Detecting Semantic Mapping of Ontologies with Inference of Description Logic

wenhui Ma, guangping Xu, gang Wang, jing Liu
College of Information Science and Technology,
Nankai University
Tianjin 300071, China
{wenhuima_nk}@hotmail.com

Abstract

Ontology is increasingly seen as a key factor of semantic web. It makes information on the web more readily accessed and processed by applications. Ontologies mapping is required for combining independent and heterogeneous ontologies, which implements the interoperability across semantic web applications. This paper presents a new approach for ontologies mapping. We introduce the inference techniques of description logic, and transform the ontologies mapping from the computing of linguistic or structural similarities to the logical reasoning between concepts of different ontologies, implementing the semantic mapping of ontologies. Experiments show our approach outperforms the CTXMATCH algorithm in recall and precision when the rich relations exist between concepts of ontologies.

1. Introduction

The Semantic Web community has achieved a good standing within the last years. The aim of Semantic web is to make information on the web more readily accessed and processed by applications or automated tools (i.e., automated agents). This is to be done through representing the content of data with ontology. Ontology has been defined as "explicit conceptualization of a domain"[1], in which information is organized into taxonomies of concepts. The concepts are labeled by terminologies, and they provided model entities of interest in the domain and are connected by relations(non-taxonomical relations). Each concept denotes a set of instances and represents the meaning of them. Unfortunately, given the decentralized nature of semantic web development, it is likely that there will be an explosion in the number of ontology. Many of them will use different terminologies and structures to describe the same or related domains. Information processing across ontologies is not possible without knowing the semantic mapping between their elements. Thus, semantic mapping of different ontologies becomes a core question.

Currently, ontology mapping is largely performed manually by domain experts, therefore a time-consuming, tedious and error-prone process[2]. In this paper we describe an approach, which applies inference technique of description logic to automatic ontology mapping. Our approach is based on the CTXMATCH algorithm proposed in [3], which is a propositional logic based algorithm. However, given the ontology, which defines diverse relations and restrictions among concepts, the power of description and reasoning of propositional logic is not enough. Our work aims to uses description logic to describe the concepts and make explicit the meaning of concepts in the ontology. The semantic mapping of ontologies is deduced via inference of description logic.

This paper is organized as follows. The related work about ontology mapping is provided in Section 2. Section 3 presents our approach of ontology mapping. And Section 4 presents an experiment evaluation. Finally, Section 5 reports conclusions.

2. Related work

Numerous researchers have addressed the ontology mapping problem. Their methods come from different disciplines such as data analysis, machine learning, language engineering, statistics or knowledge engineering. To achieve high accuracy for a large variety of ontologies, a single strategy may be unlikely to be successful. Hence, to combine different approaches is an effective way[2]. Good survey through recent years are provided in [4][5][6]. the survey of [5] focuses on schema matching, and [4][6] survey a set of methods, systems and tools related to ontology mapping. We mainly show the CTXMATCH algorithm that is related to the approach introduced in this paper in the next paragraph.

CTXMATCH is an algorithm for discovering semantic mappings across hierarchical classifications (HCs) using logical deduction. CTXMATCH takes two inputs H_1 and H_2 in HCs, and for each pair of concepts $C_1 \in H_1$, $C_2 \in H_2$, the mapping problem with the relations ($\supseteq$, $\subseteq$, $\equiv$, *, and $\perp$) between them is translated into a propositional formula of form and checked for validity. The contribution of the CTXMATCH is that mappings can be assigned a clearly defined model theoretic semantics and that structural, lexical, and domain knowledge are considered. But there is a limited in the CTXMATCH algorithm: it only deals with unary predicates, and for the binary predicates, such as properties or roles, it can not handle[5]. In fact, it is only an algorithm of schema matching. The method proposed in this paper extends the CTXMATCH algorithm. We exploit the expressive power of description logic and its efficient reasoning techniques to implement ontology mapping.

