Hybrid Ontology Matching for Solving the Heterogeneous Problem of the IoT

Zhichun Wang
College of Information Science and Technology
Beijing Normal University
Beijing, 100875, China
zcwang@bnu.edu.cn

Rongfang Bie, Mingquan Zhou
College of Information Science and Technology
Beijing Normal University
Beijing, 100875, China
{rfbie, mqzhou}@bnu.edu.cn

***Abstract*—The vision of Internet of Things is to connect everyday objects through embedding wireless devices, so that they can interact with each other and provide new services. One of major challenges for the IoT is the heterogeneous problem. Information generated by different IoT objects will not be compatible, which hinders data communications between the IoT objects. In this paper, we study semantic technology for integrating heterogeneous information in the IoT. We propose to use ontologies to model the schema of data generated by the IoT objects, and propose a hybrid ontology matching approach to solve the heterogeneous problem. The proposed approach uses multiple matchers to compute similarities between ontology elements, and computes weight for each matcher by making use of the hierarchical structure of ontology. Experimental results show that our method can effectively filter out wrong mappings and obtain alignments with high quality between ontologies.**

Keywords-Interner of Things, Heterogeneity, Semantic Web, Ontology Matching

I. INTRODUCTION

The term Internet of Things (IoT) was firstly introduced by Kevin Ashtom in1999 [1], which describes a new paradigm consisting of uniquely identifiable objects and their Internet-like linking structure. Equipped with Radio-Frequency Identification (RFID) or sensors, various smart things are able to interact with each other and corporately reach common goals [2]. The realization of IoT will tremendously change our daily life and bring tangible business benefits. However, many challenging issues still need to be solved before the IoT idea being successfully achieved. One of the most important issues is how to represent, interconnect, and organize information generated by the IoT. It is believed that the future IoT will contain trillions of objects; the data generated by the IoT will be heterogeneous which hinders effective communications between the IoT objects. Several Semantic based solutions were proposed to solve the interoperability problem of IoT. Toma et al. [3] presents a roadmap for semantic technologies in the context of the IoT. They argue that Resource Description Framework (RDF) is an elegant solution for interconnect and organize information of IoT. Katasonov et al. [4] argue that heterogeneity of the IoT components, standards and data formats, creates significant obstacles for interoperability in the IoT. They present a vision of a semantic middleware for the IoT, which can manage distributed and heterogeneous components of different nature. Song et al. [5] also present a semantic framework that automates the IoT interoperability without any changes of existing standards, devices or technologies.

In this paper, we study semantic technology for integrating heterogeneous information in the IoT. We propose to use ontologies to model the schema of data generated by the IoT objects, and propose a hybrid ontology matching approach to solve the heterogeneous problem. Ontology is the backbone of Semantic Web technologies; it provides a vocabulary describing a domain of interest [6]. As the number of IoT objects will be very high, a large amount of ontologies have to be created for various devices in the IoT. Ontology heterogeneity is an inevitable problem in the context of IoT. Ontology matching is a key solution to the semantic heterogeneity problem. It finds correspondences between semantically related entities of ontologies. The results of ontology matching are called alignments, can be used for various tasks, such as ontology merging, data integration, and data translation [7].

Currently, in the domain of Semantic Web, much effort has been made for automatic ontology matching and many matchers have been proposed. Existing techniques are mostly based on calculating similarities between entities of two ontologies based on various types of ontology information [8]. For example, Stoilos et al. [9] proposed name based matcher which compares entities' labels using string-distance metrics; Doan's matcher GLUE [10], utilized the information of ontology instances to predicate possible alignments; Qu et al. [11] integrated all kinds of features to form a virtual document for each entity, and calculated similarities using information retrieval technique. Although a significant number of matchers have been developed, their performance varies over different matching problems. One single matching strategy cannot obtain high performance in many real applications. Therefore, several ontology matching systems integrate multiple matchers in one framework, such as RiMOM [8], Falcon-AO [12] and PRIOR+ [13]. These systems achieved high performance in the campaigns of OAEI. However, in most existing systems, matchers were predefined and fixed; and the combination of different strategies was viewed as an engineering issue and dealt with by manual. This is obviously

978-0-7695-4745-9/12 $26.00 © 2012 IEEE
DOI 10.1109/TrustCom.2012.177

insufficient, in particular when dealing with a large number of matching tasks in the context of IoT.

