

Preliminary Evaluation of Multilevel Ontology Integration on the Concept Level

Adrianna Kozierekiewicz-Hetmańska^(✉) and Marcin Pietranik

Faculty of Computer Science and Management, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{adrianna.kozierekiewicz,marcin.pietranik}@pwr.edu.pl

Abstract. In many real situations it is not possible to merge multiple knowledge bases into a single one using one-level integration. It could be caused, for example, by high complexity of the integration process or geographical distance between servers that host knowledge bases that expected to be integrated. The paralleling of integration process could solve this problem and in this paper we propose a multi-level ontology integration procedure. The analytical analysis pointed out that for presented algorithm the one- and multi-level integration processes give the same results (the same final ontology). However, the multi-level integration allows to save time of data processing. The experimental research demonstrated a significant difference between times required for the one- and multi-level integration procedure. The latter could be even 20% faster than the former, which is important especially in the emerging context of Big Data. Due to the limited space we can only consider integration on the concept level.

1 Introduction

The processing of big sets of data is becoming essential problem in case of a company management. It could be stored in different, complex structures and reveal potential inconsistencies, therefore, its processing it is not an easy task. Especially, the integration of such datasets (combining a few separate data source into single one) can be both time- and cost-consuming due to the computational complexity of this process.

Let us imagine a situation in which some company needs to process a large amount of financial data coming from many different sources. Based on a final knowledge base obtained during such integration, the company's board would like to make some decisions about new investments. Too long time of processing could not be a problem in case of a longterm investments with distant deadlines. However, in many situations decisions such as selling or buying new assets, should be made quickly, even in real time and a potential delay could bring potential losses for the company. In other words, the dynamically changing environment requires easy and fast methods for data management and the time of data processing seems to be critical element for companies which need to make decisions based on a Big Data that constantly appear from different sources.

Obviously ontologies, which are the main topic of the following article, shouldn't be treated only as raw data, but more complex knowledge representations. Nevertheless, the context of gathering large amount of data from different sources that can be further processed and eventually obtain intentional semantics require not only effective methods of aforementioned data processing, but also equally effective methods for dealing with large-scale knowledge bases.

In this paper we propose a procedure for ontology integration which can serve as such source. Due to the structure of these knowledge bases, their integration needs to be done on three levels: the concept's level, the instance's level and the relation's level. Due of the limited space available, authors concentrate only on a concept level, using an algorithm taken from [10]. The definition of the multi-level integration process is proposed and the results of one- and multi-level integrations are analysed analytically. However, as it was mentioned, in the case of Big Data, the critical issue is the time required for the integration. Therefore, we have used a set of example ontologies and alignments between them, in order to compare the times required by one- and two-level integration procedures to designate final results. To conduct described comparison, a dedicated experimental environment has been implemented using Python programming language and eventually used.

The remaining part of this paper is organized as follows. In the next section we give a brief summary of related works. Section 3 contains the introduction to ontologies and basic notions used throughout our research. In Sect. 4 the multi-level integration procedure is presented. Section 5 describes the results of analytical and experimental analysis of one- and multi-level ontology integrations. Section 6 concludes this paper.

2 Related Works

Since ontologies are becoming more and more popular, the problem of their integration (also referenced as merging) and their mapping are becoming very important. Cruz and Xiao in [4] discussed the role of ontologies in data integration. They considered two different settings depending on the system architecture: central and peer-to-peer data integration.

The problem of ontology integration were raised in many papers. In [15] authors describe activities that compose integration process like: identifying integration possibility, decomposition into modules of integration, initial assumptions and ontological commitments. In general, the process can be decomposed into choosing the right representation of knowledge in each module, selecting candidate ontologies, studying and analysing candidate ontologies, choosing source ontologies, applying integration operations and finally processing a resulting ontology. For each stage of a methodology it provides support and guidance to perform those activities.

In [2] authors presented the basic framework for ontology integration. They tried to answer how to specify the mapping between the global ontology and local ontologies and eventually have proposed a mechanisms based on queries. Noy and Musen described a general approach to ontology merging and alignment called SMART [12] and PROMPT [13].

