)) \wedge (\rho(a), i_5) = e \in E_{\mathcal{O}} \wedge \texttt{type} \in \eta(e)]$	exist_R_$C \in \eta(i_5)$
C_6	$\exists R.\{b\}\, [, R(a, b)]$	$\rho(\exists R.\{b\}) = i_6 \in V_{\mathcal{O}} \wedge \rho(b)\, [\wedge \lambda(R(a, b)) \wedge (\rho(a), i_6) = e \in E_{\mathcal{O}} \wedge \texttt{type} \in \eta(e)]$	exist_R_$b \in \eta(i_6)$
C_7	$\{a_1, a_2, \cdots, a_n\}$	$\rho(\{a_1, a_2, \cdots, a_n\}) = i_7 \in V_{\mathcal{O}} \wedge (\rho(a_k), i_7) = e_k \in E_{\mathcal{O}} \wedge \texttt{type} \in \eta(e_k)$, for all $1 \leq k \leq n$.	oneof_a_1_$\cdots$_$a_n \in \eta(i_7)$
C_8	$\leq_n R.C\, [, C(b_k), R(a, b_k)]$ for all $1 \leq k \leq n$	$\rho(\leq_n R.C) = i_8 \in V_{\mathcal{O}} \wedge (\rho(a), i_8) = e \in E_{\mathcal{O}} \wedge \rho(C)\, [\wedge \texttt{type} \in \eta(e) \wedge \lambda(R(a, b_k)) \wedge \lambda(C(b_k))]$ for all $1 \leq k \leq n$	leq_n_R_$C \in \eta(i_8)$
C_9	$\geq_n R.C\, [, C(b_k), R(a, b_k)]$ for all $1 \leq k \leq m$, $(m \geq n)$	$\rho(\leq_n R.C) = i_9 \in V_{\mathcal{O}} \wedge (\rho(a), i_9) = e \in E_{\mathcal{O}} \wedge \rho(C)\, [\wedge \texttt{type} \in \eta(e) \wedge \lambda(R(a, b_k)) \wedge \lambda(C(b_k))]$ for all $1 \leq k \leq m$, $(m \geq n)$	geq_n_R_$C \in \eta(i_9)$

6.2 Treatment to Cyclic Inheritance of Vertices

Recall Definition 3, a vertex i in GDR is either a concept vertex or an individual vertex. We use $V_{\mathcal{O}}^C$ and $V_{\mathcal{O}}^I$ to denote the sets of vertices of concepts and individual instances, respectively. Because N_C and N_I are disjointed and class inheritance relations exist only between concept vertices and do not exist between concepts and instances, we have $V_{\mathcal{O}}^C \cap V_{\mathcal{O}}^I = \emptyset$. Therefore, the treatment of cycles of vertex inheritance is applied only to all the concept vertices that form a clique (i.e., vertices connected in a directed inheritance relationship cycle).

We present an algorithm to eliminate cyclic inheritance in an integrated GDR $G_{\mathcal{O}}$, which is shown in Algorithm 1. In the algorithm, the function $Circle(G_{\mathcal{O}})$ is used to find a circle of concept subsumption in $G_{\mathcal{O}}$ by `subClassOf`. It can be implemented by some conventional graph algorithms, such as the depth-first search (DFS) algorithm. The structure S is a set of all vertices contained in a cycle of inheritance. For all the vertices in S, without loss of generality, we first find out the vertex that has the minimum index, which is denoted as v_k. Then, we use v_k to replace all the other vertices in the clique S in four steps. First, the literal names of all vertices in S will be added into the set $\eta(v_k)$. Second, the literal names of edges between vertices in S except `subClassOf` will be added into the set $\eta(v_k, v_k)$. Third, all the binary relations between the vertices outside the clique S and the vertices in S will be replaced by the relations between the vertices not in S and the vertex v_k. Finally, we delete all the vertices in S except v_k. This process continues by examining the newly treated $G_{\mathcal{O}}$ and treating other cyclic inheritance of vertices (cliques). The algorithm will terminate when all the cyclic relations (cliques) in $G_{\mathcal{O}}$ are treated. In the worst case, the time complexity of Algorithm 1 is $\Theta((n + m) * n^2)$, where $n = |V_{\mathcal{O}}|$, $m = |E_{\mathcal{O}}|$. Because $m \leq n * (n - 1) + n \leq n^2$, the time complexity of Algorithm 1 can be regarded as $\Theta(n^4)$ in the worst case.

Algorithm 1 Treatment to Cyclic Inheritance

```
Require: G_O = (V_O, E_O, ρ, λ, η)
S ← Circle(G_O)
while S ! = ∅ do
    v_k ← min{v_i | v_i ∈ S ∧ 1 ≤ i ≤ |S|}
    for all each v ∈ S \ {v_k} do
        η(v_k) ← η(v_k) ∪ η(v)
        if (v_k, v) ∈ E_O then
            η(v_k, v_k) ← η(v_k, v_k) ∪ (η(v_k, v) \ {subClassOf})
            E_O ← E_O ∪ {(v_k, v_k)}
        end if
        if (v, v_k) ∈ E_O then
            η(v_k, v_k) ← η(v_k, v_k) ∪ (η(v, v_k) \ {subClassOf})
            E_O ← E_O ∪ {(v_k, v_k)}
        end if
        for all each v' ∈ V_O \ S do
            if (v, v') ∈ E_O then
                η(v_k, v') ← η(v_k, v') ∪ η(v, v')
                E_O ← E_O ∪ {(v_k, v')}
            end if
            if (v', v) ∈ E_O then
                η(v', v_k) ← η(v', v_k) ∪ η(v', v)
                E_O ← E_O ∪ {(v', v_k)}
            end if
        end for
    end for
    V_O^C ← V_O^C \ {S \ {v_k}}
    S ← Circle(G_O)
end while
```

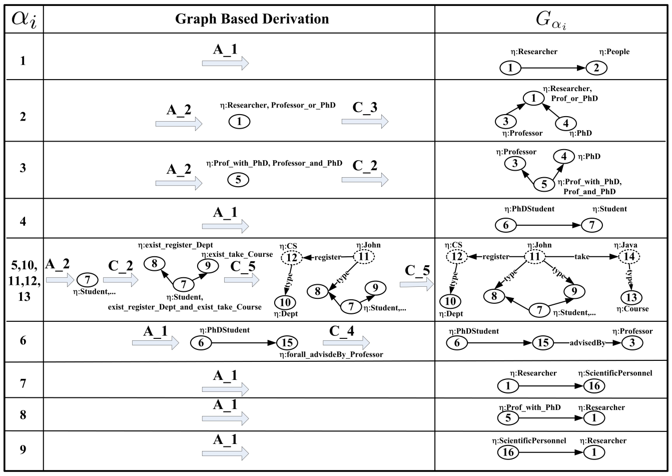

Fig. 2. Generating GDRs for Each Axiom/Assertion in Example 1

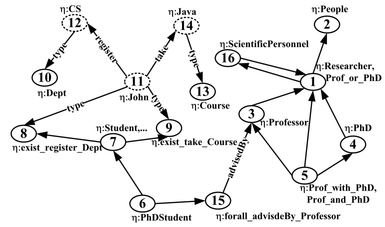

Fig. 3. The Untreated GDR by Integration Operation

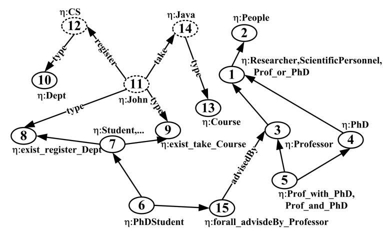

Fig. 4. The GDR of Example 1

6.3 Treatment to Non-direct Transitive Relations

The second type of treatment is to detect all non-direct relations with transitive property between vertices, aiming at eliminating all derived transitive relations between vertices, which are redundant. In principle, if some transitive relations are kept, then the relation derived from these transitive relations should not be kept to prevent the double counting of measurement entities. An obvious example of relations with transitive property is the `subClassOf` relation. An illustrative example is provided in the next section.