In order to overcome the limitations of existing methods for ontology matching in the IoT, we propose a hybrid ontology matching approach. First, several effective ontology matching strategies are defined, and then we propose a self-adaptive aggregation method to combine the results of different strategies. Contributions of this work include:

- We propose a new measure called *consistency* to evaluate the reliability of matchers' predictions. The *consistency* is computed by making use of hierarchical structure of ontology, which is accurate and need no prior knowledge of the true alignment.
- We proposed a new similarity aggregation approach based on the *consistency* measure; instead of assigning global weights to matchers, the new approach calculates the consistency of every similarity value and takes it as the weight of that similarity in the aggregation.
- We test the proposed approach on eight ontology matching tasks; the results validate the effectiveness of our approach.

The rest of this paper is organized as follows: Section 2 presents the foundational knowledge of our work. Section 3 describes our new similarity aggregation approach. Section 4 analyses the experimental results. Section 5 covers some related work. Finally, the conclusion and future work is discussed in section 6.

II. PRELIMINARIES

A. Ontology and Ontology Matching

An ontology is a formal, explicit specification of a shared conceptualization [14]. An ontology O can be formally defined as a 6-tuple:

$$O = \{C, P, H^C, H^P, A^O, I\},$$

where C and P are the sets of concepts and properties, respectively. Concepts C in O are arranged in a subsumption hierarchy H^C, while properties P are arranged in a hierarchy H^P. A^O represents a set of axioms, which are used to constrain values for classes and instances, or to infer knowledge from already existing knowledge; I is a set of instances of concepts and properties. Several standard languages, such as the Web Ontology Language (OWL) and Resource Description Framework Schema (RDFS), can describe ontologies.

Ontology matching is the process of finding correspondence between semantically related entities between different ontologies. It can be defined as follows [6]: Taking two ontologies O_1 and O_2 as input, establish semantic correspondences between entities of two ontologies. The established correspondences (or mappings) can be represented by 4-tuples:

$$\{e_{i1}, e_{i2}, c, r\},$$

where e_{i1} is a entity in O_1, e_{i2} is a entity in O_2; c is the confidence of the correspondences; r specifies the type of the correspondences, which can be *equivalence*, *disjointness* and *more general*.

B. Matcher combination

There are typically two ways to combine multiple matchers [11]: sequential combination of matchers and parallel combination of matchers. In the sequential combination, multiple matchers run sequentially, the output of one matcher is taken as input of the following matchers; this combination strategy is more classically used to refine an alignment. In the parallel combination, several matchers run independently and their results are combined to get final alignments; the parallel approach is more flexible which allows free selection and combination of matchers. Our proposed approach follows the parallel combination strategy, and its matching process is shown in Fig. 1.

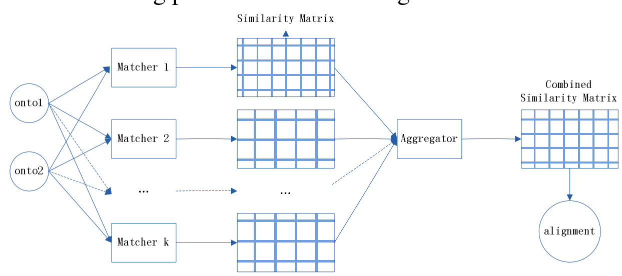

Fig. 1. The matching process of our approach.

Given two ontologies O_1 and O_2, k matchers run independently and each determines an intermediate match result consisting of a similarity value between 0 and 1 for every entity pair between O_1 and O_2. Let m entities in O_1 and n entities in O_2, the intermediate result of one matcher is a $m \times n$ similarity matrix. Therefore, after the execution of k matchers, k similarity matrixes are obtained. The following step is to combine the results by aggregating all similarity matrixes into a combined one. This step is called similarity aggregation, it is very important because different aggregation methods can get very different results. In the last step, the alignment of O_1 and O_2 is extracted from the combined similarity matrix.