Li and his team in [8] described an agent-based framework of integration of similar ontologies coexisting in a distributed and heterogeneous environment. The basic remark which served as an initial inspiration was the fact that with the presence of ontology agents, newly generated ontologies can be reused many times. The proposed solution was tested in a prototype system implemented using Jade framework. Considered research pointed out that the proposed framework provides a flexible and effective modelling approach to tackle the integration over a variety of ontologies.

In [1] a hybrid approach for ontology integration is proposed. Authors distinguished two major approaches to integration of information: (i) the data warehouse (materialised approach) and (ii) a virtual approach (also referenced as mediator-based approach). They took advantages of both and proposed a hybrid framework.

In [5] authors have presented a set of methods facilitating the integration of independently developed ontologies using mappings.

In [10] author defined ontology and subsequently described some integration techniques. Due to the accepted ontology definition the integration process were decomposed into three levels: the concept's level, the relation's level and the instance's level. For each of these levels the suitable methods were proposed and analysed.

The integration on two or more levels is a new idea and so far it has not attracted much attention in literature. There are however some papers like [6,7] or [11] that address the one- and two-level consensus and the problem of its' determination. Authors have developed a formal framework that can be easily used to designate the consensus in one- and two- steps for assumed macrostructure and microstructures. Next, for some criteria the analysis of obtained consensus were made. The researches demonstrated that both one-level and two-level consensus in comparison to the optimal solution give acceptable results. Nevertheless, to the best of our knowledge the challenges of the multi-level ontology integration topic were not frequently addressed.

In our previous research we have also focused on the problem of ontology alignment [14] which can be treated as a pre-step to any other ontology integration process. In general, the task of designating an alignment can be described as a process of selecting elements of compared ontologies that refer to the same object taken from considered universe of discourse [16]. What distinguishes our work from other research is the fact that we have developed a framework built around four functions (namely $\lambda_A, \lambda_C, \lambda_R$ and λ_I) that are used to calculate the degree to which certain elements from some selected source ontology can be mapped to elements from a target ontology. What is worth emphasising, is the fact that these functions are not symmetrical. The reason behind this comes from straightforward remark - it is easier to align detailed representation of some object (no matter if it is an attribute, a concept, a relation etc.) into general representation than to map broad description into precise one without any loss of information. Therefore, our framework does not designate the closeness of two ontological elements, but the amount of knowledge that can be unequiv-

ocally transformed. Obviously, the above consideration does not entail formal asymmetry of concerned functions $\lambda_A, \lambda_C, \lambda_R, \lambda_I$.

3 Basic Notions

Lets assume that a pair (A, V) represents some real world in which A denotes a set of attributes that can describe objects from that world and V denotes a set of valid valuations of these attributes. In other words, we can say that $V = \bigcup_{a \in A} V_a$ where V_a is a domain of an attribute a .

On the simplest level we define ontologies as a following triple:

$$O = (C, R, I) \quad (1)$$

where C is a finite set of concepts, R is a finite set of relations between concepts $R = \{r_1, r_2, \dots, r_n\}$, $n \in N$, $r_i \subset C \times C$ for $i \in [1, n]$ and I is a finite set of instances.

Elements of the set of concepts (also referenced as classes) C are defined as follows:

$$c = (Id^c, A^c, V^c) \quad (2)$$

where Id^c is a unique label, A^c is a set of attributes assigned to such concept and V^c is a set of domains of these attributes ($V^c = \bigcup_{a \in A^c} V_a$).

If the criteria $\forall_{c \in C} A^c \subseteq A$ and $\forall_{c \in C} V^c \subseteq V$ are met we can say that an ontology O is (A, V) -based.

Attributes from the set A by themselves do not carry any particular meaning. They obtain semantics by being included within particular concepts. In order to formally express it we need a set D_A of their atomic descriptions (e.g. *year_of_birth*) and in consequence a sublanguage of the sentence calculus constructed with members of D_A and elementary logic operators of conjunction, disjunction and negation. Eventually the semantics of attributes is given by a function:

$$S_A : A \times C \rightarrow L_s^A \quad (3)$$

The above equation allows to specify roles that variety of attributes obtain when they get included into different concepts. For example, an attribute *Address* means something different when used in the context of a concept *Home* and different when included in the concept *Website*. Furthermore, such approach to expressing attributes' semantics gave us a possibility to formally define *equivalency* (denoted as \equiv), *generalization* (denoted as \uparrow) and *contradiction* (denoted as \downarrow) between attributes [14].

