Semi-Automatic Definition of Attribute Semantics for the Purpose of Ontology Integration

BOGUMIŁA HNATKOWSKA, ADRIANNA KOZIERKIEWICZ, AND MARCIN PIETRANIK
Applied Informatics Department, Wrocław University of Science and Technology, 50-370 Wrocław, Poland
Corresponding author: Adrianna Kozierkiewicz (adrianna.kozierkiewicz@pwr.edu.pl)

ABSTRACT By ontology, we understand a knowledge structure which well reflects the complexity of a real world. Ontologies are built to store and process knowledge about objects and dependencies between them. Thus, ontologies not only structure raw data, but also contain the meaning of those data. So far, ontology developers have been forced to provide the semantics of modeled objects and relations between them manually. The goal of this paper is to address some still unresolved problems related to providing meanings in ontologies. A typical ontology consists of concepts with attributes, relations between them, and instances. In our research, we focus on concepts level where attributes must be interpretable because they are the primary carriers of the meaning of the entire ontology. In other words, we need to assign semantics for each attribute within a concept. For this purpose, we have proposed a semi-automatic method for defining attribute semantics based on WordNet. The attribute semantics designated by human experts and based on WordNet does not affect the integration result. However, the developed method allows reducing the time for preparing an ontology to the integration process. What is more, it is less sensitive to the subjective evaluations done by experts.

INDEX TERMS Data integration, semantics, semantic Web, WordNet.

I. INTRODUCTION
The rate of growth of the amount of data (in the Internet, private clouds, or local data centers) is enormous. Data are stored often in distributed sources; thus, for a decision-making process based on collected data, efficient integration methods are desirable. However, there is a question on how to evaluate if such integration is efficient or not.

In literature [21] it is easy to find some measures which allow calculating the effectiveness of an integration. For example, completeness allows checking how much knowledge is lost after an integration. Precision estimates how many elements are duplicated and how many new elements are introduced after a integration. Optimality tells how close the output of the integration is to its inputs in terms of some accepted distance measure. However, all these measures require to conduct an integration process beforehand.

Our framework for ontology integration, presented in previous work [11], contains a measure for estimation of effectiveness of an integration. This measure can be calculated before integration is carried out. It can be treated as an indicator of a potential increase of knowledge resulting from an integration process. We have called it a “growth of knowledge” metric (Δ). For example, if two input structures contain the same knowledge, then nothing is gained from their integration. On the contrary, if contents of input sources are entirely separate, then their merging can be valuable because during the integration process our knowledge has doubled.

The framework operates on an ontology model defined at different abstraction levels. The lowest abstraction level defines attributes with their domains. On that basis a definition of concepts and their attributes is provided together with attributes’ semantics. The highest level introduces relations between concepts with their semantics.

At this level it is possible to clearly define varying meanings that attributes may obtain when being assigned to different concepts. For example, the attribute Address has a different meaning when it is assigned to the concept Person and completely different when it is part of the concept Internet Website. This observation has become the basis for our research presented in this paper. This diversity will be formally defined as a function that assigns logic statements to every inclusion of attributes within concepts.

The developed ontology framework [11] assumes the existence of the function $S_{A}$ mentioned above, which provides...
semantics for every attribute in terms of a first-order logic formula built from atomic descriptions of attributes, e.g., $S_A = (\text{birthdate}, \text{Person}) = \text{year_of_birth} \land \text{month_of_birth} \land \text{day_of_birth}$. So far, we have assumed that attributes’ semantics are provided directly by experts or ontology developers. However, if a collection of concepts and attributes is huge, the mentioned approach is time-consuming, subjective (definition of functions may vary depending on the experts), and error-prone (the risk of inconsistency increases with the number of attributes). For example, for four classes with four attributes each, there are sixteen formulas to be provided that should be complete and consistent. The subjectivity could be partially eliminated by involving more experts for function evaluation or definition. Either additional experts can check the original definitions against quality issues, or they can provide their definitions and try to find a consensus. However, it makes the whole procedure even more expensive and thus, impractical.