3. Ontology mapping with inference of DL

3.1 Ontology mapping

Ontology mapping takes two ontologies as inputs and finds semantic relationships between the entities (concepts, relations, etc.) in the two input ontologies[7]. In this paper, we focus on finding one-to-one mapping between concepts of ontologies. Successfully mapping between concepts will greatly aid in mapping between other entities of the ontology. The specific problem that we consider is as follows: given two ontologies O_1 and O_2, semantic mapping between them means: for each pair of concepts from O_1 and O_2 respectively, we try to find the semantic relationships between them. Four semantic relationships can be held by concepts C_1 and C_2 in our approach, which are shown in the following:

- ♦ $C_1 \subseteq C_2$ means C_1 is less general than C_2, i.e. the meaning of C_1 is included by the meaning of C_2;
- ♦ $C_1 \supseteq C_2$ means C_1 is more general than C_2, i.e. the meaning of C_1 includes the meaning of C_2;
- ♦ $C_1 \equiv C_2$ means C_1 is equivalent to C_2, i.e. C_1 and C_2 have the same meaning;
- ♦ $C_1 \perp C_2$ means C_1 is disjoint from C_2, i.e. C_1 and C_2 have not the semantic relationships.

3.2 Description Logic

We follow the approach of semantic coordination described in [3], and use logic formulas to represent the concepts of ontologies. But different to the [3] that present the logic formulas with propositional logic, we use Description logic(DL)[8] to construct the logic formulas.

DL is a family of knowledge representation languages that can been used to represent the knowledge in a structured and formally well understood way. DL equips with a formal, logic based semantics, and it can describe the relations and restrictions over concepts, which can not be done by propositional logic. For example, for the concept "Network" in the ontology O_2 showed in Figure 1, it has relations "teachBy" and "teachFor", so its logical formulas of DL *logic_formula(Network)* is showed as follows:

$$
\begin{aligned}
&logic\_formula\ (Network) \\
&\quad = Network \sqcap Course \sqcap \\
&\qquad Computer\_Science \sqcap \\
&\qquad \forall\, teachBy.Lecture \sqcap \\
&\qquad \forall\, teachFor.Undergraduate
\end{aligned}
$$

In this paper, we look at the logic formulas of concepts as complex concepts of DL, and accordingly the concepts and relations of ontology are as the atomic concepts and roles of DL. The complex concepts capture the meaning of the corresponding concepts in the ontologies. Such as for the concept "Network", its complex concept(logical formula) represents its meaning in ontology O_2 as "Network is a course of computer science, and it is taught by lecturer and for undergraduate". Subsequently, semantic mapping between concepts of different ontologies are discovered by determining the semantic relationships between their corresponding complex concepts.

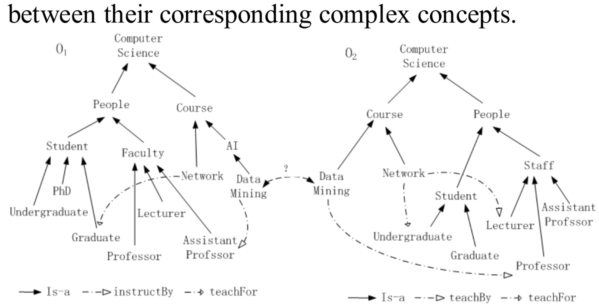

Figure 1. Two independent ontologies of computer science department

We use inference of DL to get semantic relationships between complex concepts. Before to do it, two things must be done. Firstly, the labels of atomic concepts and atomic relations in the complex concepts are rewritten in consistent forms. Secondly, the relationships in real world between atomic concept labels in complex concepts are found, which are as the premise for reasoning.