Definition 5. ***(Direct Relation between Vertices)*** *Let i and j be two vertices, and R be the role with transitive property. The edge (i, j) with $R \in \eta(i, j)$ is a direct relation iff there is no vertex k such that $R \in \eta(i, k)$ and $R \in \eta(k, j)$.*

We can provide an algorithm to eliminate non-direct relations of R with transitive property in the integrated GDR, which is shown in Algorithm 2. The function $findIntermVertices(i, j)$ in the algorithm is implemented by the definition 5. It will detect whether the edge (i, j) is a direct relation, and return a set of vertices that are the intermediate vertices between vertices i and j, i.e., the set of all intermediate vertices k in Definition 5. Then, the set is stored by MV. If MV is empty, then it means that the edge (i, j) is a direct relation, and vice versa. In the worst case, the time complexity of Algorithm 2 is $\Theta(n^3)$, where $n = |V_{\mathcal{O}}|$.

Algorithm 2 Treatment to Non-direct Transitive Relations w.r.t R

```
Require: G_O = (V_O, E_O, ρ, λ, η)
MV ← ∅
for all i ∈ V_O do
    for all j ∈ V_O^C \ {i} do
        if (i, j) ∈ E_O and η(i, j) ∩ {R} ≠ ∅ then
            MV ← findIntermVertices(i, j)
            if MV ! = ∅ then
                E_O ← E_O \ {(i, j)}
            end if
        end if
    end for
end for
```

6.4 An Example of Generating GDR of Ontology

In the section, we will use Example 1 to illustrate how to generate its GDR. According to the procedure described in Figure 1, we can traverse each of axioms and assertions in the example ontology, and generate its corresponding description of structural semantics using GDR. Then, they are integrated for treatment, and form a complete and semantically stable GDR.

Figure 2 illustrate the derivation process of generating one GDR for each of the axioms/assertions in Example 1. α_i here refers to an axiom/assertion, and i is α_i's index. G_{α_i} is the structural semantic description of α_i. An ellipse is a class vertex, and a dotted ellipse is an instance vertex. An arrow is a `subClassOf` relation, and an arrow associated with a literal name is a binary relation. In the middle column, a gray bold arrow represents the use of a derivation rule, and the literal name over a gray bold arrow is the rule name. Considering the axiom "2.Researcher≡Professor⊔PhD", we first use Rule **A_2** to generate a vertex numbered 1 instead of 2 because the vertex associated with Concept "Researcher" has been already generated when Axiom 1 is transformed, where $\eta(1)$=$\{Researcher, Prof\_or\_PhD\}$. Then Rule **C_3** is further applied, and generates the G_{α_2}. All the structural semantic descriptions of axioms/assertions in Example 1 are merged into an integrated GDR by the integration operation "+", as shown in Figure 3.

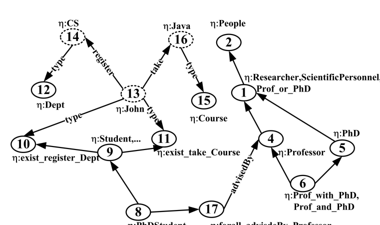

Fig. 5. The GDR of Example 2 By Using Treatments

Now we enter the phase 3 in which we treat cyclic vertices and non-direct transitive relations. According to Algorithm 1, we detected the cycle between the two vertices 1 and 16. To treat the cycle, the vertex 1 will be reserved such that $\eta(1) = \eta(1) \cup \eta(16)$={$Researcher$, $ScientificPersonnel$, $Professor$_or_PhD} because $\min\{1, 16\} = 1$. Then, the vertex 16 will be deleted, and its semantics is preserved by literal names of the vertex 1. It is straightforward to prove that we can use anyone of the vertices in a clique S to break the clique and without loss of generality, we choose the vertex with the smallest vertex index for convenience. After removing all inheritance cliques (circles), we apply the second type of treatment operation. According to Algorithm 2, we detect that there exist two vertices 3 and 4 between vertices 1 and 5 such that the edge (1,5) is a non-direct inheritance (`subClassOf` is transitive). The edge (1,5) will be deleted, but its semantics can be still preserved because it can be derived from other relations of direct inheritance. At last, the GDR of Example 1 after using treatment operations is shown in Figure 4. Similarly, we can generate the GDR for Example 2, which is shown in Figure 5.

Given a GDR, the ontology knowledge that it represents is uniquely determined. The uniqueness of ontology knowledge description is determinate in terms of labels, connecting structure and graph isomorphism, e.g., Figures 4 and 5. From the perspective, the GDR and the OWL-DL are mutually equivalent.

7 PROPERTY ANALYSIS OF GDR

From the rule **A_2**, the structural semantics of class equivalence is represented in a GDR as follows: the names of equivalent classes belong to the same vertex.

Proposition 1. *Let $G_{\mathcal{O}}$=$(V_{\mathcal{O}}, E_{\mathcal{O}}, \rho, \lambda, \eta)$ be the GDR of Ontology $\mathcal{O}$. For any vertex $i \in V_{\mathcal{O}}^{C}$, if there exist two concept names $L_1, L_2 \in \eta(i)$, then the concepts L_1 and L_2 in $\mathcal{O}$ are equivalent, i.e., $L_1 \equiv L_2$.*

From the derivation process of generating GDR, it is not difficult to find that each vertex in a GDR is assigned at least one literal name. So we we obtain the Lemma 1.

Lemma 1. *Let $G_{\mathcal{O}}$=$(V_{\mathcal{O}}, E_{\mathcal{O}}, \rho, \lambda, \eta)$ be the GDR of Ontology $\mathcal{O}$. For any vertex $i \in V_{\mathcal{O}}$, $\eta(i) \neq \emptyset$.*

According to Definition 3, each concept or an axiom/assertion in an ontology is assigned to a unique positive number by the function ρ or the function λ. Meanwhile, it is impossible that two concepts in the same ontology have the same concept names. So the intersection of sets of literal names of two different vertices in a GDR must be empty. It is formally represented in Lemma 2.

Lemma 2. *Let $G_{\mathcal{O}}$=$(V_{\mathcal{O}}, E_{\mathcal{O}}, \rho, \lambda, \eta)$ be the GDR of Ontology $\mathcal{O}$. For any vertices $i, j \in V_{\mathcal{O}}$, $i \neq j$ if and only if (iff) $\eta(i) \cap \eta(j) = \emptyset$.*

We can alternatively formalize Lemma 2 as follows.

