A method of conflict checking for ontology integration based on distributed interpretation

Kun Liu
School of Information Engineering
Hubei University of Economics
Wuhan, China
lovecpp@sohu.com

Weifeng Pan
School of Computer Science and Information Engineering
Zhejiang Gongshang University
Hangzhou, China
panweifeng1982@gmail.com

Xiaogao Yu
School of Information Management
Hubei University of Economics
Wuhan, China
yxg@hbue.edu.cn

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. A phenomenon called conflict is found on this situation. We define what is conflict and make some difference with global interpretation. Then we design its checking algorithm. Depending on ontology closure, all relations of a concept pair are found and sent to global ontology to see whether one of them triggers conflict. Our method is suitable for the situation under which local ontology can not be opened to public for the safety , secret or other reason. Last a simulation is given to verify its feasibility.

Keywords- ontology integration; distributed interpretation; conflict; ontology closure; concept pair

I. INTRODUCTION

It is a nature method to interpret semantics of ontology integration which is used to resolve the heterogeneous of information[1] by viewing integrated ontologies as a whole ontology which is given a global interpretation. Concerned to those situations, we have proposed a distributed interpretation based on DDL[2-3] to interpret semantics of ontology integration. A set of interpretations explain semantics of each local ontology and global ontology respectively and use domain relation to interpret mapped relations between them.

But on this situation, a phenomenon is happened even though ontology integration is consistent on the semantics depending on distributed interpretation. In this paper, conflict is used to define this phenomenon. Because most research rely their semantics on classic description logic, this phenomenon is seen as one kind of inconsistency. The usual way is to debug ontology mapping and find incorrect mappings[4-5]. Relying on ontology closure, all relations of a concept pair are found and each of them is sent to global ontology to check whether it triggers conflict. In this way, it is suitable for the situation under which local ontology can not be opened to public for the safety , secret or other reason. A simulation is given to verify its feasibility.

The rest of paper is organized as follows: Section 2 introduces some preliminary knowledge about distributed interpretation for ontology integration. Section 3 discusses our proposed method and gives a corresponding algorithm. In section 4, we make a simulation on our algorithm. Section 5 introduces some related work. Conclusions are made in section 6.

II. PRELIMINARY KNOWLEDGE

A. Distributed interpretation

In our method[6], ontology integration is denoted by T= <Tg, {Ti}, {Big}>. Tg means global ontology and Ti represents each local ontology. Big 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: = ,$\leqslant$ and $\perp$. A role mapping from ontology Ti to Tg includes two types of relations: = and $\leqslant$.

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. 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 be violated.

B. Difference with semantics on single domain

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.

Take an example with classic question whether Penguim can fly.

Example1：

T1: Penguim $\subseteq$ Bird

Tg: Flying $\subseteq \neg$ notFlying

B: 1:Penguim ≤ g:notFlying

1:Bird ≤ g:Flying

In this example, Pengium is mapped to as subclass of notFlying and Bird is mapped to as subclass of Flying. If this ontology integration is interpreted with classic description logic, apparently, this ontology integration is not consistent because of the fact that Penguim can fly is not confirmed.

But using our proposed distributed interpretation, it is consistent. First, T1 and Tg is respectively consistent. When Penguim and Bird are mapped to Tg, according to semantics of Tg, Penguim belongs to notFlying and Bird to Flying. The meaning of Penguim and Bird subjects to Tg.So they don’t make Tg inconsistent. At the same time, Penguim is still a kind of Bird. This fact is only tenable in T1.

III. CONFLICT AND ITS CHECKING METHOD

A. Definition of conflict

Example 2:

T1:C ⩽ D

Tg: E⊥ F

B: 1:C ⩽ 2:E

1:D ⩽ 2:F

In this example, C belongs to D in T1. They are respectively mapped to Tg as sub classes of E and F. On this situation, when a user query who are sub classes of F. Tg will response this question. It distributes this question to all local ontologies according to mapping relations. It finds 1:D. But in T1, C belongs to D. Should C be an element of the final answer? This example is similar to example1. If a user asks which animal can fly, according to our general knowledge, Penguim should not be returned. So in example2, 1:C should not be a part of final answer.