The paper addresses the aforementioned problems by a semi-automatic method used for establishing the attribute semantics, i.e. the definition of the $S_A$ function. This function is used to determine integration results and is also the main contributor to calculation of growth of knowledge metric ($\Delta$) at the concept level. However, there is an open question on how to provide its values in an objective and consistent manner. The method proposed in this paper uses WordNet as a base of knowledge. It also requires presence of a domain expert, but with a minimal involvement compared to the fully manual approach. The method was implemented in a tool, which enables its practical assessment by a set of experiments. The paper aims to present the proposed method together with a list of decisions and justifications made for the framework adaptation for the integration of OWL ontologies.

The rest of the paper is organized as follows. The first part contains basic ontology definitions that are a mathematical foundation for our research. Section 3 presents a motivating example which justifies the importance of the problem raised in this paper. Section 4 characterizes the current state of knowledge in the considered field. Section 5 provides a detailed description of a semi-automatic method for establishing attribute semantics. This section consists of a detailed description of WordNet, an ontology integration process on the concept level and implementation method. Section 6 presents the results of an experiment which proves usability of the method. A summary and a short overview of our upcoming research can be found in Section 7.

II. BASIC NOTIONS

By using a pair $(A,V)$ we define a real world, where $A$ is a finite set of attributes that can be used to describe objects, and $V$ is a set of values of such attributes (their domains). By $V_a$ we denote a domain of a particular attribute $a$, hence it is obvious that the following property occurs: $V = \bigcup_{a \in A} V_a$.

We define an $(A,V)$-based ontology as a quintuple [25]:

$$O = (C, H, R^C, I, R^I)$$

(1)

where:

- $C$ is a finite set of concepts,
- $H$ is a concept hierarchy (this element is not applied in this paper, therefore we do not provide its formal definition; for details please refer to [11]),
- $R^C$ is a finite set of relations between concepts, $R^C = \{r^C_1, r^C_2, \ldots, r^C_n\}$, $n \in N$, such that every $r^C_i \in R^C$, $i \in [1, n]$ is a subset of cartesian product $r^C \subseteq C \times C$,
- $I$ denotes a finite set of instance identifiers,
- $R^I = \{r^I_1, r^I_2, \ldots, r^I_n\}$ is a finite set of relations (complementary to elements of the set $R^C$) between concepts’ instances.

A concept taken from the set $C$ is defined as:

$$c = (id^c, A^c, V^c, I^c)$$

(2)

where:

- $id^c$ is an identifier of the concept $c$,
- $A^c$ is a set of its attributes, such that $A^c \subseteq A$,
- $V^c$ is a set of attributes domains (formally: $V^c = \bigcup_{a \in A^c} V_a$),
- $I^c$ is a set of $c$ instances, defined as pairs $(id^1, v^1)$, where $id^1$ is an instance identifier from the $I$ set and $v^1$ is a function with the signature $v^1 : A^c \rightarrow V^c$ which assigns specific values to the attributes of the $c$ concept.

For short, we write $a \in c$ to denote that an attribute $a$ belongs to the concept’s $c$ set of attributes.

By $O$ we denote a family of all $(A,V)$-based ontologies.

We can say that some ontology is $(A,V)$-based if the following condition are all met:

- $\forall c \in C A^c \subseteq A$,
- $\forall c \in C V^c \subseteq V$.

Attributes from the set $A$ do not have any semantics. They become interpretable only when they are a part of a selected concept. Let $D_A$ be a set containing atomic descriptions of attributes (for example, it may contain a label $\text{address}$). Subsequently, we define a sub-language of the sentence calculus built from elements of $D_A$ and logic operators of conjunction, disjunction and negation. We denote it as $L^A$ and we use it for description of semantics of attributes. Formally, we can define a function which assigns a logical sentences from $L^A$ to attributes within particular concept:

$$S_A : A \times C \rightarrow L^A$$

(3)

Such approach allows to formally define different meanings that one attribute can express whereas being a part of different concepts. For example, the $\text{address}$ attribute within the $\text{Person}$ concept can obtain the following semantics: $S_A(\text{address}, \text{Person}) = (\text{street_name} \land \text{number_of_flat} \land \text{postal_code} \land \text{city} \land \text{country})$. However, its meaning may change when it is used within different concepts, e.g. within the $\text{Homepage}$ concept $S_A(\text{address}, \text{Homepage}): (\text{www_address} \land \text{domain})$.

For calculating a distance between two logic statements expressed using the language $L^A$ we define a function with
the following signature:

\[ dist : L^A \times L^A \rightarrow [0, 1] \] (4)

The function \( dist \) meets the criteria:

- \( dist(a, b) = 0 \) \( \iff \) \( a = b \)
- \( dist(a, b) = dist(b, a) \).