III. CONSISTENCY BASED SIMILARITY AGGREGATION

The process of similarity aggregation takes multiple similarity matrixes as input and generates a combined similarity matrixes as outputs. Ideally, the similarity values of correct mappings are supposed to get high weights and those of wrong mappings are supposed to get low weights. However, we cannot decide whether a mapping is correct or not without a reference alignment. Here, we have observed that the hierarchical structure of ontologies can help us find some bad mappings. Recall the formal representation of an ontology defined in Section 2, an ontology is denoted as a 6-tuple $O = \{C, P, H^C, H^P, A^O, I\}$. Here H^C and H^P are the hierarchical relations of concept and property respectively. The hierarchy is the backbone of an ontology, which poses a basic constraint on ontology matching: alignment of two ontologies should not change the hierarchical relations of original entities. Based on this constraint, we define a measure called consistency to evaluate the reliability of similarity values assigned by matchers. This measure is then used as weighting factor in the similarity aggregation process. Since concepts and properties have different characteristics, their mappings' consistency is computed in different way, which is presented in the following subsections.

A. *Consistency of concept mappings*

Concept hierarchy classifies concepts into categories thus facilitating its search, reuse and understanding[15]. Fig. 2 shows an example of the concept hierarchy of an ontology. Here, concepts are arranged in a partial order by *is-a* relation. For example, Student is a subconcept of Person, while it is also the superconcept of Graduate, PHDStudent and UnderGraduate.

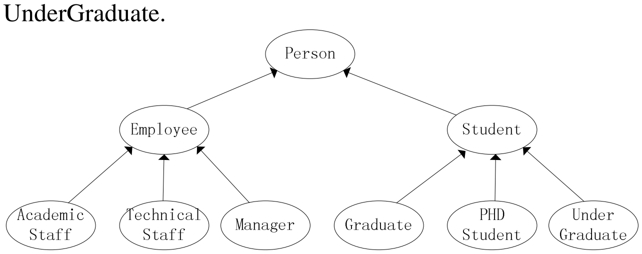

Fig. 2. An example of concept hierarchy

Here, we define the consistency for similarity values between concepts by checking how the mappings preserve hierarchical relation. Suppose there is a matching task as shown in Fig. 3, three green lines denote three highly possible mappings because these concept pairs all have high similarity values. Then a mapping such as $<A,d>$ tend to be wrong because it violates the hierarchy established by the three mappings, therefore a low consistency will be assigned to $<A,d>$.

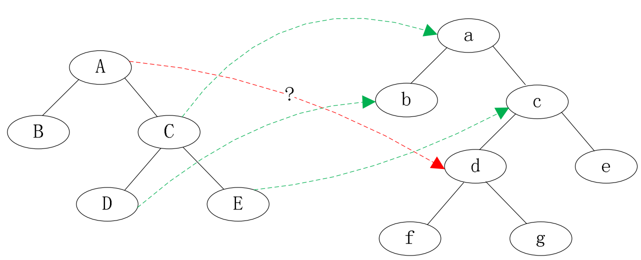

Fig. 3. An example of conflict among concept mapping

Let $Sup(c)$ is a set of superconcepts of concept c; $Sub(c)$ is a set of subconcepts of concept c. For each concept mapping $<c_1,c_2>$ where c_1 is from O_1 and c_2 is from O_2, we define $conflict(c_1,c_2)$ as a set of concept mapping that are conflict with $<c_1,c_2>$ in terms of ontology hierarchy:

$$conflict(c_1,c_2)=\{<c_1^{'},c_2^{'}>| c_1^{'}\in Sup(c_1)\wedge c_2^{'}\in Sub(c_2)\}\\ \cup\{<c_1^{'},c_2^{'}>| c_1^{'}\in Sub(c_1)\wedge c_2^{'}\in Sup(c_2)\}$$