Here, a stranger phenomenon appears. C is a sub class of D in T1. But it cannot be return as sub class of 2:F though 1:D is sub class of 2:F. To explain this phenomenon，conflict is defined.

Definiton1(*concept pair*): A concept pair is expressed by Sig(C,D) which represents two concepts have some relations.

Defintion2(*conflict*): If Sig(C,D) ∪ B ∪ Tg is not consistent, the concept pair C and D trigger conflict of global ontology.

If global interpretation is used to interpret semantics of ontology integration, conflict will be viewed as one kind of inconsistency. For example, if global interpretation is used on ontology integration of example1, Penguim is a kind of Bird. And at the same time, Bird is mapped to Flying. So Penguim can fly. This contradicts to the fact that Penguim is notFlying. Our definition is different from the situation that mapping triggers inconsistency of ontology integration.

Conflict is defined a from global ontology point of view, because it is global ontology to handle queries from users. This is decided by role of ontology integration for solving heterogeneous information.

When conflict exists on some concept pairs, there should be some mechanisms to tell Tg to properly handle global query. A natural way is to find out those concept pairs which will trigger conflict. When these concept pairs are confirmed, they will be stored by the side of local ontologies and be used when a query is handed out from global ontology.

B. Our proposed method

From definition 2, it is known that the essential of conflict is to see whether relation of a concept pair such as <C,D> exists contradiction before and after their being mapped to global ontology.

In ontology integration T, concept C and D of one local ontology may change their relation when they are mapped to global ontology. This situation can be divided to three kinds:

- (i) C and D have relation which disappears when they are mapped to global ontology.
- (ii) C and D have no relation in local ontology and have some relation after being mapped to global ontology.
- (iii)C and D have relation, but change to another kind of relation.

But for an ontology, we can not directly know whether C and D have relation, because asserted axioms only tell us explicit knowledge and there are much implicit knowledge which may means that C and D has some relations.

Explicit knowledge can be directly got from asserted axioms and implicit knowledge can be got from inferred axioms which can be reasoned from the closure of an ontology[7-8].

Of course , an anonymous class is not be considered because there seldom exists query on anonymous class from users.

After being mapped to global ontology, the relation of C and D is handled by global ontology. So We can first get all relations of concept pair<C,D> and send each of them to global ontology and check whether it trigger inconsistency of global ontology. This means our method takes at least two steps: finding all relations of concept pair through ontology closure and then focusing on the relation of C and D after they are mapped to global ontology.

TABLE I. FINDING CONFLICT CONCEPT PAIRS ALGORITHM:

Input: a local ontology *T*,global ontology *Tg* and bridge rules *B* Output: conflict set	
01.	load ontology T, Tg and turn B into OWL axioms
02.	compute inferred axioms of T though ontology closure
03.	extract mapped concepts from B
04.	for each pair <C, D> of T
05.	{ if (getting NTriples about <C,D> is not null)
06.	record NTriples in a map with key <C,D>
07	}
08.	merge B and Tg
09.	for each NTriple in map
10.	{ turn NTriple into axiom ax
11.	check consistency of {ax}with B and Tg, if not consistent, then record this NTriple in conflict concept pair map with key <C,D>
12.	}
13.	return conflict concept pair map

C. Finding out conflict concept pairs

Table I shows an algorithm to find out those conflict concept pairs. This algorithm is mainly divided to two phrases: line 1-7 compute all relations including asserted and inferred axioms of all concept pairs for every local ontology. Line 8-12 compute conflict concept pairs.