3.3 Consistency in labels

Because of the richness of natural language, different ontologies maybe use different terminologies to denote the same concept. The consistency in concept labels is to unify the terms as concept labels. We access the WordNet[9] to unify the terms. WordNet is an on-line lexical reference system developed at Princeton University. It consists of synonym sets called synsets, and each synset represents a single distinct concept. For example, in figure 1 the term "staff" in O_1 and the term "faculty" in O_2 denote the same concept "academic people". In the WordNet, they have the same synset "staff#2". So, they can be represented using the same string "staff#2". But for other concepts, such as "Course", they use the same concept labels, and correspond to eight synsets. So, the concept "Course" is represented by the union of the eight synset labels, "course#1 ∪ course#2 ∪ … ∪ course#8" in logical form.

The consistency in relation labels means unifying the terms as relation labels. The relation is determined by its range and domain. So we can determine that two relations are consistent, if they have the same range and domain, or the range and the domain have subsumption relation respectively. For example, in figure 1, the relation "instructBy" in O_1 and the relation "teachBy" in O_2 have the same domain defined as concept "Course". But they have range defined as concept "Faculty" and "Staff" respectively. Through the process of unifying for concept labels, "Staff" and "Faculty" correspond to the same label "staff#2". So we can determine the relation "teachBy" is consistent to the relation "instructBy", and they can be represented by the same label.

3.4 Relationships between atomic concept labels

Each term as label has a meaning independently from the ontology where it occurs. In order to do inference between complex concepts, the relationships between atomic concept labels in different complex concepts must be discovered. They are transformed to the axioms as the premises of inference. In this paper, we access WordNet to get relationships between terms. WordNet organizes terms based on the semantic relations of them. So these relations are just origin of our axioms generating. The relations of terms in the WordNet corresponding to subsumption axioms are shown in table 1.

Table 1. WordNet relations vs. subsumption axioms

Relations in WordNet	Subsumption Axioms
$term_1$ meronym $term_2$	$term_1 \subseteq term_2$
$term_1$ holonym $term_2$	$term_2 \subseteq term_1$
$term_1$ hyponym $term_2$	$term_1 \subseteq term_2$
$term_1$ hypernym $term_2$	$term_2 \subseteq term_1$

For example, in figure 1, the concept "Professor" correspond to the synset "professor#1" and the concept "Assistant_Professor" correspond to the synset "assistant professor#1" of WordNet. They have the hypernym relation, i.e "professor#1" hypernym of "assistant professor#1". So there is an axiom for "Professor" and "Assistant_Professor", which is Assistant_Professor ⊆ Professor described with logic formula.

3.5 The process of ontology mapping

The process of ontology mapping is in four steps. We describe these steps with example of ontologies in figure 1.

Step 1: transforming the concepts of ontologies into complex concepts of DL using the labels of concepts and relations, and restrictions of ontologies. In the ontology O_1, the complex concept $Complex\_Concept(Data\_Mining)_1$ of concept "Data Mining" is:

$$Complex\_Concept(Data\_Mining)_1 = Data\_Mining \cap AI \cap Course \cap Computer\_Science \cap \forall instructBy.\ Assistant\_Professor \quad (1)$$

And in ontology O_2, the complex concept $Complex\_Concept(Data\_Mining)_2$ of concept "Data_Mining" is:

$$Complex\_Concept(Data\_Mining)_2 = Data\_Mining \cap Course \cap Computer\_Science \cap \forall teachBy.\ Professor \quad (2)$$

Through this process, the two input ontologies are transformed into complex concept sets.

Step 2: unifying the representation form of concept labels and relation labels in complex concepts by accessing WordNet. We use the methods described in senction 3.3 to unify the representation form of labels. For example, we define the unified form for relations "teachBy" and "instructBy" as "teachBy".