Based on previously defined function \( S_A \) we can formally define how attributes included in concepts are related to each other. We distinguish three relations between attributes: equivalence (denoted as \( \equiv \)), generalization (denoted as \( \rightarrow \)) and contradiction (denoted as \( \sim \)):

- Two attributes \( a \in A^c, b \in A^c \) are semantically equivalent if the formula: \( S_A(a, c) \iff S_A(b, c') \) is a tautology for any two \( c \in C, c' \in C' \). It can be denoted as \( a \equiv b \).
- The attribute \( a \in A^c \) in concept \( c \in C \) is more general than the attribute \( b \in A^c \) in concept \( c' \in C' \) if the formula \( S_A(b, c') \Rightarrow S_A(a, c) \) is a tautology. It can be denoted as \( a \rightarrow b \).
- Two attributes \( a \in A^c, b \in A^c \) such that \( c \in C, c' \in C' \), are semantically contradicting if the formula \( \sim (S_A(a, c) \leftrightarrow S_A(b, c')) \) is a tautology. It can be denoted as \( a \sim b \).

The overall meaning that a particular concept can carry is described by a context function with the following signature:

\[ ctx : C \rightarrow L^A \] (5)

This function can be defined as the conjunction of logical sentences for all attributes embedded within the concept:

\[ ctx(c) = S_A(a_1, c) \land S_A(a_2, c) \land \ldots \land S_A(a_n, c) \] (6)

where: concept \( c \) contains a set of attributes \( \{a_1, a_2, \ldots, a_n\} \), \( |A^c| = n \).

To simplify upcoming notions, we assume that the distance between two concepts from two \((A, V)\)-based ontologies \( O_1 \) and \( O_2 \) is calculated as the distance between their concepts. It is a function with a following signature:

\[ d_S : C_1 \times C_2 \rightarrow [0, 1] \] (7)

Function \( d_S \) meets the criteria of identity of indiscernibility and symmetry and is defined as:

\[ d_S(c_1, c_2) = dist(ctx(c_1), ctx(c_2)) \] (8)

In our previous work [11] the problem of estimation of the knowledge increase which may occur during the ontology integration on the concept level was considered. We claim that it is possible to measure the potential knowledge increase before the integration starts. This allows for deciding in advance whether it is worth doing it, in particular when considering other costs.

Thus, we have worked out the measure which allows establishing a balance between cost and effectiveness of the integration process. The estimation of the growth of knowledge (which can be understood as the quality of the integration) would be based on a function with the signature:

\[ \Delta : C_1 \times C_2 \rightarrow [0, 1] \] (9)

Formally the task of creating a tool that allows such estimation can be defined as follows: for given ontologies \( O_1, \ldots, O_n \) and the integration function \( \sigma \), one should determine that the function \( \Delta \) represents the increase of knowledge between these ontologies and the result of their merging defined as \( \sigma(O_1, \ldots, O_n) = O^* \).

Such an approach can be decomposed into several subtasks that cover a variety of ontology elements, namely: concepts, relations among them, their hierarchy, instances, and relations connecting instances. However, because this paper is devoted to the definition of semantics of attributes, we limit the presentation only to the measure defined at the concept level:

\[ \Delta = 1 - \frac{|\text{map}(O_1, O_2)| - \sum_{<s,t> \in \text{map}(O_1, O_2)} \max_{c_i \in \mathbb{C}} d_S(s, c_i)}{|C_1| + |C_2| - \sum_{<s,t> \in \text{map}(O_1, O_2)} |t|} \] (10)

where \( \text{map}(O_1, O_2) \) contains tuples in the form \(<s, t> : s \) represents a concept either from \( O_1 \) or \( O_2 \) and \( t \) represents a set of concepts from the corresponding ontology that can be unequivocally matched with the given \( s \). In other words, it is a set of candidate concepts that have been chosen to be integrated with the following property:

\[ \forall_{<s, t> \in \text{map}(O_1, O_2)}((s \in C_1 \implies t \subseteq C_1) \land ((s \in C_2 \implies t \nsubseteq C_2)) \] (11)

It should be mentioned that in further parts of the following article, the \( t \) set is usually limited to contain only one element.