We also accept the existence of a set D_R containing descriptions of relations.

By analogy, L_s^R denotes another sublanguage of the sentence calculus that is used to define a function that gives semantics of relations from the set R :

$$S_{R,O} : R \rightarrow L_s^R \quad (4)$$

Hence, we have provided a set of criteria for relationships between relations including equivalency, generalisation and contradiction.

An instance i (a member of the set I) of some concept is defined as a triple $i = (id, A_i, v_i)$, where id is its unique identifier, A_i stands for a set of assigned attributes and v_i denotes a function $v_i : A_i \rightarrow \bigcup_{a \in A_i} V_a$ which assigns values from the corresponding sets V_a to particular elements of the set A_i . We say that $i = (id, v_i, A_i)$ is an instance of a concept $c = (Id^c, A^c, V^c)$ only if $A^c \subseteq A_i$ and $\forall a \in A_i \cap A^c v_i(a) \in V^c$. For convenience we will use the notation $Ins(O, c)$ to denote a set of instances of a concept c within ontology O .

4 Multi-level Integration

Out of many ways of defining the knowledge integration, we can describe it as a process of joining several, independent knowledge bases (in our case - ontologies) into a single one. In some cases it is impossible to do so during only one-level integration due to high complexity of required transformations or simply geographical distance between them that entails unacceptable latency due to too large data transfer.

A multi-level knowledge integration, i.e. simultaneously combining knowledge from a small number of sources for many subgroups and the eventual merging of the results into the one final knowledge base, might be applied to solve the described issue. The general idea for such approach is presented in Fig. 1.

The problem of ontology integration can be formulated as follows: *for given n ontologies O_1, O_2, \dots, O_n one should determine an ontology O^* which represents given ontologies in the best way.* As it was mentioned, the integration process can be conducted on one level or in special cases on two or more levels. The definitions of one level and multi level ontology integration process is presented below:

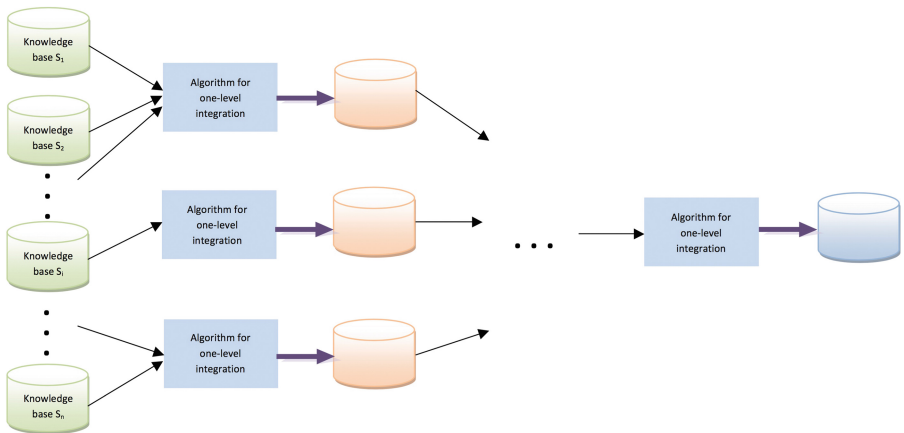


Fig. 1. The general idea for a multi-level integration process.

Definition 1. *The input of the one-level integration process is a sequence of n ontologies: $O_1^1, O_2^1, \dots, O_n^1$. The output of the integration process is a single ontology O^{1*} , which is in multiple relationships with input ontologies, as defined by a group of criteria. Integration criteria $K_1^1, K_2^1, \dots, K_n^1$ are the parameters of the integration task and tying O^{1*} with $O_1^1, O_2^1, \dots, O_n^1$ each at least at a given level $\alpha_1^1, \alpha_2^1, \dots, \alpha_n^1$ $K_i^1(O^{1*}|O_1^1, O_2^1, \dots, O_n^1) \geq \alpha_i^1$.*

Based on the Definition 1 the multi-level integration is defined as follows:

Definition 2. *Let $O_1^{m-1*}, O_2^{m-1*}, \dots, O_n^{m-1*}$ be ontologies obtained during $m-1$ level of the knowledge integration, where $m \geq 2$. The output of the m -level of integration is a single ontology O^{m*} , which is in multiple relationships with input structures, as defined by a group of criteria: $K_1^m, K_2^m, \dots, K_n^m$.*

According to the literature [9], the following integration criteria are known: *completeness* (after the integration no data/elements are lost), *minimality* (the output of the integration is not much larger than its inputs), *precision* (the integration does not duplicate data), *optimality* (the output of the integration is the closest to inputs, in terms of some distance measure), *sub-tree agreement* (the output includes all the sub-trees from its inputs).