So, the formulas (1) and (2) are transformed to the formulas (3) and (4) which are represented by consistent labels:

$$
\begin{aligned}
&Complex\_Concept(Data\_Mining)_1 \\
&\quad = Data\_Mining\#1 \cap AI\#1 \\
&\quad\quad \cap\ (course\#1 \cup course\#2 \cup \ldots \cup course\#8) \\
&\quad\quad \cap\ Computer\_Science\#1 \\
&\quad\quad \cap \forall\, teachBy.Assistant\_Professor\#1 \qquad (3)
\end{aligned}
$$

$$
\begin{aligned}
&Complex\_Concept(Data\_Mining)_2 \\
&\quad = Data\_Mining\#1 \\
&\quad\quad \cap\ (course\#1 \cup course\#2 \cup \ldots \cup course\#8) \\
&\quad\quad \cap\ Computer\_Science\#1 \\
&\quad\quad \cap \forall\, teachBy.\ Professor\#1 \qquad (4)
\end{aligned}
$$

Step 3: get the subsumption axioms among the concept labels. This process is described in section 3.4. For the formulas (3) and (4), we can get subsumption axioms from WordNet showed as follows:

$$
\begin{gathered}
Assistant\_Professor\#1 \sqsubseteq Professor\#1 \\
AI\#1 \sqsubseteq Computer\_Science\#1 \qquad (5)
\end{gathered}
$$

Step 4: discover the semantic relationships between complex concepts through inference of DL. With the subsumption axioms among the concept labels generated in the step 3 as the premises for reasoning, we can use tableau algorithm[10] of DL to reason the semantic relationships between complex concepts.

The basic idea of tableau algorithm is to try to prove the satisfiability of a concept C by exhaustively appling tableau rules that decompose the syntactic structure of the concepts to construct a so-called completion tree, a tree where each node x of the tree is labeled with a concept set $L(x) \subseteq sub(C)$ and each edge $<x, y>$ is labeled by $L(<x, y>)=R$ for some role R occurring in $sub(C)$, where the $sub(C)$ is the set of sub-concepts of C. The tree is called complete when for some node x, $L(x)$ contains a clash, or when none of rules is applicable. For a node x, $L(x)$ is said to contain a clash if, for some concept D, $\{D, \neg D\} \subseteq L(x)$. For a input concept C, if the expansion rules can be applied in such a way that they yield a complete, clash-free completion tree, then the algorithm returns "C is satisfiable", and "C is unsatisfiable" otherwise.

So for two concepts C and D, checking the subsumption relation $C \sqsubseteq D$ corresponds to test the (un)satisfiability of the concept $C \cap \neg D$, that is $C \sqsubseteq D \Leftrightarrow C \cap \neg D$ is unsatisfiable. The identify of semantic relations between concepts of different ontologies presented in section 3.1 can be transformed to test unsatisfiability of the concept, shown in the following

- ♦ $C_1 \sqsubseteq C_2 \Leftrightarrow C_1 \cap \neg C_2$ is unsatisfiable.
- ♦ $C_1 \sqsupseteq C_2 \Leftrightarrow C_2 \cap \neg C_1$ is unsatisfiable.
- ♦ $C_1 \equiv C_2 \Leftrightarrow C_1 \cap \neg C_2$ is unsatisfiable and $C_2 \cap \neg C_1$ is unsatisfiable.
- ♦ $C_1 \perp C_2 \Leftrightarrow C_1 \cap C_2$ is unsatisfiable.

For example, we can use tableau algorithm for *ALC*[11][12] to check semantic relations between two complex concepts showed in formulas (3) and (4). The expansion rules for *ALC* are showed in figure 2.

$\cap$-rule: if (1) $(C1 \cap C2) \in L(x)$
(2) $\{C1,C2\} \not\subseteq L(x)$
then $L(x) \rightarrow L(x) \cup \{C1,C2\}$
$\cup$-rule: if (1) $(C1 \cup C2) \in L(x)$
(2) $\{C1,C2\} \cap L(x) = \perp$
then $L(x) \rightarrow L(x) \cup \{C\}$ for some $C \in \{C1,C2\}$
$\exists$-rule: if (1) $\exists S.C \in L(x)$
(2) x has no S-neighbor y with $C \in L(y)$
then create a new node y with $L(<x, y>)=S$
and $L(y)=\{C\}$
$\forall$-rule: if (1) $\forall S.C \in L(x)$
(2) there is an S-neighbor y of x with $C \not\subseteq L(y)$,
then $L(y) \rightarrow L(y) \cup \{C\}$