Then the consistency of similarity of concepts $<c_1,c_2>$ is defined as:

$$consistency_c(c_1,c_2)=1-\frac{\sum_{<c_1^{'},c_2^{'}>\in conflict(c_1,c_2)}M(c_1^{'},c_2^{'})}{|conflict(c_1,c_2)|}$$

where $M(c_1^{'},c_2^{'})$ is the similarity value of $<c_1^{'},c_2^{'}>$. According to the definition, for a given mapping, if its conflict mappings have low average similarities, it will get high consistency; otherwise, it will get low consistency. Therefore, consistency can be good indicator of the confidence of mappings.

B. *Consistency of properties' similarity*

There is usually less hierarchical information of properties in ontologies. However, properties are closed related to concepts because they specify certain features of concepts or describe relations between concepts. Therefore, we define the consistency of properties by utilizing the hierarchical structure of concepts. Suppose a property mapping $<p_1,p_2>$ as shown in Fig. 3, the domain of p_1 is A and the domain of p_2 is c. Because the mapping $<A,c>$ has low consistency, mapping $<p_1,p_2>$ is probably a wrong mapping and thus should get low consistency. Based on this idea, we define the consistency of property mappings as follows:

Let $dom(p)$ denotes the domain of property p, the consistency of property p is:

$$consistency_p(p_1,p_2)=\frac{\sum_{<c_1,c_2>\in dom(p_1)\times dom(p_2)}consistency_c(c_1,c_2)}{|dom(p_1)\times dom(p_2)|}$$

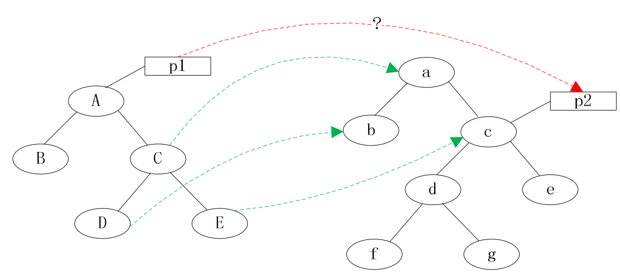

Fig. 4. An example of conflict between property mappings.

C. *Similarity aggregation based on consistency*

According to the definition of consistency, mappings with high consistency values tend to be more reliable while mappings with low consistency values are possible wrong. Therefore, consistency is used to weight similarity values in our similarity aggregation approach. Let $M_1,...,M_k$ be the similarity matrixes returned by k matcher, the similarity aggregation process is as follows:

1) For each similarity matrix M_i, compute the consistency of each similarity values in M_i. The results of this step is a set of consistency matrixes $C_1,...,C_k$, the elements in these consistency matrixes are consistency values of the corresponding similarity.

2) For each mapping $<e_i,e_j>$, compute its combined similarity value by

$$M_{combine}(e_i,e_j)=\frac{\sum_{q=1,..,k} M_q(e_i,e_j)\cdot C_q(e_i,e_j)}{\sum_{q=1,..,k} C_q(e_i,e_j)}$$

IV. EXPERIMENTS

In this section, we present experimental analysis of our aggregation method.

A. Datasets

For our experiments, eight ontology matching tasks are tested. Four of these tasks are from OAEI 2009 benchmark, and the other four tasks are from I^{3}CON. Each ontology matching task contains two real-world ontologies, and also provides a reference alignment. Ontology matching tasks from OAEI are #301, #302, #303 and #304. In these tasks, ontologies are in the domain of bibliography; four ontologies designed by different organizations are to be matched with one global ontology. Tasks from I^{3}CON include Russia, Animals, Tourism and People-pet. Table 1 lists the number of concepts and properties and the size of reference alignments of these matching tasks.

Table 1. Information of ontology matching tasks.