Lemma 3. *Let $G_{\mathcal{O}}$=$(V_{\mathcal{O}}, E_{\mathcal{O}}, \lambda, \eta)$ be the GDR of Ontology $\mathcal{O}$. For any two vertices $i, j \in V_{\mathcal{O}}$, $\eta(i) \cap \eta(j) \neq \emptyset$ iff $i = j$.*

The graph derivation approach fully captures the semantics of all the elements in an ontology, i.e., concepts, instances, properties/relations and axioms. We generate their structural semantic description strictly in accordance with their underlying semantics. Therefore, the GDR of an ontology not only preserves the explicit expressed semantics in the ontology, but also considers the implicit semantics derived from the explicit knowledge. During treatment to the integrated GDR of Ontology $\mathcal{O}$, Proposition 1 can guarantee the semantics of class equivalence is preserved although the vertices of equivalent classes and the relevant relations are deleted from $G_{\mathcal{O}}$. The non-direct relations with transitive property are deleted from $G_{\mathcal{O}}$, but we still preserve their semantics because they can be obtained by the other direct relations.

Lemma 4. *$G_{\mathcal{O}}$ and its derivation based process preserve the semantics of Ontology $\mathcal{O}$.*

The graph derivation based process of generating GDR is performed in a recursive manner. We formalize the termination property of GDR derivation process as follows.

Definition 6. *(**Terminability**) Given an ontology $\mathcal{O}$ with finite set of concepts and finite set of axioms/assertations, the GDR derivation process of generating $G_{\mathcal{O}}$ is terminable iff the following conditions hold:*
—For any axiom/assertion α in $\mathcal{O}$, α is included in its GDR G_α through its mapping functions λ and η.
—For any concept C in $\mathcal{O}$, C is included in $V_{\mathcal{O}}$ of its GDR through its mapping functions ρ and η.
—There is no other rules applicable to any axiom/assertion and concept in $\mathcal{O}$ during the graph derivation based process.

The axioms and assertions contained in an ontology $\mathcal{O}$ are finite, the proposed rules are also finite and the construction of GDR removes all cliques (cyclic inheritance). Thus, the recursive derivation process should be terminated in a finite time.

Let $G_{\mathcal{O}_1} = (V_{\mathcal{O}_1}, E_{\mathcal{O}_1}, \rho_1, \lambda_1, \eta_1)$ and $G_{\mathcal{O}_2} = (V_{\mathcal{O}_2}, E_{\mathcal{O}_2}, \rho_2, \lambda_2, \eta_2)$ be the GDRs of two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. We have the following two theorems.

Theorem 1. *For a function $f : V_{\mathcal{O}_1} \rightarrow V_{\mathcal{O}_2}$, if $\forall i \in V_{\mathcal{O}_1}$, there exists $j \in V_{\mathcal{O}_2}$ such that $\eta_1(i) \subseteq \eta_2(j)$, then f is injective.*

Proof: For any vertex $i \in V_{\mathcal{O}_1}$ such that $\exists j_1, j_2 \in V_{\mathcal{O}_2}$, $\eta_1(i) \subseteq \eta_2(j_1)$ and $\eta_1(i) \subseteq \eta_2(j_2)$. Thus, we have $\eta_2(j_1) \cap \eta_2(j_2) \neq \emptyset$. By Lemma 3, we have $j_1 = j_2$. Now we can conclude that f is injective. □

Theorem 2. *For a function $f : V_{\mathcal{O}_1} \rightarrow V_{\mathcal{O}_2}$, if $\forall i \in V_{\mathcal{O}_1}$, there exists $j \in V_{\mathcal{O}_2}$ such that $\eta_1(i) = \eta_2(j)$, then f is bijective.*

Proof: For any vertices $i \in V_{\mathcal{O}_1}$ and $j \in V_{\mathcal{O}_2}$, we know the following fact: $\eta_1(i) = \eta_2(j) \Leftrightarrow \eta_1(i) \subseteq \eta_2(j) \wedge \eta_2(j) \subseteq \eta_1(i)$. Thus, by Theorem 1, we can obtain Theorem 2. □

The GDRs of ontologies will be used for ontology comparison. What is first made for the similarity comparison of GDRs is to determine the identical vertices and edges in the two GDRs being compared. We can determine whether two vertices from two different GDRs are identical by the sets of literal names they have obtained in the GDR transformation.

Definition 7. ***(Identical Vertices and Edges)*** *For any vertices $i \in V_{\mathcal{O}_1}$ and $j \in V_{\mathcal{O}_2}$, if $\eta_1(i) = \eta_2(j)$, then vertices i and j are identical, denoted as $Ident(i) = j$. For any edges $(i, j) = e \in E_{\mathcal{O}_1}$,$(i', j') = e' \in E_{\mathcal{O}_2}$, and $Ident(i) = i'$ and $Ident(j) = j'$ such that $\eta_1(e) = \eta_2(e')$, then edges (i, j) and i', j' are identical, denoted as $Ident(i, j) = (i', j')$.*

Vertex 15 in Figure 4 and Vertex 17 in Figure 5 are identical (i.e., $Ident(15) = 17$) because $\eta_1(15) = \eta_2(17) = \{\text{forall\_}advisedBy\_Professor\}$. Similarly, Edge $(6, 7)$ in Figure 4 and Edge $(8, 9)$ in Figure 5 are identical.

Definition 8. ***(Graph Isomorphism of GDRs)*** *A bijective function $Ident : V_{\mathcal{O}_1} \rightarrow V_{\mathcal{O}_2}$ is a graph isomorphism from $G_{\mathcal{O}_1}$ to $G_{\mathcal{O}_2}$ if the following conditions hold:*
1) $\eta_1(i) = \eta_2(Ident(i))$ for all $i \in V_{\mathcal{O}_1}$,
2) For any edge $(i, j) = e \in E_{\mathcal{O}_1}$ there exists an edge $(Ident(i), Ident(j)) = e' \in E_{\mathcal{O}_2}$ such that $\eta_1(e)$=$\eta_2(e')$, and for any edge $(i', j') = e' \in E_{\mathcal{O}_2}$ there exists an edge $(Ident^-(i'), Ident^-(j')) = e \in E_{\mathcal{O}_1}$ such that $\eta_2(e')$=$\eta_1(e)$.

Theorem 3. *The GDR $G_{\mathcal{O}}$ of an ontology $\mathcal{O}$ is unique in terms of labels, connecting structure and isomorphism.*

Proof: Assume that Ontology $\mathcal{O}$ has two GDRs $G_{\mathcal{O}}$ and $G^-_{\mathcal{O}}$, where $G_{\mathcal{O}}$=$(V_{\mathcal{O}}, E_{\mathcal{O}}, \rho, \lambda, \eta)$ and $G^-_{\mathcal{O}}$= $(V^-_{\mathcal{O}}, E^-_{\mathcal{O}}, \rho, \lambda, \eta)$.

All the concepts defined in $\mathcal{O}$ must be respectively mapped onto vertices of $G_{\mathcal{O}}$ and $G^-_{\mathcal{O}}$ because both $G_{\mathcal{O}}$ and $G^-_{\mathcal{O}}$ preserve the same semantics of $\mathcal{O}$ according to Lemma 4. So there must exist a bijective function $Ident : V_{\mathcal{O}} \rightarrow V^-_{\mathcal{O}}$ such that $\forall i \in V_{\mathcal{O}}$, $\exists j \in V^-_{\mathcal{O}}$ such that $\eta(i) = \eta(j)$. According to Definition 7, $Ident(i) = j$. That is, $\eta_1(i) = \eta_2(Ident(i))$ for all $i \in V_{\mathcal{O}}$.