Figure 2. Tableau expansion rules for *ALC*

To be simple we do following substitutions. Let:
C_0= $Complex\_Concept\ (Data\_Mining)_1$ and
C'_0= $Complex\_Concept\ (Data\_Mining)_2$
denote the complex concepts, and
C_1=$Computer\_Science\#1$,C_2=$Data\_Mining\#1$,
C_3=$AI\#1$, C_4=$Professor\#1$,
C_5=$Assistant\_Professor\#1$,
$(D1, D2, \ldots, D8) = (Course\#1, Course\#2, \ldots, Course\#8)$,
$R=teachBy$
denote atomic concepts and atomic relation in figure 1. So, the formula (3), (4) and (5) are transformed abstract formulas (7), (8) and (6).

$$
\begin{gathered}
C_5 \sqsubseteq C_4 \\
C_3 \sqsubseteq C_1 \qquad (6)
\end{gathered}
$$

$$C_0 = C_1 \cap (D_1 \cup D_2 \cup \ldots \cup D_8) \cap C_2 \cap C_3 \cap \forall R\,.C_5 \qquad (7)$$

$$C'_0 = C_1 \cap C_2 \cap (D_1 \cup D_2 \cup \ldots \cup D_8) \cap \forall R\,.C_4 \qquad (8)$$

Using the formula (6) as premise, we are going to test if $C_0 \sqsubseteq C'_0 \Leftrightarrow C_0 \cap \neg C'_0 \sqsubseteq \perp$.

$$C_0 \cap \neg C'_0 = (C_1 \cap\ (D_1 \cup D_2 \cup \ldots \cup D_8)\ \cap C_2 \cap C_3 \cap \forall R\,.C_5) \cap \neg(C_1 \cap C_2 \cap\ (D_1 \cup D_2 \cup \ldots \cup D_8) \cap \forall R\,.C_4)$$

Applying the De Morgan's law to transform the concepts to negation normal form:

$$\neg C'_0 = \neg C_1 \cup \neg C_2 \cup (\neg D_1 \cap \neg D_2 \cap \ldots \cap \neg D_8) \cup \exists R.\ \neg C_4$$

$$C_0 \cap \neg C'_0 = (C_1 \cap (D_1 \cup D_2 \cup \ldots \cup D_8) \cap C_2 \cap C_3 \cap \forall R\,.C_5) \cap (\neg C_1 \cup \neg C_2 \cup (\neg D_1 \cap \neg D_2 \cap \ldots \cap \neg D_8) \cup \exists R.\neg C_4)$$

The tableau algorithm initializes a tree T to contain a node x_0, concept set $L(x_0) = \{C_0 \cap \neg C'_0\}$, called the root node. We apply $\cap$-rule and $\cup$-rule to extend

$$L(x_0) = \{C_1, C_2, C_3, \forall R\,.C_5, \exists R.\neg C_4\}.$$

Here, we remove other cases for $L(x_0)$ that include apparently a clash, such as $L(x_0)$ including $\{C_1, \neg C_1\}$. From $\exists$-rule and $\forall$-rule, we extend $L(x_0)$, and create a new node x_1, which is a neighbor of x_0 for relation R, with

$$L(x_1) = \{C_1, C_2, C_3, C_5, \neg C_4\}.$$

According the axioms $C_5 \subseteq C_4 \Leftrightarrow C_5 \cap \neg C_4 \subseteq \bot$, so $L(x_1)$ includes a clash: $\{C_5, \neg C_4\}$. Thus $C_0 \cap \neg C'_0$ is unsatisfiable and $C_0 \subseteq C'_0$ is valid.

