A coherence-checking algorithm for ontology integration

Kun Liu[1*], Xiaogao Yu[1], Weifeng Pan[2]

[1]*Department of Information Management, Hubei University of Economics, Wuhan, China*

[2]*School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hanzhou 310018, Zhejiang, China*

Received 1 June 2014, www.cmnt.lv

Abstract

Ontology integration can be used to solve heterogeneity of different information. Different to the usual global interpretation, distributed interpretation based on DDL is taken to interpret its semantics. As a result, traditional coherence checking algorithm is not suitable any more. In this paper, we propose an algorithm to measure coherence of ontology integration under distributed interpretation. In our proposal, ontology integration is taken as global ontology and local ontologies connected by ontology mapping. Consistency and coherence are viewed as different things. Then a two-phrases checking algorithm is designed to test coherence of ontology integration. Some experiments are made to test its feasibility. We compare checking results with other algorithms, especially with that under global interpretation. Our algorithm can improve efficiency to some degree, but it is subjected to mapping relations which are found by mapping tools.

Keywords: ontology integration, ontology mapping, ontology coherence

1 Introduction

Ontology integration has been used to resolve the heterogeneous of information [1-3]. Generally, ontologies are mapped to an upper ontology in order to communication among information sources. It is similar to global-as-view (GAV) pattern in relation database integration. In this case, ontology integration which comprises a set of ontologies is viewed as a whole large one and given a global interpretation. This method makes it easy to reuse classical description logic reasoner to check satisfiability and make further diagnoses. But in some cases, ontologies cannot be putted together because of security, scalability and some other reasons. As a result semantics based on global interpretation will not be suitable any more.

Concerned to those situations, we have proposed a distributed interpretation based on DDL [4] to interpret semantics of ontology integration. A set of interpretations explain semantics of each local ontology and global ontology respectively and use semantics import to interpret mapped relations between them. Our method can make understanding and maintenance of ontology integration more convenient.

Under distributed interpretation, some issues are discovered. One of important issues is that how to measure consistency of ontology integration by using distributed interpretation. In this paper, we state that there is difference between consistency and coherence. An algorithm to measure coherence is designed and proved its feasibility through experiments. Our algorithm can improve efficiency of coherence checking to some degree. But our algorithm takes mapping relations found by mapping tools and affected by these automated tools.

The rest of paper is organized as follows: Section 2 explains our method of ontology integration based on DDL and semantics based on distributed interpretation. Section 3 introduces our proposed algorithm. In Section 4, we make some experiments on our algorithm and compare with other methods. Section 5 introduces some related work. Conclusions are made in Section 6.

2 Semantics based on distributed interpretation

In our method [5], ontology integration is denoted by $T=<Tg,\{Ti\},\{B_{ig}\}>$. Tg means global ontology and Ti represents each local ontology. B_{ig} shows that one of local ontologies has mapping relations with global ontology. It comprises two kinds of mapping: concept mapping and role mapping. Concept mapping includes three types of relations:

(i) equivalence: =

(ii) into: $\leq$

(iii) disjointness: $\perp$

A role mapping from ontology Ti to Tg includes two types of relations:

(i) equation: =

(ii) into: $\leq$

There are relations between global ontology and local ontologies in information integration and they should not be separated. It means that some concepts or roles have direct relations between global ontology and local ontologies. It is similar with some cases of ontology reuse, but ontologies on each side do not include any syntactic symbols from the other side.

* ***Corresponding author's***-e-mail: lovecpp@sohu.com

2.1 DISTRIBUTED INTERPRETATION

Global ontology and local ontologies have concepts or roles. A distributed interpretation which is denoted by *I* comprises a set of interpretation *I*=<{{Ii},Ig}>. Ig denotes interpretation of global ontology and Ii denotes that of each local ontologies.

For concepts or roles which are mapped to global ontology, there domains are listed as follows. Of course, they conform to classic definitions of DL.

(i) if i:C is a class name in Ti then (i:C)Ij=(i:C) Ii∩⊿Ij.

(ii) if i:R is a role name in Ti, then for all d∈⊿Ii ∩⊿Ij and for all d'∈$⊿_{Ij}$, <d,d'>∈R_{Ij}, if <d,d'>∈$(i:R)_{Ii}$.

A mapping i:C to j:C is satisfied when (i:C) Ij∈(j:C)Ij. When *I* can satisfy all mappings and each concept and axiom in all global ontology and local ontologies, then *I* is a model of *T*.

From a syntactic point of view, mappings do not appear on global or local side. It lists concepts or roles names and their relation type.

In the situation of information integration, global ontology is the center and its semantics is the most important. All other local ontologies' semantics should conform to it.