Then, we detect whether the edges in $E_{\mathcal{O}}$ and $E^-_{\mathcal{O}}$ are identical. Let $\alpha \in \mathcal{O}$ be any axiom/aasertion, and the concepts/individuals at both ends of α be A and B. When α is transformed to $G_\alpha = (V_\alpha, E_\alpha, \rho, \lambda, \eta)$ and $G^-_\alpha = (V^-_\alpha, E^-_\alpha, \rho, \lambda, \eta)$, the derivation rules in Table 1 are always first applied by λ. Moreover, the derivation rule applicable to α can be determined because α can only satisfy exact one of the five forms of axioms/assertions. After that, A and B will be transformed by the derivation rules in Table 2. For each of A and B, its applicable derivation rule is also determinable. However, because of the different orders of applicable rules for A and B, Function ρ will respectively transform them to two different positive integers in V_α and V^-_α. Assume $i_A, i_B \in V_\alpha$, $j_A, j_B \in V^-_\alpha$, we have $\eta(i_A) = \eta(j_A)$ and $\eta(i_B) = \eta(j_B)$. But, the literal name of Edge $(\rho(A), \rho(B))$ w.r.t α is the same in E_α and E^-_α, i.e., $\eta((i_A, i_B)) = \eta((j_A, j_B))$. According to Definition 7, $Ident((i_A, i_B)) = (j_A, j_B)$. For any two concepts nestedly defined in α, similar is to A and B, ie., the relations (if any) between them are identical in E_α and E^-_α. According to Definition 8, G_α and G^-_α are isomorphic. Obviously, $G_{\mathcal{O}}$ and $G^-_{\mathcal{O}}$ after using integration operation are also isomorphic.

So GDRs $G_{\mathcal{O}}$ and $G^-_{\mathcal{O}}$ are the same in terms of labels, connecting structure and isomorphism. Based on this consideration, the GDR of $\mathcal{O}$ is unique. □

Theorem 3 will be helpful to eliminate the polymorphism of ontology representation, and make the measurement results between GDRs of different ontologies comparable in terms of labels, connecting structure and isomorphism of graphs. According to the definition 2, we can further obtain the conclusion.

Theorem 4. *Ontology measurement based on GDR is stable.*

8 Ontology Measurement and Comparison Based on GDRs

8.1 Classification of Measurement Entities

Although various ontology measures have been proposed in the last decade, the types of measurement entities can be generally classified into two classes of entity types in terms of granularity: *Fine-grained measurement entity types* and *coarse-grained measurement entity types*. We argue that most of the existing ontology measures should be utilized to measure the semantic structures of ontologies in the form of their GDRs. Measurement entities are fine-grained if they are the basic elements of ontologies such as concepts/classes, properties, binary relations, axioms and instances. In contrast, coarse-grained measurement entities are the intrinsic constructs in ontologies, and consist of different kinds of basic ontological elements, such as fanin, fanout, path, ontology partition, and so on. For examples, fanouts are one of the intrinsic constructs of ontologies. A fanout necessarily deals with two vertex classes and the inheritance relation between them. The fanouts of a class refer to the direct subclasses of the class. A path is an end-to-end chain, consisting of classes and direct inheritance relation from the root class to a leaf class.

Table 3 gives the types of measurement entities that the existing ontology measures deal with, and summarize their formal definitions based on GDRs. Nine types of measurement entities are listed, and are classified into the two categories of granularity. For each of these measurement entities, we give their formal definitions of collecting these entities based on GDRs. Therefore, we can automatically measure ontologies by implementing the algorithms of the existing measures based on GDRs.

8.2 Measurement Evaluation Based on GDR

8.2.1 Exemplar Ontologies

We collected 12 exemplar ontologies with different expressivity for measuring their entities. They were ob-

TABLE 3
Classification of Measurement Entities Based on GDRs

Granularity	Entity Type	Set of Measurement Entities	Formal Definition
Fine-grained	Class	Set of classes: SC	$\{C \mid \forall v \in V_{\mathcal{O}}^{C}(C \in \eta(v))\}$
	Property	Set of properties: SP	$\{P \mid \forall e \in V_{\mathcal{O}}^{I} \times V_{\mathcal{O}}^{I}(P \in \eta(e))\}$
	Instance	Set of instances: SI	$V_{\mathcal{O}}^{I}$
	Leaf class	Set of leaf classes: SLC	$\{v \mid v \in V_{\mathcal{O}}^{C} \wedge (\nexists v' \in V_{\mathcal{O}}^{C}(\texttt{subClassOf} \in \eta(v', v)))\}$
	Class inheritance	Set of class inheritance: SCI	$\{e \mid e \in V_{\mathcal{O}}^{C} \times V_{\mathcal{O}}^{C} \wedge \texttt{subClassOf} \in \eta(e)\}$
	Axiom	Set of axioms: SA	$\{e \mid e \in V_{\mathcal{O}}^{C} \times V_{\mathcal{O}}^{C} \wedge (\texttt{subClassOf} \in \eta(e) \vee \texttt{disjointWith} \in \eta(e))\}$
Coarse-grained	Fanout	Set of fanouts w.r.t class C: $SFO(C)$	$\{v' \mid C \in \eta(v) \wedge (\exists v'(\texttt{subClassOf} \in \eta(v', v)))\}$
	Fanin	Set of fanins w.r.t class C: $SFI(C)$	$\{v' \mid C \in \eta(v) \wedge (\exists v'(\texttt{subClassOf} \in \eta(v, v')))\}$
	Path	Path w.r.t the leaf class C: $Path(C)$	$(v_1, v_2, \cdots, v_n)$, where v_1 is the root vertex, $C \in \eta(v_n) \wedge v_n \in SLC$, and $subClassOf \in \eta(v_i, v_{i+1})$ for all $1 \leq i \leq n-1$

tained by searching some ontology libraries and visiting the websites of the related institutes such as NCI. Some of the ontologies are common in real life. E.g., *Person*, *Wine*, *miniTambis*, and *GlycO*, and so on. The others can be regarded as industrial ontologies such as *Biochemistry*, *Mouse* and *Pharmacogenomics*, which contain a large number of classes and relations. We will provide the selection of measures in the next section and the experimental analysis of exemplar ontologies in Section 8.2.3.

8.2.2 Selection of Measures

Ontology measures are selected according to their types of their measurement entities. We select the following measures, which cover almost all types of measurement entities mentioned in Section 8.1.

NOC (number of classes): $NOC(\mathcal{O}) = |SC|$.

NOP (number of properties): $NOP(\mathcal{O}) = |SP|$.

NOA (number of axioms): $NOA(\mathcal{O}) = |SA|$.

AFC (average fanouts per class, also called inheritance richness in [26]): $AFC(\mathcal{O}) = \frac{\sum_{v_i \in SC} |SFO(v_i)|}{|SC|}$.

$ADILC$ (average depth of inheritance of leaf classes), $ADILC(\mathcal{O}) = \frac{\sum_{p \in PS} |p|}{|PS|}$, where $PS = \{Path(C) \mid C \in SLC\}$ is the set of all paths of $\mathcal{O}$, and $|p|$ is the depth of the path p (i.e., the total number of vertices in p).

For examples, Measures NOC, NOP and NOA are related to some fine-grained measurement entities such as classes, properties and axioms, respectively. Measure AFC measures fan-outs and classes. The measurement entities related to Measure ADILC often include paths, class inheritance and leaf classes. Instances are not considered here because of the fact that most of ontologies in real world seldom contain instances.