Also, we test if $C'_0 \subseteq C_0$ by the same process above-mentioned, we can find that $C'_0 \cap \neg C_0$ is satisfiable and $C'_0 \subseteq C_0$ is not valid. So a subsumption relationship was detected between C_0 and C'_0, which means that the meaning of “Data Mining” in Ontology O_1 is subsumed by the meaning of “Data Mining” in ontology O_2. So, the is a semantic subsumption relationship($\supseteq$) between the concept “Data Mining” in Ontology O_2 and the concept “Data Mining” in Ontology O_1.

Notably, when we find the semantic relationships between concepts, we always find the closest semantic relationship they held, i.e. we always firstly find the equivalent relationship between concepts, then other relationships.

4 Experiments

4.1 Experiment Setup

We use standard information retrieval metrics, recall and precision, to evaluate our method and to compare with algorithm of CTXMATCH[2].

Recall R: it describes the number of correct concept mappings discovered in comparison to the total number of existing mappings.

$$R = \frac{|correct\_found\_mappings|}{|existing\_mappings|}$$

Precision P: it measures the number of correct concept mappings discovered versus the total number of mappings discovered.

$$P = \frac{|correct\_found\_mappings|}{|found\_mappings|}$$

Data set: we evaluated our approach on three data sets: course catalog, conference ontologies and computer science department ontologies. The course catalog describes course at Cornell University and the University of Washington[13]. The conference ontologies are from the OAEI 2007(Ontology Alignment Evaluation Initiative 2007)[14]. They are dealing with the conference organization, and two ontologies are selected in my experiment: Cmt and Pcs. The CS department ontology describes the courses, people and other things about the department of CS. The characteristics of ontologies used in this paper are shown in table 2.

Table 2. The characteristics of ontologies

Ontologies		Concepts	non-taxonomical relations	Max Depth
Course catalog	Cornell	34	0	4
	Washington	39	0	4
CS department	CS_A	38	8	4
	CS_B	40	10	5
Conference	Cmt	38	49	5
	Pcs	23	24	5

Processing of concept labels: when a label contains two or more words we do process as follows: firstly, we access the WordNet to extract the single expressions (multiword); secondly, if “and” is contained by the label, it is replaced by logic conjunction “∩”; if “or” is contained by the label, it is replaced by logic disjunction “∪”; thirdly, if “of” is contained by the label, we remove the word “of” and replace it with logic conjunction “∩”. For example, the concept label “College_of_Arts_and_Sciences”, its logic form transformed is “(College ∩ Arts) ∩ Sciences”.

We conducted an experimental study to compare the performance of our approach with CTXMATCH. We have implemented the approach presented in this paper in an open source OWL-DL reasoner, Pellet[15]. Pellet supports the tableau algorithm of OWL-DL[16], i.e. description logic SHOIN(D). For the relationships between concepts of different ontologies, we firstly check the equivalent relationship, and then check the subsumption and subsumed relationships, finally check the disjoint relationship. The order is: “≡” > “⊇” > “⊆” > “⊥”.

4.2 Experiment Results

Figure 3 shows the comparison between CTXMATCH and our approach about precision and recall of mapping between concepts of different ontologies. Because the course catalog only has the taxonomical relation between concepts (it is like a hierarchical classification), so the CTXMATCH and our approach have the same results.

For the conference ontologies and the CS department ontologies, concepts have different non-taxonomical relations with other concepts in different ontologies. By these relations the intended meaning of concepts are specified. It is not enough to identify the meanings of concepts only considering taxonomical relation. So our approach outperforms the CTXMATCH.

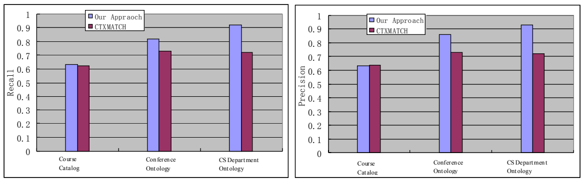

Figure 3. The comparison between our approach and CTXMATCH about recall and precision

We evaluate the precision and recall of mapping between concepts for relationships "≡", "⊆", "⊇" and "⊥" between our approach and CTXMATCH over CS department ontologies respectively. The results compared are shown in figure 4.