Due to the structure of an ontology which consists of three main elements: concepts, relations and instances, the problem of one-level ontology integration should be conducted in three steps: integration of concepts, integration of relations and integrations of instances. This problem has been solved in [10] where author decomposed problem of ontology integration into three phases and for each phase the appropriate algorithm were proposed. Integration on an instance level were solved using consensus methods, integration on a concept level required defining some additional postulates and an algorithm for relational level includes in the final set of relations only those relations which appear most often in the ontologies, and do not cause any contradiction.

The multi-level ontology integration task required to primarily divide the sequence of n ontologies O_1, O_2, \dots, O_n into k classes X_1, X_2, \dots, X_k where $k < n$. For each class X_i of ontologies one-level integration process is conducted in the way described above. Ontologies $O_1^{2*}, O_2^{2*}, \dots, O_k^{2*}$ are the result of the integration process obtained during 2nd level. Ontologies $O_1^{2*}, O_2^{2*}, \dots, O_k^{2*}$ can be further integrated (based on basic one-level integration procedure) into the one, final ontology O^{2*} . The division of a sequence of ontologies into classes and integrating them can be carried out many times and then we can say about the multi-level integration process.

5 Evaluation of One- and Multi-level Ontology Integration

5.1 Formal Analysis of Integration Algorithm

Due to the limited space available for this paper we have focused only on the evaluation of one- and multi-level concept integration. The base algorithm taken from [10] is conducted in the following steps:

Algorithm 1. Concept integration**Require:** Concepts structures (A^i, V^i) ;

-
- 1: Set $A^* = \bigcup_{i=1}^n A^i$;
 - 2: **for all** $(a, b) \in A^* \times A^*$ **do**
 - 3: **if** $a \uparrow b$ **then**
 - 4: $A^* = A^* \setminus b$ if b does not occur in any relationship with any other attribute from A^* ;
 - 5: **end if**
 - 6: **end for**
 - 7: **for all** $a \in A^*$ **do**
 - 8: determine its domain V_a as the sum of its domains from pairs (A_i, V_i)
 - 9: **end for**
-

Theorem 1. For an ontology integration on a concept level and for $m \geq 2$ the following condition is always satisfied: O^{m*} is equal to O^{1*} .

Proof. In the first step we show that O^{m*} is equal to O^{1*} for $m = 2$. Due to the fact that we consider only concept integration we want to show that A^{m*} is equal A^{1*} and V^{m*} is equal V^{1*} where A^{m*} , A^{1*} are the results of attribute integration on multi- and one-level respectively and V^{m*} , V^{1*} are integrated values of attributes for multi- and one-level algorithm.

From Step 1 of Algorithm 1 it is obvious that $A^{1*} = \bigcup_{i=1}^n A^i$. Two-level integration process is more complicated. Let us assume that A_1, A_2, \dots, A_n were divided into k classes. Therefore, $S_1 = \{i : A_i \text{ belongs to a class 1}\}, S_2 = \{i : A_i \text{ belongs to a class 2}\}, \dots, S_k = \{i : A_i \text{ belongs to a class } k\}$. In the first step of the multi-level integration process we obtain $A_1^{1*} = \bigcup_{i \in S_1} A^i, A_2^{1*} = \bigcup_{i \in S_2} A^i, \dots, A_k^{1*} = \bigcup_{i \in S_k} A^i$. In the second step we get $A^{2*} = \bigcup_{i \in S_1} A^i \cup \bigcup_{i \in S_2} A^i \cup \dots \cup \bigcup_{i \in S_k} A^i$. Therefore, A^{2*} is A^{1*} equal because union of sets is associative. The same reasoning could be conducted for the set of attributes values. For $m \geq 2$ it is easy to show by using mathematical induction. \square

From Theorem 1 we know that the results of ontology integration for one- and multi-level give the same results. Therefore, in the next part of our paper we examine the influence that one- and multi-level integration processes have on the time required to determine the final ontology.