8.2.3 Measurement Experiments and Analysis

In this section we present some experimental analysis on the effectiveness of stable measurement based on GDRs. We use a dozen of real world known ontologies to evaluate how well the GRDs represent ontologies and how well our GRD approach will measure and compare structural semantics of ontologies in comparison with the traditional graphical models (denoted as GMs). GMs are often based on the intuitive ontology graphs that represent the *explicit* information of ontologies, where nodes are the *explicitly defined* concepts/instances and edges are the *explicitly defined* relations between nodes. For example, GMs consider the concepts explicitly defined by `owl:Class` instead of complex concepts, and the relations explicitly defined by some labels such as `owl:subClassOf`, `owl:onProperty` and `owl:type` instead of implicitly derived relations. In contrast, GDRs can resolve the polymorphism of ontology representation by representing both explicit and implicit knowledge of an ontology uniquely. We empirically evaluate the difference between GMs and GDRs by analyzing the number of measurement entities each of them may have. The measures selected are used to measure these exemplar ontologies based on their GMs and GDRs respectively. The time complexity of measurements is also computed. Table 4 shows the specific measurement results of entities for the dozen exemplar ontologies.

The first observation is that GDRs have significantly higher number of classes (measures NOC) and higher number of axioms (measure NOA) between GMs and GDRs for all twelve ontologies. This is because GDRs can successfully excavate and represent more classes that are implicitly defined in the given dozen of ontologies. Axioms related to these classes can also be excavated to enrich semantic structures of ontologies, for example, they can also make some axioms/assertions explicit. Similarly, for the same set of ontologies, the AFC measure value of its GDR is larger than that of its GM, which indicates that GDRs can find more semantic structure and links between classes than GMs. Another interesting observation is the exception that the number of properties (measure NOP) in GMs and GDRs of the tested ontologies is the same because all the properties/roles are explicitly defined in the ontologies. By observation, we can empirically validate that GDRs are useful and effective in representing both explicit and implicit information of ontologies.

As shown in the last two columns of Table 4, ontology measurement based on GDRs requires more processing time. Especially for a large volume of expressive ontologies, it has higher time complexity and runs for a longer

TABLE 4
Ontology Measurement Comparison Based on GMs and GDRs

	Measure NOC		Measure NOP		Measure NOA		Measure AFC		Measure $ADILC$		Time(s)	
	GM	GDR	GM	GDR	GM	GDR	GM	GDR	GM	GDR	GM	GDR
Wine	76	209	24	24	15	272	0.197	1.838	1.284	3.289	2.196	2.869
GlycO	370	496	243	243	506	842	1.368	1.700	9.148	6.522	2.131	7.656
MiniTambis	182	315	44	44	72	330	0.396	1.222	1.609	2.532	1.094	1.536
Univbench	43	57	48	48	34	54	0.791	1.059	2.531	3.708	0.543	0.903
OTN	179	249	123	123	204	298	1.140	1.197	3.813	3.711	1.037	1.281
Terrorism	21	30	33	33	16	29	0.762	1.036	2.500	3.364	0.529	1.289
Publication	13	24	10	10	11	22	0.846	0.917	2.818	3.687	0.521	0.898
Person	21	26	6	6	20	25	0.952	0.962	2.706	3.706	0.447	0.842
SWRC	55	99	160	160	47	115	0.855	1.161	2.546	3.099	0.486	1.003
Biochemistry	64	136	0	0	12	98	0.410	0.717	7.242	8.172	0.513	1.010
Mouse	2744	8012	3	3	4493	5647	1.642	1.890	3.140	4.170	30.183	66.081
Pharmacogenomics	145	800	3	3	29	893	0.734	1.114	5.916	6.564	8.521	15.130

duration. For example, the time complexity of measuring Ontology *Mouse* is 66.081 seconds compared to 30.183 seconds for GM, though there are a few ontologies that requires slightly higher time by GDR compared to GM, such as Wine (2.196 in GM and 2.869 in GDR), MiniTambis (1.094 in GM and 1.536 in GDR), and OTN (1.037 in GM and 1.281 in GDR). We argue that higher time complexity can be tolerable in order to provide a stable measurement of the complete structural semantics of ontologies and achieve reliable ontology measurement.

8.3 Comparison Evaluation Based on GDR

In this section, we evaluate the usefulness of GDRs with respect to their capabilities of comparing ontologies. GDR can resolve polymorphism of ontology representation because it is a normalized and "unique" representation of an ontology. Therefore, ontology measurement results based on GDRs are comparable and reliable. Ontology comparison needs to consider two aspects. One is to detect whether an ontology is sub-ontology of another ontology (inclusion relationship). The other is to compute the semantic similarity between two ontologies.

8.3.1 Subontology Comparison

Definition 9. ***(Subontology)*** *Ontology $\mathcal{O}_1$ is a subontology of Ontology $\mathcal{O}_2$ iff $G_{\mathcal{O}_1}$ is a subgraph of $G_{\mathcal{O}_2}$. $G_{\mathcal{O}_1}$ is a subgraph of $G_{\mathcal{O}_2}$, denoted as $G_{\mathcal{O}_1} \subseteq G_{\mathcal{O}_2}$, iff there exists an injective function $sub : V_{\mathcal{O}_1} \rightarrow V_{\mathcal{O}_2}$ such that:*
—For any vertex $i \in V_{\mathcal{O}_1}$, $\eta_1(i) \subseteq \eta_2(sub(i))$.
—For any edge $(i, j) \in E_{\mathcal{O}_1}$, $\eta_1(i, j) \subseteq \eta_2(sub(i), sub(j))$.

Based on Definition 9, we develop Algorithm 3, which detects whether an GDR $G_{\mathcal{O}_1}$ is a subontology of another GDR $G_{\mathcal{O}_2}$. As shown in Algorithm 3, for any edge $(i, j) = e$, functions $front(e)$ and $end(e)$ return the vertices i and j respectively. Thus, Algorithm 3 determines sub-ontology relationship between two ontologies by detecting inclusion between the sets of labels of vertices and edges from $G_{\mathcal{O}_1}$ and $G_{\mathcal{O}_2}$. In the worst case, its complexity is $\Theta(n_1^2 * n_2^2)$, where $n_1 = |V_{\mathcal{O}_1}|$, $n_2 = |V_{\mathcal{O}_2}|$.

Consider the following Example 3 and its GDR shown in Figure 6, we want to determine whether the GDR in Figure 6 is a subgraph of the GDR in Figure 4 by using Algorithm 3. By testing the inclusion relationship between the sets of labels of vertices and edges from the two GDRs, we can easily conclude that Example 3 is a subontology of Example 1.

Algorithm 3 Detecting whether $G_{\mathcal{O}_1} \subseteq G_{\mathcal{O}_2}$

```
Require: G_O1=(V_O1, E_O1, ρ1, λ1, η1), G_O2=(V_O2, E_O2, ρ2, λ2, η2)
if |V_O1| > |V_O2| then
    return false
end if
for all i ∈ V_O1 do
    i ← 0
    for all j ∈ V_O2 do
        if η1(i) ⊆ η2(j) then
            break
        else
            i ← i + 1
        end if
    end for
    if i == |V_O2| then
        return false
    end if
end for
for all e ∈ E_O1 do
    i ← 0
    for all e' ∈ E_O2 do
        if η1(front(e))==η2(front(e')) and η1(end(e))==η2(end(e')) then
            if η1(e) ⊆ η2(e') then
                break
            else
                i ← i + 1
            end if
        end if
    end for
    if i == |E_O2| then
        return false
    end if
end for
return true
```

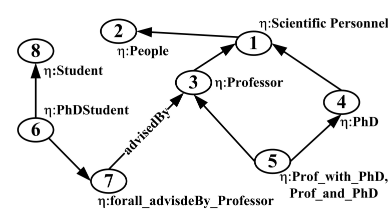

Fig. 6. The GDR of Example 3

Example 3. *TBox={1.ScientificPersonnel $\sqsubseteq$ People, 2.Professor $\sqsubseteq$ ScientificPersonnel, 3.PhD $\sqsubseteq$ ScientificPersonnel, 4.Prof_with_phD $\equiv$ PhD $\sqcap$ Professor, 5.PhDStudent $\sqsubseteq\forall$advisedBy.Professor, 6. PhDStudent $\sqsubseteq$Student}.*

ABox=$\emptyset$.