Task		#concepts	#properties	#refaligns
#301	O_1	15	40	58
	O_2	37	72	
#302	O_1	16	31	47
	O_2	37	72	
#303	O_1	56	72	48
	O_2	37	72	
#304	O_1	41	51	76
	O_2	37	72	
Russia	O_1	151	76	285
	O_2	103	58	
Animals	O_1	10	15	24
	O_2	10	14	
Tourism	O_1	340	97	226
	O_2	474	100	
People-Pet	O_1	60	15	93
	O_2	58	15	

B. Metrics

We use precision, recall and F1-measure to evaluate the matching results.

Precision: It is the percentage of the correct discovered alignments in all discovered alignments;

$$p=\frac{\#correct\_found\_alignments}{\#all\_found\_alignments}$$

Recall: It is the percentage of the correct discovered alignments in all correct alignments;

$$r=\frac{\#correct\_found\_alignments}{\#all\_correct\_alignments}$$

F1-Measure: combines both precision and recall;

$$f=\frac{2pr}{p+r}$$

C. Matchers

To test our aggregation approach, we use the following four matchers. We select these matchers because they are widely used in ontology matching problem. Since our aggregation approach is independent of matchers, it can also works with other matchers.

Substring-Label Matcher: This matcher compares the labels of two entities, and returns a similarity value based on the common substring of two labels.

$$S_{sub\_l}(e_1,e_2)=\frac{2|t|}{|label(e_1)|+|label(e_2)|}$$

where t is the longest common substring of $label(e_1)$ and $label(e_2)$.

Edit-Distance-Label Matcher: Edit distance between two strings is the minimal cost of operations (insertion, replacement and deletion of characters) to be applied to one the strings in order to obtain the other one. This matcher calculates the similarity between two entities based on the edit distance of their labels. It is defined as

$$S_{edit\_l}(e,e^{'})=1-\frac{|\{ops\}|}{\max(|label(e)|,|label(e^{'})|)}$$

where $|\{ops\}|$ denotes the number of operations, $|label(e)|$ and $|label(e^{'})|$ are the length of two labels.

Edit-Distance-Comment Matcher: Comments usually consist of a phrase or sentence in natural language; this matcher is a variation of Edit-Distance-Label matcher but applied to tokens. The operations are applied to tokens instead of characters. This matcher was used in [11].

$$S_{edit\_c}(e,e^{'})=1-\frac{|\{ops\}|}{\max(|comment(e)|,|comment(e^{'})|)}$$

where $|\{ops\}|$ denotes the number of operations, $|comment(e)|$ and $|comment(e^{'})|$ are the number of tokens in the comments of two entities.

Vector-Space Matcher: Vector-Space matcher employs the vector space technique in information retrieval to calculate the similarity between ontology entities. It constructs a virtual document for every ontology entity, which contains the context information of the entity. The document consists of words from the entity's description (e.g. name, comments) and descriptions of all related entities (e.g. instances, superclasses, subclasses). The similarity between ontology entities is calculated by computing similarity between virtual documents in the Vector Space Model (VSM). In VSM each document is represented by a weighted feature vector, the weights are typically computed by TFIDF method. For a word i in document j, TFIDF defines the weight of word as

$$\omega_{ij} = tf_{i,j} \times \log(\frac{N}{df_i})$$

where $tf_{i,j}$ is the number of occurrences of i in j, df_i is the number of documents containing i, N is the total number of documents. The similarity of documents is computed by the cosine value between their feature vectors:

$$S_{vs}(e,e^{'}) = \cos(D_e, D_{e^{'}}) = \frac{\sum_{k=1}^{N} \omega_k \cdot \omega_k^{'}}{\sqrt{\sum_{k=1}^{N} \omega_k^2} \cdot \sqrt{\sum_{k=1}^{N} \omega_k^{'2}}}$$

D. Results

In this subsection, we present the results of the experiment. For each matcher, we use the Naïve Descendant Extraction [17] algorithm to get the final alignment from similarity matrix. We also set thresholds to filter out mappings with low similarity values; multiple thresholds are used for one execution of matching algorithm and the best results are kept as the final results.