For ontology integration, mapping relations point from local ontologies to global ontology. Under this situation, the semantics of global ontology should maintain and its consistency should not violate.

Global interpretation is a nature way to explain ontology integration which means that integrated ontologies are seen as a whole ontology. In this way, many existing methods and tools can be reused. But, if there are so many heterogeneous ontologies and so much dissimilarity, it is hard work to check consistency and repair inconsistency, especially when the amount of ontologies reaches a degree.

2.2 CONSISTENCY AND COHERENCE

In [6], it regards consistency and coherence as different things.

Definition 1 (consistency): for an ontology integration *T*, if a model can be found to satisfied *T*, then *T* is consistent, or else it is inconsistent.

But an empty model can satisfy an ontology integration when a concept is empty. Coherence is used to describe this situation. In case of information integration, an empty concept is useless because of it cannot transfer data from one ontology to another. So these empty concepts should be found.

Definition 2 (coherence): for an ontology integration *T*, if a mapped concept i:C is not satisfied, *T* is not coherent.

An ontology integration T may be consistent, but when a mapped concept is empty, it is not coherent.

***Example 1**:*

T_1:C ≤ D

T_2: E⊥ F

B: 1:C ≤ 2:E

1:C ≤ 2:F

In this example, 1:C is mapped to 2:E and 2:F in ontology T_2. Because E and F are disjoint, 1:C cannot be interpreted by an interpretation. It is empty. So *T* is not coherent.

3 Coherency checking algorithm

3.1 REDUCTION

In our method, relations between global ontology and local ontologies are expressed with bridge rules. Because these rules cross ontologies, they can not be directly used by reasoner such as Pellet. Our method translates these bridge rules into axioms through reduction. The rules of reduction are listed as follows.

1) Into mapping between i:C and g:D is reduced to subsumption axiom i:C ≤g:D.

2) Equivalence mapping between i:C and g:D is reduced to equivalent axiom i:C =g:D.

3) Disjointness mapping between i:C and g:D is reduced to disjoint axiom i:C ⊥g:D.

3.2 PROPOSED ALGORITHM

Based on the definition of coherence of ontology integration, we give the following coherence checking algorithm in Table 1. It comprises two phrases. Firstly, consistency of ontology integration is checked. And then finding empty mapped concepts, if an empty one is found, its coherence is decided.

Firstly, it checks coherence of global ontology and each local ontology separately. We should separate them from *T* (line.1) and then it will invoke the classical DL reasoner, such as Pellet, Racer Pro etc. to perform this work (line2-5).

Secondly, mapping relations are retrieved from global ontology and each local ontology (line6). Generally, these relations are translated from the results of ontology matching tools, such as Alignment API and OLA etc. Whatever the format of matching result is, they all will be expressed with axioms which are stated aforementioned. Based on those changes, these mapping relations can be handled by DL reasoners. Obviously, all these are based on reduction principles.

These mapping relations are combined with global ontology (line7) and tested whether they are coherent or not (line8). This results from the semantics proposed in our approach. According to the definition, when a class or role is mapped into another ontology, its meaning is subjected to that ontology. Its original relations with other concepts can not be transferred to that ontology.

When all these steps are performed, true is returned to show the ontology integration is coherent.

TABLE 1 Consistency checking algorithm

Input: an integrated ontology *T* Output: true or false	
01.	Load ontology integration O={T_i}
02.	For each T_i in O
03.	{
04.	Invoke sub_function *UnsatisfiableConcepts_Detecting* to check whether T_i is coherent or not. If T_i is not coherent Return false;
05.	}
06.	Extract B={ B_{ig} } from O
07.	Combine B and global ontology T_g, and invoke sub_function *UnsatisfiableConcepts_Detecting to* check theire coherence.
08.	If (T_g $\cup$ B is not coherent) Return false;
09.	Return true;
sub_function *UnsatisfiableConcepts_Detecting* Input: an ontology T Output: true or false	
10.	Load ontology T
11.	Ontology list clss equavilent(nothing) and
12.	If (unsatisfialeConcepts.size is zero) Return true
13.	else Return false

In line 10-13, a function named *UnsatisfiableConcepts Detecting* is defined to find empty concepts in an ontology and invoked by line 2-5 and 7. In OWL reasoners, empty concepts is expressed by class *Nothing*. We only find out those concepts equivalent to class *Nothing* and count is size. If there are any empty concept, its size will be zero. It means that the ontology includes unsatisfiable concepts.