Line 2 computes inferred axioms comes from ideas of [7-8]. In our algorithm they are expressed with NTriples, such as：

[
http://www.hbue.edu.cn/sie/university#Student,
http://www.w3.org/2000/01/rdf-schema#subClassOf,
http://www.hbue.edu.cn/sie/universtiy#Person
]

But in the following consistency checking in line11, they will be translated into OWL axiom which Pellet or other reasoners can use.

Line3-7 shows how to find related axioms about concept pair<C,D>. But not all concept pairs of a local ontology T should be concerned. We choose those concepts which are mapped to global ontology and check their pairs. This can be understood from example 1. If another kind of bird such as Parrot is not mapped into global ontology. When an user asks subclasses of Flying, it will be returned definitely. So the pair <Parrot, Bird> is not considered ,but <Penguim, Bird> should be.

If we do not compute inferred axioms about concept pairs, their relations will be acquired when global ontology distributes query to this local ontology. To save time, we take the strategy showed in this algorithm.

When relation of a concept pair <C,D> is found, it will be checked whether triggers conflict of Tg. Line 8-12 shows this process which depends on definition 2. For concept C and D are respectively mapped to global ontology, their mappings in B are require. So it needs to merge B and Tg. After getting relation of concept pair from NTriple, it is combined with B and global ontology. Though classic descripion logic reasoner such as Pellet, thier consistency is easily verified. Then conflict is confirmed.

In this algorithm, a kind of data structure is employed which is map that has key and value. It is convenient to get value from a key for a map. For a concept pair <C,D>, there may be more than one kind of relation found out in an ontology and some of them trigger conflict. So <C,D> is used as key of map and their relations as value which is hold in an array or some other data structures. If a query is distributed to this ontology, it is very convenient to get relations for the key <C,D>.

The algorithm repeatedly checks consistency of global ontology and each time with different relation of a concept pair. This should be improved in the future work.

IV. SIMULATIONS AND ANALYSIS

Simulation circumstance : CPU corei51.7G, 4G RAM, Windows8 + Pellet2.0 + Eclipse + OLA. OLA is mainly used to find mapping relations between each local ontology and global ontology.

The simulation adopts ontologies from bibliographic reference domain[9] which are named MIT, UMBC,AIFB and INC. INC is global ontology and the three others are local ontologies. We use OLA tools to find mapping relations between three local ontologies and global ontology. Because OLA is an automatic mapping tool, some errors cannot be avoid. It requires manual work to move those apparent incorrect mappings. In table II, we can see that finally MIT has 12 concepts mapped to INC over its 20 concepts, UMBC has 12 over 18 and AIFB has 20 over 58.

TABLE II. CONCEPT PAIRS

ontology	concepts	mapped concepts		concept pairs	
		quantity	percent	effective	percent
MIT	20	12	60%	11	8.3%
UMBC	18	12	66.7%	24	18.2%
AIFB	58	20	34.5%	97	25.5%

Finding concept pair conflict algorithm runs every local ontology with its mapping relations and global ontology. TableII lists their result. For example, MIT has 12 concepts mapped to global ontology. Each concept may have relation with other 11 concepts. In our method, a concept pair <C,D> is seen as different with <D,C>. So it has 12*11 concept pairs on theory. After running algorithm, there are 11 concept pairs which don't trigger conflict. The other two ontologies are showed in table II.

In this simulation, each local ontology is mapped to global ontology and concept pairs are computed respectively. For example, MIT is mapped to INC. After running algorithm in tableI, all concept pair are found and each of them is sent to INC to check whether it can trigger conflict. It gives us an inspiration to use parallel computing techniques to compute concept pairs and check conflict.

V. Related work

Chang and Xu[10] propose a suffage algorithm which transforms mappings into a matrix and find inconsistency of the structure. Because this method takes the view point of structure, it is efficient to find improper mapping relation. But for complexity of ontology, some concepts may not be transformed to edges of matrix, this affects accuracy of mapping inconsistency checking.