8.3.2 Ontology Comparison Based on Distance Metric

Some well-known approaches for measuring similarity between graphs often deal with the problem of graph isomorphism, subgraph isomorphism and maximal common subgraphs [47]–[49]. In this paper, we will use the distance based metric in [47] by utilizing the bijective function $Ident$ in Definition 8 for comparing two ontologies based on their GDRs. The distance metric $d(G_{\mathcal{O}_1}, G_{\mathcal{O}_2})$ between $G_{\mathcal{O}_1}$ and $G_{\mathcal{O}_2}$ in [47] can be formally defined by Equation 1 as follows.

$$d(G_{\mathcal{O}_1}, G_{\mathcal{O}_2}) = 1 - \frac{|mcs(G_{\mathcal{O}_1}, G_{\mathcal{O}_2})|}{\max(|G_{\mathcal{O}_1}|, |G_{\mathcal{O}_2}|)} \tag{1}$$

Where $mcs(G_{\mathcal{O}_1}, G_{\mathcal{O}_2})$ represents the maximal common subgraph of GDRs $G_{\mathcal{O}_1}$ and $G_{\mathcal{O}_2}$. Here, $|G_{\mathcal{O}}|$ represents the number of vertices of $G_{\mathcal{O}}$ (i.e., $|V_{\mathcal{O}}|$).

The algorithmic complexity for comparing the similarity between $G_{\mathcal{O}_1}$ and $G_{\mathcal{O}_2}$ mainly relies on that of obtaining the vertices in their maximal common subgraph (MCS), which is shown in Algorithm 4. In the algorithm, V is the set of vertices of the MCS of $G_{\mathcal{O}_1}$ and $G_{\mathcal{O}_2}$. In the worst case, the complexity of Algorithm 4 is $\Theta(n^4)$, where $n = \min\{|G_{\mathcal{O}_1}|, |G_{\mathcal{O}_2}|\}$.

Algorithm 4 Obtaining the set of vertices of MCS

```
Require: G_O1=(V_O1, E_O1, ρ1, λ1, η1), G_O2=(V_O2, E_O2, ρ2, λ2, η2)
V ← ∅
for all e ∈ E_O1 do
    for all e' ∈ E_O2 do
        if η1(front(e))==η2(front(e')) and η1(end(e))==η2(end(e')) then
            if η1(e) == η2(e') then
                V ← V ∪ {front(e), end(e)}
            end if
        end if
    end for
end for
return V
```

We then use the distance based metric to pairwise compare their similarity among the fifteen exemplar ontologies by their GDRs. They include the twelve exemplar ontologies in Section 8.2 and the three ontologies (i.e., Examples 1, 2 and 3) presented in this paper. The comparison results are shown in Table 5.

Because $d(G_{\mathcal{O}_1}, G_{\mathcal{O}_2}) = d(G_{\mathcal{O}_2}, G_{\mathcal{O}_1})$, we only list the similarity values between GDRs in the lower left in Table 5. Moreover, by Equation 1, the distance metric $d(G_{\mathcal{O}_1}, G_{\mathcal{O}_2})$ is to measure the *dissimilarity* between $G_{\mathcal{O}_1}$ and $G_{\mathcal{O}_2}$. That is, the larger the distance value between $G_{\mathcal{O}_1}$ and $G_{\mathcal{O}_2}$ is, the more dissimilar they are.

The experimental evaluations for ontology comparison are made based on the following considerations.

1) If the distance similarity between two ontologies is 0.000, then they represent the same semantic knowledge in the same domain. For example, the GDRs of Examples 1 and 2 have the same distance similarity 0.000, which indicates that they represent the same semantic knowledge in the same domain. Considering the fact that the two ontologies are presented in different structures, we argue that the GDR based representation can handle the problem of polymorphism of ontology representation.

2) If an ontology has been detected as a subontology of another ontology, they represent the same domain, but the knowledge scopes they cover in the domain are possibly different. The degree to which the subontology covers the knowledge scope compared with the ontology, is just the distance similarity between them subtracted by 1. The larger the similarity between them is, the less knowledge scope the subontology will cover. For example, as is mentioned in Section 8.3.1, Example 3 is a subontology of Example 1. Their distance similarity is 0.467 in Table 5, which means that Example 3 only covers 53.3 percent of the semantic knowledge in Example 1.

It is worth noting that the distance similarity between GDRs of Examples 2 and 3 is also 0.467, which also shows that our graph derivation based approach can resolve the problem of polymorphism of ontology representation for stable and meaningful comparison between different ontologies.

3) If the distance similarity between two ontologies is 1.000, then they represent the different semantic knowledge in different domains. For example, Ontology *Wine* has different domains compared with Examples 1, 2 and 3. In fact, we found that most of the exemplar ontologies are to represent different domains by the experiments.

4) If the distance similarity between two ontologies is larger than 0.000 but less than 1.000, then the partial semantic knowledge that they carry are overlapped. The degree of overlap is the similarity value between the two ontologies subtracted by 1, which represents the dissimilarity between the two ontologies, denoted by $d(G_{\mathcal{O}_1}, G_{\mathcal{O}_2})$ in Formula (1). The larger the value of $d(G_{\mathcal{O}_1}, G_{\mathcal{O}_2})$ is, the less amount of the common semantic knowledge the ontologies $G_{\mathcal{O}_1}$ and $G_{\mathcal{O}_2}$ will overlap. For example, in Table 5, the distance similarities between *Person* and *Univbench* and between *Person* and *Publication* are 0.882 and 0.955 respectively. Thus *Person* has the larger degree of semantic overlap with *Univbench* than the overlap it has with *Publication*.

In this section, our ontology similarity comparison based on the *overall* semantic information of ontologies. For the same ontologies, the semantic knowledge that the GMs method compares is only a part of the semantic knowledge that the GDRs method compares. We argue that ontology similarity comparison based on complete semantic information (i.e., GDRs) must be more accurate than ontology similarity comparison based on partial semantic information (i.e., GMs) for the same ontologies. It can be predicted that GDR-based comparison will have higher time complexity than GM-based comparison, which is tolerable for more accurate ontology similarity comparison w.r.t the complexity of Algorithm 4.