4 Experiments

Compared to global interpretation, distributed interpretation has some advantages. It conforms to the status of scattered ontologies. For example, as to the case of ontology integration in Figure 1, four ontologies will be checked to test whether they are consistent or not. But depending on distributed interpretation, there is no need to check all these four ontologies at the same time. It first check the consistency of ontology UNIV, University, College and Publication respectively and then check the combination of UNIV and those concepts and roles which are mapped from University, College and Publication to UNIV.

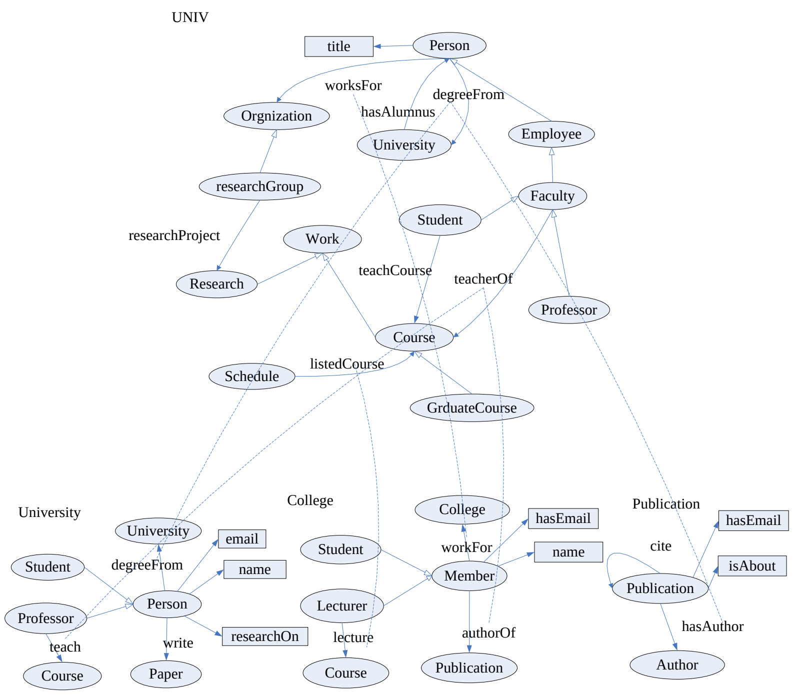

FIGURE 1 An example of ontology integration

The following Table 2 lists expressivity, classes, properties and axioms of each ontology in Figure 1.

TABLE 2 Ontologies

Ontology	Expressivit	Classes	Properties	Axioms
MIT	ALCH	20	40	140
UMBC	ALCIN	18	31	96
AIFB	ALCI	58	73	235
INR	ALCH	39	96	232

We use OLA tools to find mapping relations between three local ontologies and global ontology.

In Figure 2, D means consuming time of our method and G of global checking algorithm. We also test consuming time of each ontology and mark with Aifb, Mit and Umbc. As we can see in Figure 2 that our algorithm apparently improve efficiency of checking coherence than that with global interpretation. It due to its semantics based on distributed interpretation and reduce amount of mapping relations to check when these ontologies are integrated. Compared to checking on single ontology, our algorithm show different cases.When local ontology is mapped to global ontology, only a part of classes or properties are mapped and quantity of mapping relations added to global ontology may be different.

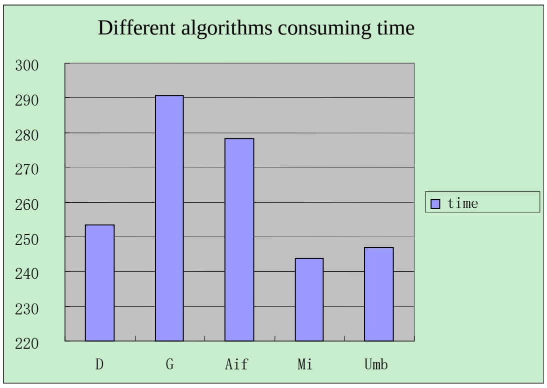

FIGURE 2 Efficiency of different algorithms

5 Related work

Jimenez-Ruiz and Grau [7] base their work on global interpretation. They propose a framework named ContentMap to check and repair consistency which fully makes use of existing ontology debugging technology.

Similar to ontology integration, Fahad [8] talks about semantics of ontology merge and algorithms of checking consistency. As Flouris [6] have talked about, there are some differences between ontology integration and ontology merge.

Some work proposes distributed Tableau algorithms to check consistency or coherence of integrated ontologies, but each one of these algorithms only adapt to some specified integrated ontologies.

Serafini [9,10] continues to design reasoning algorithms of DDL. In [4], DDL is discussed. Borgida and Serafini introduce distributed description logic to express ontologies connected with ontology mappings which called bridge rules. They use domain relation to interpret bridge rules.