In the first experiment, we test the performance of four individual matchers. Table 2, 3, 4 show the precision, recall and f-measure of individual matchers respectively. Here we use *Edit-label*, *Sub-label*, *Edit-comment* and *VSM* to denote the four matchers described in Section 4. According to the results of individual matchers, we find that each matcher has different performance in different matching tasks. Taking sub-label matcher as an example, it gets the best f-measures in #301-#304 matching tasks, but it has poor performance in Russia and Animals tasks. *Sub-label* matcher gets the best precision on average, but *VSM* matcher gets the best recall. According to f-measure, *VSM* performs best, and then *Sub-label* and *Edit-label*. *Edit-comment* gets the lowest f-measure on average; it returns no results in 5 tasks because there is no comments information available in these tasks.

In the second experiment, we test the performance of our proposed similarity aggregation method. In each task, four matchers run independently and return four similarity matrixes. Then these similarity matrixes are combined by similarity aggregation method, the final matching result is extracted from the combined similarity matrix.

We compare the results of our aggregation method with the best results of individual matchers and the following four other aggregation methods:

MAX: return the maximum similarity value of any matcher;

MIN: return the minimum similarity value of any matcher;

AVG: return the average similarity over all matchers;

SIGMOID: return the average of similarity values transformed by a sigmoid function.

Table 2. Precision of individual matchers.

Dataset	Edit-label	Sub-label	Edit-comment	VSM
301	77.78%	90.00%	90.91%	79.25%
302	38.10%	87.50%	0.00%	78.79%
303	92.31%	92.50%	0.00%	76.74%
304	94.52%	95.89%	95.08%	92.41%
Russia	96.05%	80.95%	0.00%	99.37%
Animals	100.00%	100.00%	0.00%	100.00%
Tourism	92.79%	88.29%	0.00%	91.79%
Peo-pet	95.59%	98.53%	90.91%	98.55%
Avg	85.89%	91.71%	34.61%	89.61%

Table 3. Recall of individual matchers.

Dataset	Edit-label	Sub-label	Edit-comment	VSM
301	72.41%	77.59%	68.97%	72.41%
302	34.04%	59.57%	0.00%	55.32%
303	75.00%	77.08%	0.00%	68.75%
304	90.79%	92.11%	76.32%	96.05%
Russia	25.61%	17.89%	0.00%	55.09%
Animals	8.33%	8.33%	0.00%	83.33%
Tourism	85.40%	86.73%	0.00%	79.20%
Peo-pet	69.89%	72.04%	10.75%	73.12%
Avg	57.69%	61.42%	19.50%	72.91%

Table 4. F-measure of individual matchers.

Dataset	Edit-label	Sub-label	Edit-comment	VSM
301	75.00%	83.33%	78.43%	75.68%
302	35.96%	70.89%	0.00%	65.00%
303	82.76%	84.09%	0.00%	72.53%
304	92.62%	93.96%	84.67%	94.19%
Russia	40.44%	29.31%	0.00%	70.88%
Animals	15.38%	15.38%	0.00%	90.91%
Tourism	88.94%	87.50%	0.00%	85.04%
Peo-pet	80.75%	83.23%	19.23%	83.95%
Avg	63.98%	68.46%	22.79%	79.77%

The results of aggregation methods are compared in Table 5. We can find that combination of matchers makes limited improvements of the results compared with the best results of individual matchers in each task. However, since we don't know which matcher performs best without a reference alignment, the composed strategy is still a good choice to solve ontology matching problem. It is observed that our consistency based aggregation method gets best performance in all the matching tasks except in #303. And only our aggregation method gets better performance than best individual matchers on average.

Table 5. F-measure of individual matchers.