9 Conclusion

We have presented a GDR derivation based approach to stably measure and compare ontologies. By theoretical analysis of the properties of GDR, we show that the GDR of an ontology is semantic-preserving and "unique" in terms of labels, connecting structure and isomorphism, which guarantees stable semantic ontology measurement. We analyze and evaluate the usefulness of our

TABLE 5
Ontology Similarity Comparison Based on GDRs

	Ex. 1	Ex. 2	Ex. 3	Wine	GlycO	MiniT.	Univb.	OTN	Terr.	Pub.	Person	SWRC	Bioche.	Mouse	Pharm.
Ex. 1	——	——	——	——	——	——	——	——	——	——	——	——	——	——	——
Ex. 2	0.000	——	——	——	——	——	——	——	——	——	——	——	——	——	——
Ex. 3	0.467	0.467	——	——	——	——	——	——	——	——	——	——	——	——	——
Wine	1.000	1.000	1.000	——	——	——	——	——	——	——	——	——	——	——	——
GlycO	1.000	1.000	1.000	1.000	——	——	——	——	——	——	——	——	——	——	——
MiniT.	1.000	1.000	1.000	1.000	0.993	——	——	——	——	——	——	——	——	——	——
Univb.	0.947	0.947	0.965	1.000	1.000	1.000	——	——	——	——	——	——	——	——	——
OTN	1.000	1.000	1.000	0.996	1.000	1.000	1.000	——	——	——	——	——	——	——	——
Terr.	1.000	1.000	1.000	1.000	1.000	1.000	0.980	1.000	——	——	——	——	——	——	——
Pub.	1.000	1.000	1.000	1.000	1.000	1.000	0.941	1.000	1.000	——	——	——	——	——	——
Person	0.885	0.885	0.885	1.000	1.000	1.000	0.882	1.000	1.000	0.955	——	——	——	——	——
SWRC	0.980	0.980	0.980	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	——	——	——	——
Bioche.	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	——	——	——
Mouse	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	——	——
Pharm.	1.000	1.000	1.000	1.000	1.000	1.000	0.993	1.000	1.000	0.993	1.000	1.000	1.000	1.000	——

GDR approach and compare our GDR with conventional graph models (GM). First, the GDR approach offers stable and reliable semantic measure of ontologies and provides a feasible solution for automated ontology comparison and measurement. Second, the measurement and comparison based on GDRs are more useful and meaningful for the ontologies with a large number of complex concepts. Thus, our GDR based approach can also be used as a complementary mechanism by the existing ontology measurement approaches.

Our work on stable and reliable semantic measurements of ontologies continues along three directions. First, the GDR-based ontology comparison assumes that concepts have the same semantics if their names are the same, which is not always true in real life ontology comparison. We will explore compare the similarity between ontologies with polysemy and toponomy. Second, by the GDR-based ontology comparison, the candidate ontologies that are most similar to a given ontology can be selected from the ontology repositories. The selected ontologies can be further used to help users enrich the ontology design and improve its quality for specific application domain of interest. Three, we can further explore new methods for ontology reuse by utilizing the semantically clean and enriched structures in GDRs.

Acknowledgements

We are grateful to all the referees for their helpful comments. This work is partially supported by the National Natural Science Foundation of China (61001197), the State Key Laboratory Program of Computer Science, Institute of Software, Chinese Academy of Sciences (SYSKF1010), Beijing Municipal Natural Science Foundation (4102058), and the Fundamental Research Funds for the Central Universities. The second author acknowledges the partial support by grants from US NSF CISE NetSE program and SaTC program.

References

[1] D. Fensel, ”Ontology-Based Knowledge Management,” IEEE Computer, vol.35, no.11, pp.56–59, 2002.

[2] L. Chen N.R. Shadbolt, and C.A. Goble, ”A semantic Web-based approach to knowledge management for grid applications,” IEEE Trans. Knowl. and Data Eng., vol.19, no.2, pp.283–96, 2007.

[3] L. Razmerita, ”An ontology-based framework for modeling user behavior-A case study in knowledge management,” IEEE Trans. Syst., Man and Cybernetics, Part A, vol.41, no.4, pp.772–83, 2011.

[4] N. Shadbolt, T. Berners-Lee, and W. Hall, ”The semantic web revisited,” IEEE Intell. Syst., vol.21, no.3, pp.96–101, 2006.

[5] S. Philippi and J. Kohler, ”Using XML technology for the ontology-based semantic integration of life science databases,” IEEE Trans. Info. Tech. in Biomed., vol.8, no.2, pp.154–160, 2004.

[6] S. Kraines, W. Guo, B. Kemper, and Y. Nakamura, ”EKOSS: A knowledge-user centered to knowledge sharing, discovery, and integration on the semantic Web,” Proceedings of ISWC’06, pp.833–846, 2006

[7] M. Nagy and M. Vargas-Vera, ”Multiagent ontology mapping framework for the semantic web,” IEEE Trans. Syst., Man and Cybernetics, Part A, vol.41, no.4, pp.693–704, 2011.

[8] D. Vallet, M. Fernandez, and P. Castells. ”An ontology-based information retrieval model,” Proceedings of ESWC’05, pp.455–470, 2005.

[9] A. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani, ”Algorithmic Detection of Semantic Similarity,” Proceedings of WWW’05, pp.107–116, 2005.

[10] Y. Qu and G. Cheng. ”Falcons Concept Search: A Practical Search Engine for Web Ontologies,” IEEE Trans. Syst., Man, and Cybernetics, Part A, vol.41, no.4, pp.810–816, 2011.

[11] M. d’Aquin and N.F. Noy, ”Where to publish and find ontologies? A survey of ontology libraries,” Journal of Web Semantics, vol.11, no.8, pp.96–111, 2012.

[12] P. Porzel and R. Malaka, ”A task-based approach for ontology evaluation,” Proceedings of the ECAI workshop on ontology learning and population, 2004, Spain.

[13] J. Park, S. Oh, and J. Ahn, ”Ontology selection ranking model for knowledge reuse,” Expert Systems with Applications, vol.38, no.10, pp.5133–5144, 2011.

[14] Z. Khan and M. Keet, ”ONSET: Automated foundational ontology selection and explanation,” Proceedings of EKAW’12, pp.237–251, 2012.

[15] A. Doan, J. Madhavan, P. Domingos, and A.Y. Halevy. ”Learning to map between ontologies on the semantic web,” Proceedings of WWW’02, pp.662–673, 2002.

[16] N. Choi, I.Y. Song, and H. Han, ”A Survey on Ontology Mapping,” ACM SIGMOD Record, vol.35, no.3, pp.34–41, 2006.

[17] S. Kaza and H. Chen, ”Evaluating ontology mapping techniques: An experiment in public safety information sharing,” Decision Support Systems, vol.45, no.4, pp.714-728, 2008.

[18] M. Mao, Y. Peng, and M. Spring, ”An adaptive ontology mapping approach with neural network based constraint satisfaction,” Journal of Web Semantics, vol.8, no.1, pp.14–25, 2010.

[19] A.M. Khattak, Z. Pervez, K. Latif, and S. Lee, ”Time efficient reconciliation of mappings in dynamic web ontologies,” Knowledge-Based Systems, vol.35, no.11, pp.369–374, 2012.