5.2 Experimental Evaluation

For experimental evaluation we have used ontologies taken from datasets provided by *Ontology Alignment Evaluation Initiative (OAEI)* for their annual evaluation campaigns. These campaigns are aimed at evaluation of plethora of ontology alignment frameworks which main goal is to designate a set of mappings

that indicate equivalent elements taken from separate ontologies. The aforementioned evaluation methodology is based on a broad dataset containing pairs of ontologies (for convenience grouped into smaller subsets referred to as *tracks*) along with some gold standard - a reference mappings between them. During the actual evaluation of some selected alignment tool, its output is compared with such reference mappings and *Precision* and *Recall* values are calculated along with other quality metrics.

Due to the accessibility of the domain, for our particular experiment we have used four ontologies (namely *Sigkdd*, *Edas*, *ConfTool* and *Sofsem*) taken from the conference track of the latest 2015 evaluation campaign [17]. We have also used the provided reference alignments that have been designated between these ontologies in order to fulfil initial requirements: (i) selecting equivalent concepts that may be integrated into the final ontology and (ii) selecting equivalent attributes for the sake of Step 3 of Algorithm 1.

In our experiment we have tested the one- and two-level approach using a dedicated experimental environment written in Python programming language. The integration of all ontologies into the one, consistent version incorporating standard one-level approach took **0.0788 s**.

In Table 1 we present different times in seconds taken by the two-level integration approach. We have tested seven different selections of initial ontologies' classes X_1 and X_2 . They can be understood as an initial division of used set of four ontologies into subsets containing respectively one, two or three of initial ontologies. Each of such division is represented as a row in Table 1.

Table 1. Experimental results for two-level integration process

Ontologies classes	1 level		2nd level	Total time
	X_1	X_2		
$X_1 = \{\text{Sofsem, Sigkdd}\}$ $X_2 = \{\text{Edas, ConfTool}\}$	0.006 s	0.0072 s	0.06 s	0.0672 s
$X_1 = \{\text{Sofsem, Edas}\}$ $X_2 = \{\text{Sigkdd, ConfTool}\}$	0.0026	0.0098 s	0.0564 s	0.0662 s
$X_1 = \{\text{Sofsem, ConfTool}\}$ $X_2 = \{\text{Sigkdd, Edas}\}$	0.007 s	0.0071 s	0.0559 s	0.0631 s
$X_1 = \{\text{Sofsem}\}$ $X_2 = \{\text{Sigkdd, Edas, ConfTool}\}$	0 s	0.0212 s	0.0628 s	0.084 s
$X_1 = \{\text{Sigkdd}\}$ $X_2 = \{\text{Sofsem, Edas, ConTool}\}$	0 s	0.0323 s	0.0557 s	0.088 s
$X_1 = \{\text{Edas}\}$ $X_2 = \{\text{Sofsem, Sigkdd, ConfTool}\}$	0 s	0.0182 s	0.0573 s	0.0756 s
$X_1 = \{\text{ConfTool}\}$ $X_2 = \{\text{Sofsem, Sigkdd, Edas}\}$	0 s	0.0348 s	0.0764 s	0.1111 s

Columns represent times taken by each level of performed integration for different classes and the time taken by the final step of the investigated method.

The presented values are obtained from 10 repeats of the same integration process and the arithmetic means of all of the times taken by partial iterations is provided. This allowed to rule out any potential distortions that may be caused by random technical issues such as memory access downtime etc.

From obtained results of our experiment we can draw a conclusion that the multi-level approach to the integration is significantly faster than the one-level procedure. As easily seen, from the last column of Table 1, the experimental verification pointed out that such integration process is shorter even by 20% in comparison to the simple one-level integration. In the context of Big Data [3] the shortest possible time required to obtain the expected results is a critical factor in providing reliable business solutions in due course.

6 Future Works and Summary

Because of the complexity of ontologies and their semantic expressiveness, managing them is a difficult task. Moreover, ontologies allow to easily store big sets of data (eventually enriching them with some intentional meanings), so methods for their convenient, quick, reliable and low-budget processing are critical.