Dataset	BEST Indi	AVG	MAX	MIN	SIG	CONSIS
301	83.33%	85.45%	86.24%	58.41%	84.40%	86.62%
302	70.89%	69.23%	68.42%	39.13%	68.35%	73.79%
303	84.09%	80.00%	81.32%	77.27%	79.12%	83.00%
304	94.19%	96.10%	94.12%	74.81%	95.30%	97.30%
Russia	70.88%	71.46%	71.46%	26.76%	71.01%	73.46%
Animals	90.91%	88.37%	90.91%	8.89%	90.91%	92.91%
Tourism	88.94%	89.34%	88.44%	83.42%	88.21%	91.16%
Peo-pet	83.95%	83.95%	81.48%	79.27%	83.95%	85.95%
Avg	83.40%	82.99%	82.80%	55.99%	82.66%	85.52%

V. CONCLUSION AND FUTURE WORK

In this paper, we study semantic technology for integrating heterogeneous information in the IoT. We propose a consistency based aggregation method for automatically combining multiple matchers in ontology matching tasks. The new method makes use of ontology hierarchical information to compute weights for similarity values. We test our method with four popular matchers in eight ontology matching tasks; the experimental results show that our method can get better results than other aggregation methods.

The reliability of similarity values is evaluated by checking to what extent they satisfy the hierarchical constraint of ontology. Since some other constraints can also be obtained when there are axioms asserted in an ontology, we will investigate various kinds of conflicts might occur in ontology matching, and added new constraints to our consistency based aggregation method.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61171014, 61170203) and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Kevin Ashton: That 'Internet of Things' Thing. In: RFID Journal, 22 July 2009.

[2] Atzor, L., A. Iera, G. Morabito, The Internet of Things: A survey, Computer Networks, 2010. 54(15): p. 2787-2805

[3] I. Toma, E. Simperl, Graham Hench, A joint roadmap for semantic technologies and the internet of things, in: Proceedings of the Third STI Roadmapping Workshop, Crete, Greece, June 2009.

[4] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, V. Terziyan, Smart semantic middleware for the internet of things, in: Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics, Funchal, Madeira, Portugal, May 2008.

[5] Zhexuan Song, Cárdenas, A.A., Masuoka, R. , Semantic middleware for the Internet of Things, *Internet of Things (IOT), 2010* , vol., no., pp.1-8.

[6] Euzenat, J. and P. Shvaiko, Ontology matching. 2007, Heidelberg (DE): Springer-Verlag. 341.

[7] Shvaiko, P. and J. Euzenat, Ten Challenges for Ontology Matching, in On the Move to Meaningful Internet Systems: OTM 2008. 2008. p. 1164-1182.

[8] Juanzi, L., et al., *RiMOM: A Dynamic Multistrategy Ontology Alignment Framework.* IEEE Trans. on Knowl. and Data Eng., 2009. **21**(8): p. 1218-1232.

[9] Stoilos, G., G. Stamou, and S. Kollias, *A String Metric for Ontology Alignment*, in *The Semantic Web – ISWC 2005*. 2005. p. 624-637.

[10] Doan, A., et al., *Learning to match ontologies on the Semantic Web.* The VLDB Journal, 2003. **12**(4): p. 303-319.

[11] Yuzhong, Q., H. Wei, and C. Gong, Constructing virtual documents for ontology matching, in Proceedings of the 15th international conference on World Wide Web. 2006, ACM: Edinburgh, Scotland.

[12] Hu, W. and Y. Qu, *Falcon-AO: A practical ontology matching system.* Web Semantics: Science, Services and Agents on the World Wide Web, 2008. **6**(3): p. 237-239.

[13] Mao, M., Y. Peng, and M. Spring, *An adaptive ontology mapping approach with neural network based constraint satisfaction.* Web Semantics: Science, Services and Agents on the World Wide Web. **8**(1): p. 14-25.

[14] Thomas, R.G., A translation approach to portable ontology specifications. Knowl. Acquis., 1993. **5**(2): p. 199-220.

[15] Cimiano, P., Ontology Learning and Population from Text. 2006, Springer US.

[16] Jean-Mary, Y.R., E.P. Shironoshita, and M.R. Kabuka, *Ontology matching with semantic verification.* Web Semantics: Science, Services and Agents on the World Wide Web, 2009. **7**(3): p. 235-251.

[17] Meilicke, C. and H. Stuckenschmidt. Analyzing mapping extraction approaches. in Proceedings of ISWC 2007 Ontology Matching Workshop. 2007. Busan, Korea.