Pan [11] talks about how to check satisfiability of ontology reuse. Their work ared based on ontology reuse which comprises ontology space and extends Tableau algorithm to check coherence of ontology reuse.

Bao [12] also proposes a distributed algorithms to check P-DL ontology which considers modular ontologies. They take this kind of ontology as SHOIQP and propose a revised P-DL semantics.

6 Conclusions

In this paper, we have talked about coherence checking algorithm for ontology integration under the background of information integration. Consistency and coherence are distinguished according to distributed interpretation which is used to interpret semantics of ontology integration. For the purpose of information integration, coherence can check whether a concept is empty or not. It is important to information integration. In our algorithm, bridge rules describing mapping relations between global ontology and local ontology are reduced to axioms. Then a two phrases algorithm is proposed to check coherence of ontology integration. Due to distributed interpretation, coherence checking is performed on each ontology, not on whole integrated ontologies. Through experiments, our algorithm is found out that the efficiency of checking is improved.

Nowadays, our approach adopts OLA tools to help find into and equivalence mapping relations between global ontology and local ontologies and handle disjointness mapping by man. Our algorithm is subjected to mapping relations which are found by mapping tools. In the future, we will make some research on how to find disjointness mapping relations by tools.

Acknowledgments

This work is supported by Science and Technology Research Funds of Educational Ministry of Hubei Province, China (No. B2013034).

References

[1] Wache H, Vogele T, Visser U, Schuster G, Neumann H, Stuckenschmidt H, Hubner S 2001 Ontology-based Integration of Information – A survey of existing approaches *Proceedings of International Joint Conference of Artificial Integlligence Seattle WA* 108-17

[2] Chen H, Wu Z, Wang H, Mao Y 2006 RDF/RDFS-based Relational Database Integration *Proceedings of International Conference on Data Engineering*

[3] Dou D, Le Pendu P 2006 Ontology-based Integration for Relational Databases *Proceedings of SAC2006 Dijon France* 461-6

[4] Borgida A, Serafini L 2002 Distributed Description Logics: Directed Domain Correspondences in Federated Information Sources *Proceedings of CoopID/DOA/ ODBAS2002* 36-52

[5] Liu K, Tang S, Pan W 2012 Semantics based on Distributed Integration for Ontology Integration *Journal of Theoretical and Applied Information Technology* **45**(2) 681-7

[6] Flouris G, Huang Z, Pan J Z, et al.(2006 Inconsistencies, negations and changes in ontologies *Proceedings of the National Conference on Artificial Intelligence(AAAI)* **21**(2) 1295-300

[7] Jimernez-Ruiz E, Grau B C, Horrocks I, Berlanga R 2009 Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences *Proceedings of Europe Semantic Web Conference* 173-87

[8] Fahad M, Moalla N, Bouras A 2011 Towards ensuring Satisfiability of Merged Ontology *Journal of Procedia Computer Science* **4** 2216-25

[9] Serafini L, Tamilin A 2005 DRAGO:Distributed Reasoning Architecture for the semantic web *Proceedings of Europe Semantic Web Conference* 361-76

[10] Serafini L, Borgida A, Tamilin A 2005 Aspects of distributed and modular ontology reasoning *Proceedings of International Joint Conference of Artificial Integlligence* 570-5

[11] Pan J Z, Serafini L, Zhao Y 2006 Semantic Import: An Approach for Partial Ontology Reuse *Proceedings of 1st International Workshop on Modular Ontologies* 1-13

[12] Bao J, Caragea D, Honavar V 2006 A Distributed tableau algorithm for package-based Descrption Logics *Proc. of the 2nd International Workshop on Context Representation and Reasoning*

Authors	
	**Kun Liu, 20.07.1979, China.** **Current position, grades**: researcher at Hubei University of Economics, China. **University studies**: Ph.D. degree in computer science and technology from Wuhan University, China in 2013. **Scientific interest**: ontology, software engineering and cloud computing.
	**Xiaogao Yu, 12.08.1969, China.** **Current position, grades**: professor at Hubei University of Economics, China. **University studies**: PhD degree in computer science and technology from Wuhan University of Technology, China in 2007. **Scientific interest**: data mining, business intelligence and information system. **Publications**: more than 20 papers.
	**Weifeng Pan, 27.11.1980, China.** **Current position, grades**: associate professor in School of Computer Science and Information Engineering at Zhejiang Gongshang University. **University studies**: PhD degree in Computer Science from Wuhan University, China, in 2011. **Scientific interest**: software engineering, service computing, complex networks and intelligent computation. **Publications:** more than 30 papers.