Jimenez-Ruiz and Grau[11] base their work on global interpretation. They propose a framework named *ContentMap* to check and repair consistency which makes full use of existing ontology debugging technology. This work can find improper mapping relations very well and accuracy is better than referecne10. But it need to put all ontologies together and this hinders efficiency of finding those imporper ontology mapping relations. If some local ontologies are banned to public use for considering safety or other reason, this method will lose effectiveness.

Giuchiglia[12] computes minimal mappings on the all possible correspondences. They define several relations of mappings and identify four redundancy pattern. Trough moving redundant mappings, minimal mappings can be got between two lightweight ontologies.

VI. Conclusions

In this paper, we have talked about conflict in the situation of distributed interpretation. Different to semantics on single domain, conflict is triggered when concepts are mapped to global ontology and relation is changed. Many other methods view conflict as one kind of inconsistency. First we give definition of conflict and make some difference with global interpretation. Then we propose our method which depends on concept pair. By using ontology closure, all relations of a concept pair can be computed. Each of these relations is sent to global ontology to see whether it triggers conflict. Finally though a simulation, the feasibility of our method is verified.

For local ontologies can be stored individually on distributed interpretation. In the future, parallel computing will be introduced to check conflict of concept pairs from local ontologies.

Acknowledgment

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

References

[1] D. Fensel, F.Baader, M. Rousset and H. Wache,"Heterogeous information resources need semantic access", Data&Knowledge Engineering, vol. 36, Mar. 2001, pp:211-213,doi:10.1016/S0169-023X(00)00046-X.

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

[3] M. Atencia, Alexander Borgida, Jérôme Euzenat, Chiara Ghidini, Luciano Serafini "A Formal Semantics for Weighted Ontology Mappings", 11th International Semantic Web Conference, Boston, USA, nov. 2012, Proceeding, Springer Berlin Heidelberg, pp:17-33Doi:10.1007/978-3-642-35176-1_2.

[4] Peng Wang, Baowen Xu. "Debugging ontology mappings: a static approach", Journal of Computing and Inforrmatics, vol.27, 2008, pp:21-36.

[5] Meilicke,C., Stuckenschmidt, H., Tamilin, "Reasoning Support for Mapping Revision", Journal of Logic and Computation, vo.19, Feb. 2009, pp: 807-829. doi: 10.1093/logcom/exn047.

[6] Kun Liu, Shengqun Tang, Weifeng Pan. "Semantics based on Distributed Integration for Ontology Integration", Journal of Theoretical and Applied Information Technology. vol. 45, Nov. 2012, pp:681-687.

[7] D. Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio lenzerini and Riccardo Rosati. "Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Framily", Journal of Automated Reasoning, vol.39, 2007, pp:385-429, doi:10.1007/s10817-007-9078-x.

[8] F. Baader, Martin Knechtel, Rafael Penaloza. "Context-dependent views to axioms and consequences of Semantic Web ontologies", Journal of Web Semantics, vol.12-13, 2012, pp:22-4,. doi:10.1016/j.websem.2011.11.006.

[9] FLOURIS G,MNAKANATAS D, KONDYLAKIS H, et al. Ontology change: classification and survey[J]. The Knowledge Engineering Review, 2008, 23(2): 117-152.

[10] Si Chang, Qingjun Zhang, Xin Liu, Bo Liu, Dezhi Xu. Sufferage Extraction Algorithm of Ontology mapping Based on Reasoning. In: Proc. of ICACTE 2008, IEEE computer society press. pp.215-219. 2008.

[11] E.Jimernez-Ruiz, B.C.Grau, I.Horrocks, and R. Berlanga, "Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences", *Proceedings of Europe Semantic Web Conference*, Springer Heidelberg, May.2009, pp:173-187, doi:10.1007/978-3-642-02121-3_16.

[12] F. Giuchiglia, Vincenzo Maltese and Aliaksandr Autayeu. Computing Minimal Mappings between Lightweight Ontologies. International Journal of Digital Library, vol.11(2). 2012.