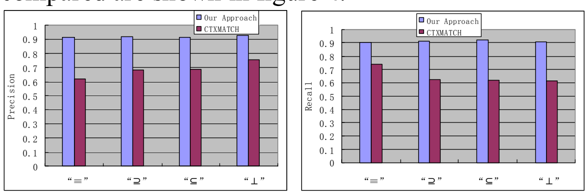

Figure 4. The results of mapping for different logic relationships about recall and precision

5. Conclusions

Ontology mapping is one of the main challenges for semantic web. In this paper, we introduce inference of description logic to detect the semantic relationships between concepts of different ontologies. We enrich the CTXMATCH algorithm through considering the non-taxonomical relations between concepts, which specifies the intended meaning of concepts. Our approach can be applied to complex ontologies, and implement ontology mapping based on the intended meaning of concepts. Experiments show that our approach improves the precision and recall of ontology mapping for complex ontologies compared with CTXMATCH.

6. Acknowledgements

This paper is sponsored by NSF of China (90612001), Education Ministry Doctoral Research Foundation of China(20070055054), Science and Technology Development Plan of Tianjin, (08JCYBJC13000).

7. References

[1] Gruber, T. A., A translation approach to portable ontology specifications, Knowledge Acquisition, 1993, 5(2)

[2] Jie Tang, Bang-Yong Liang and Juan-Zi L, "Toward Detecting Mapping Strategies for Ontology Interoperability" The 14th Int'l Conf. on World Wide Web (WWW2005), Makuhari Messe, Chiba, Japan, Tuesday May 10, 2005.

[3] Paolo Bouquet, Luciano Serafini, Stefano Zanobini, "Semantic Coordination: A New Approach and an Application", ISWC 2003, LNCS 2870, pp.130-145, 2003.

[4] Kalfoglou, Y., and M. Schorlemmer, "Ontology mapping: the state of the art", The Knowledge Engineering Review, 18(2003) (1) 1–31.

[5] P. Shvaiko, J. Euzenat, "A Survey of Schema-based Matching Approaches", Journal on Data Semantics, 2005.

[6] Namyoun Choi, Il-Yeol Song, Hyoil Han, "A survey on ontology mapping", SIGMOD Record 35(3): 34-41 (2006).

[7] Rahm E., Bernstein P. A., "A survey of approaches to automatic schema matching:, The VLDB Journal, 2001, 10 (4):334～350

[8] Baader, F., A. Borgida, R. J. Brachman et al, "The Description Logic Handbook, Theory, Implementation and Applications", Baader, F., Calvanese D., McGuinness D., Nardi D. and Patel-Schneider P. (Eds), Cambridge University Press, 2003.

[9] Miller,G. A. et al, "Introduction to WordNet: An On-line Lexical Database", in the attached specification of WordNet 1.6, 1993.

[10] Baader F, Sattler U., "An Overview of Tableau algorithms for description logics", Studia Logica, 2001, 69:5～40.

[11] Manfred Schmidt-Schauβ, "Attributive concept descriptions with complements", Artificial Intelligence, 1991, 48(1): 1～26

[12] Franz Baader, et al, „Tableau algorithms for description logics", In R. Dyckhoff editor, Proc. Tableaux 2000. Berlin: Springer, 2000.

[13] http://pages.cs.wisc.edu/~anhai/wisc-si-archive/.

[14] http://oaei.ontologymatching.org/2007/

[15] Bijan Parsia and Evren Sirin, "Pellet: An OWL DL reasoner". In the Third International Semantic Web Conference – Poster, 2004.

[16] Dean, M., G. Schreiber (Eds), "OWL Web Ontology Language Reference, W3C Recommendation" 2004. URL: http://www.w3.org/TR/2004/REC-owl-ref 20040210/.