[20] J. Li, J. Tang, Y. Li, and Q. Luo, ”RiMOM: A dynamic multi-strategy ontology alignment framework,” IEEE Trans. Knowl. Data Eng., vol.21, no.8, pp.1218–1232, 2009.
[21] A. Maedche and S. Staab, ”Measuring Similarity between Ontologies,” Proceedings of EKAW’02, pp.251–263, 2002.
[22] M. Popescu, J.M. Keller, and J.A. Mitchell, ”Fuzzy measures on the gene ontology for gene product similarity,” IEEE/ACM Trans. Comput. Bio. and Bioinfo., vol.3, no.3, pp.263–274, 2006.
[23] H. Al-Mubaid and H. Nguyen, ”Measuring semantic similarity between biomedical concepts within multiple ontologies,” IEEE Trans. Syst., Man and Cybernetics, Part C, vol.39, no.4, pp.389–398, 2004.
[24] D. Sanchez, M. Batet, D. Isern, and A. Valls, ”Ontology-based semantic similarity: A new feature-based approach,” Expert Systems with Applications, vol.39, no.9, pp.7718–7728, 2012.
[25] A. Rodriguez and M. Egenhofer, ”Determining semantic similarity among entity classes from different ontologies,” IEEE Trans. Knowl. and Data Eng., vol.15, no.2, pp.442–456, 2003.
[26] S. Tartir, I.B. Arpinar, M. Moore, A.P. Sheth, and B. Aleman-Meza, ”OntoQA: Metric-based ontology quality analysis,” Proceedings of IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources, 2005.
[27] A. Gangemi, C. Catenacci, M. Ciaramita, and Jos. Lehmann, ”A theoretical framework for ontology evaluation and validation,” Proceedings of SWAP’05, 2005.
[28] A. Lozano-Tello and A. Gomez-Perez, ”OntoMetric: A method to choose the appropriate ontology,” Journal of Database Management, vol.15, no.2, pp.1–18, 2004.
[29] H. Yao, A. Orme, and L. Etzkorn, ”Cohesion Metrics for Ontology Design and Application,” Journal of Computer Science, vol.1, no.1, pp.107–113, 2005.
[30] A. Orme, H. Yao, and L. Etzkorn, ”Coupling Metrics for Ontology-Based Systems,” IEEE Software, vol.23, no.2, pp.102–108, 2006.
[31] A. Burton-Jones, V.C. Storey, V. Sugumaran, and P. Ahluwalia, ”A semiotic metrics suite for assessing the quality of ontologies,” Data & Knowledge Engineering, vol.55, no.1, pp.84–102, 2005.
[32] H. Stuckenschmidt, ”A Semantic Similarity Measure for Ontology-Based Information,” Proceedings of FQAS’09, pp.406–417, 2009.
[33] Y. Li, Z.A. Bandar and D. McLean, ”An approach for measuring semantic similarity between words using multiple information sources,” IEEE Trans. Knowl. and Data Eng., vol.15, no.4, pp.871–882, 2003.
[34] A.G. Maguitman and F. Menczer, H. Roinestad and A. Vespignani, ”Algorithmic detection of semantic similarity,” Proceedings of WWW’05, pp.107–116, 2005.
[35] H. Zhang, Y.-F. Li, and H.B.K. Tan, ”Measuring design complexity of semantic web ontologies,” Journal of Systems and Software, vol.83, no.5, pp.803–814, 2010.
[36] R. Kontchakov, F. Wolter and M. Zakharyaschev, ”Logic-based ontology comparison and module extraction, with an application to DL-Lite,” Artificial Intelligence, vol.174, no.15, pp. 1093–1141, 2010.
[37] J. Lawler and B. Kitchenham, ”Measurement Modeling Technology,” IEEE Software, vol.20, no.3, pp.68–75, 2003.
[38] Y. Ma, B. Jin, and Y. Feng. ”Semantic oriented ontology cohesion metrics for ontology-based systems,” Journal of Systems and Software, vol.83, no.1, pp.143–152, 2010.
[39] F. Ensan and W. Du. ”A semantic metrics suite for evaluating modular ontologies,” Information Systems, Vol.38, No.5, pp.745–770, 2013.
[40] D. Vrandecic and Y. Sure. ”How to Design Better Ontology Metrics,” Proceedings of ESWC’07, pp.311–325, 2007.
[41] Y. Ma. ”Towards Stable Semantic Ontology Measurement,” Proceedings of ISWC’10, pp.21–24, 2010.
[42] G. Guizzardi, G. Wagner, and H. Herre. ”On the Foundations of UML as an Ontology Representation Language”, Proceedings of EKAW’04, pp.47–62, 2004.
[43] H. Zhuge, ”Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning,” IEEE Trans. Knowl. and Data Eng., vol.21, no.6, pp.785-799, 2009.
[44] B. Motik, B.C. Grau, I. Horrocks, and U. Sattler, ”Representing ontologies using description logics, description graphs, and rules,” Artificial Intelligence, vol.173, no.14, pp.1275–1309, 2009.
[45] S. Rudolph, M. Krotzsch, and P. Hitzler. ”Description Logic Reasoning with Decision Diagrams: Compiling SHIQ to Disjunctive Datalog”, Proceedings of ISWC’08, pp.435-450, 2008.
[46] M. Dean and G. Schreiber, ”OWL Web Ontology Language Reference”, http://www.w3.org/TR/owl-ref/.
[47] H. Bunke and K. Shearer, ”A Graph Distance Metric Based on The maximal Common Subgraph,” Pattern Recognition Letters, vol.19, pp.255–259, 1998.
[48] Z. Zou, J. Li, H. Gao, and S. Zhang. ”Mining Frequent Subgraph Patterns from Uncertain Graph Data,” IEEE Trans. Knowl. and Data Eng., vol.22, no.9, pp.1203–1218, 2010.
[49] H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone, and M. Vento, ”A comparison of algorithms for maximum common subgraph on randomly connected graphs,” Proceedings of SSPR/SPR’02, pp.85–106, 2002.

Yinglong Ma received the BSc degree and M.Sc. degree in 1999 and 2002, and received the Ph.D. degree in 2006 from Institute of Software, Chinese Academy of Sciences, Beijing, China, all in Computer Science. He is now an associate professor in School of Control and Computer Engineering, North China Electric Power University, Beijing, China. His research interests include Semantic Web, knowledge engineering, and Software Engineering and Formal Method. His research is primarily sponsored by the National Natural Science Foundation of China (NSFC) and the Ministry of Science and Technology of P.R.China.

Ling Liu is a full Professor in College of Computing, Georgia Institute of Technology. Her research focus is on performance, availability, security, privacy, and energy efficiency. Dr. Liu has published over 250 International journal and conference articles in the areas of databases, data engineering, and distributed computing systems. Dr.Liu served on the editorial board of IEEE Transactions on Knowledge and Data Engineering (TKDE) and VLDB Journal from 2004 - 2008. She is currently the Editor-in-Chief of IEEE Transactions on Service Computing (TSC). Dr. Liu's current research is primarily sponsored by NSF, IBM, and Intel.

Ke Lu received his PhD degree in 2003 in Department of Computer Sciences, Northwestern University, Xi'an, China. He is a full professor in University of Chinese Academy of Sciences, Beijing, China. His research interests include Intelligent Information Processing and Software Engineering. Dr. Lu's current research is primarily sponsored by NSFC and Beijing Municipal Natural Science Foundation.

Beihong Jin received her BS degree in 1989 from the Tsinghua University, Beijing, People's Republic of China, and her MS degree in 1992 and her PhD degree in 1999 from Institute of Software, Chinese Academy of Sciences, Beijing, China, all in computer science. She is now a full professor and the deputy director in Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences. Her research interests include Distributed Computing and Software Engineering. She has published over 60 research papers in international conferences and journals of these areas.

Xiangjie Liu received his PhD degree in 1997 from Northeastern University, Shenyang, China. He is a professor in School of Control and Computer Engineering, North China Electric Power University, Beijing, China. His research interests include neural network and automatic control. He has published over 60 papers in some international conferences and journals of these areas.