In this work we have proposed the multi-level method of their integration. During this process, integrated ontologies are divided into some classes and for each of such class the one, consistent ontology is designated. Finally all of the partial results are merged into a final ontology. Such solution allows to decrease the time required for performing desired integration thanks to a parallelisation of the fragmentary calculations.

In our future work we would like to conduct more experiments using more ontologies and for more levels. Due to the limitations of this paper we were able only to examine four ontologies integrated only on two levels. Therefore, more sophisticated experiments could bring interesting conclusions. Additionally, we are planing to expand our framework with the integration of both instances and relations that are also important elements of ontologies.

Acknowledgment. This work was partially supported by the European Commission under the 7th Framework Programme, Coordination and Support Action, Grant Agreement Number 316097, ENGINE - European research centre of Network intelliGence for INnovation Enhancement (<http://engine.pwr.wroc.pl/>).

References

1. Alasoud, A., Haarslev, V., Shiri, N.: A hybrid approach for ontology integration. In: Proceedings of the 31st VLDB Conference, Trondheim, Norway (2005)
2. Calvanese, D., Giacomo, G., Lenzerini, M.: A framework for ontology integration. In: Proceedings of the 2001 International Semantic Web Working Symposium (SWWS 2001), pp. 303–316 (2010)

3. Chen, H., Chiang, R.H., Storey, V.C.: Business intelligence and analytics: from big data to big impact. *MIS Q.* **36**(4), 1165–1188 (2012). Society for Information Management and the Management Information Systems Research Center, Minneapolis, MN, USA
4. Cruz, I.F., Xiao, H.: The role of ontologies in data integration. *J. Eng. Intell. Syst.* **13**, 245–252 (2005)
5. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Berlanga, R.: Ontology integration using mappings: towards getting the right logical consequences (2009). doi:[10.1007/978-3-642-02121-3_16](https://doi.org/10.1007/978-3-642-02121-3_16)
6. Kozierekiewicz-Hetmańska, A.: Comparison of one-level and two-level consensuses satisfying the 2-optimality criterion (2012). doi:[10.1007/978-3-642-34630-9_1](https://doi.org/10.1007/978-3-642-34630-9_1)
7. Kozierekiewicz-Hetmańska, A., Nguyen, N.T.: A comparison analysis of consensus determining using one and two-level methods, vol. 243. *Advances in Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 159–168 (2012)
8. Li, L., Wu, B., Yang, Y.: Agent-based ontology integration for ontology-based applications. In: Meyer, T., Orgun, M. (eds.) *Proceedings of the Australasian Ontology Workshop*, vol. 58, Sydney, Australia (2005)
9. Maleszka, M., Nguyen, N.T.: A model for complex three integration tasks (2011). doi:[10.1007/978-3-642-20039-7_4](https://doi.org/10.1007/978-3-642-20039-7_4)
10. Nguyen, N.T.: *Advanced Methods for Inconsistent Knowledge Management*. Springer, London (2008)
11. Nguyen, N.T.: *Consensus Choice Methods and Their Application to Solving Conflicts in Distributed Systems*. Wrocław University of Technology Press, Wrocław (2002). (in Polish)
12. Noy, N.F., Musen, M.A.: An algorithm for merging and aligning ontologies: automation and tool support. In: *Proceedings of the Workshop on Ontology Management at the Sixteenth National Conference on Artificial Intelligence (AAAI 1999)* (1999)
13. Noy, N.F., Musen, M.A.: PROMPT: algorithm and tool for automated ontology merging and alignment. In: *AAAI/IAAI*, pp. 450–455 (2000)
14. Pietranik, M., Nguyen, N.T.: A Multi-attribute based framework for ontology aligning. *Neurocomputing* **146**, 276–290 (2014). doi:[10.1016/j.neucom.2014.03.067](https://doi.org/10.1016/j.neucom.2014.03.067)
15. Pinto, M., Martins, J.P.: A methodology for ontology integration. In: *Proceedings of K-CAP 2001*, Victoria, British Columbia, Canada, 22–23 October 2001
16. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. *IEEE Trans. Knowl. Data Eng.* **25**(1), 158–176 (2013)
17. <http://oaei.ontologymatching.org/2015/>