### III. MOTIVATING EXAMPLE

To motivate our work and explain the source of the problems mentioned in the introduction, we use a simple example of the ontology integration. Let’s assume there are two ontologies \( O_1 \) and \( O_2 \) to be integrated. Each ontology introduces only one concept \texttt{Person} with a few attributes. Informal definitions of the ontologies are given below.

\[ O_1 = \{\texttt{Person}(\text{name} : \text{string}, \text{birthdate} : \text{date})\} \]
\[ O_2 = \{\texttt{Person}(\text{first\_name} : \text{string}, \text{last\_name} : \text{string}, \text{age} : \text{integer})\} \]

For the ontologies’ integration, the value of \( S_A \) function has to be defined for each attribute. This task was done separately for each of the paper co-authors, who worked as domain experts. In Table 1, the function together with its application to the ontologies’ integration and knowledge increase metric calculations are shown.

The merging algorithm [11] considers the existing relationships between attributes and on that basis eliminates some of them. In the considered example, only a generalization relationship is present. The more general attribute is removed as its semantics is included in the semantics of its parent. It should be mentioned that the meaning of genericity is completely opposite to the known one, e.g., from the object-oriented paradigm or Wordnet. In our framework, a more specific attribute contains more information than a
TABLE 1. An example of ontologies’ integration-manual approach.

| Element              | Expert 1                              | Expert 2                              | Expert 3                              |
|----------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| S(name, Person)      | (first_name ∧ last_name) ∧            | first_name ∧ last_name                 | first_name ∧ last_name                 |
|                      | (first_name ∧ middle_name ∧ last_name)|                                       |                                       |
| S(birthdate, Person) | day_of_birth ∧ month_of_birth ∧ year  | day ∧ month ∧ year ∧                  |                                       |
|                      | ∧ year ∧ age ∧ last_name              |                                       |                                       |
| S(first_name, Person)| first_name                            | first_name                            | first_name                            |
| S(last_name, Person) | last_name                             | last_name                             | last_name                             |
| S(age, Person)       | year_of_birth ∧ age                   | age                                   | age                                   |
| Relationships        |                                       |                                       |                                       |
| between attributes   |                                       |                                       |                                       |
|                      | name ← first_name, last_name          | name ← first_name, last_name          | name ← first_name, last_name          |
|                      | name ← name, first_name               | name ← name, first_name               | name ← name, first_name               |
|                      | name ← name, last_name                | name ← name, last_name                | name ← name, last_name                |
|                      | birthdate ← last_name, birthday       | birthdate ← birthday, age             | birthdate ← birthday, age             |
|                      |                                        |                                        |                                        |
| Integrated           | Person(name: string, birthday: date,  | Person(name: string, birthday: date,  | Person(name: string, birthday: date,  |
| ontology G*          | ∨ integer)                           | ∨ integer)                           | ∨ integer)                           |
| Knowledge increase Δ | 0.429                                 | 0.625                                 | 0.667                                 |

IV. RELATED WORKS

Incorporating a lexical database, WordNet in particular, to acquire some additional knowledge about used terminology within some ontology application is not uncommon. Two widely known problems related to ontologies, that were frequently approached using WordNet, are ontology alignment and ontology merging. Both tasks require a method of calculating similarities between two processed ontologies, which can indicate whether or not elements from these ontologies can be matched and eventually merged. In [15], the authors provide a survey of different methods of applying lexical databases to improve these similarity measures in many different ways. An overview of a plethora of such measures is also given in [18].

The main difference between the works mentioned above and the ideas presented in this article lies in the approach to the definition of ontologies. We proved in [25] that an approach to ontology alignment based on analyzing attributes semantics (as described in Section 2) gives good results. However, defining semantics manually is very error-prone, time and cost consuming. It is the reason why we attempted to incorporate WordNet not as a straightforward similarity calculation tool, but rather as a method of enhancing base definitions and making them useful beyond theoretical considerations.

Authors of [28] utilize WordNet as a source of background knowledge for identifying semantic relations between concepts of two matched ontologies. In this context, a matching task is understood as producing mappings (connections) between semantically related nodes of two ontologies. Authors propose a set of element level semantic matchers exploiting WordNet to extend a variety of semantic similarity measures. They were evaluated using OAEI ([30], [31]) datasets, producing good results and no time penalties. However, the aforementioned semantic relations are identified purely lexically, without the context given by ontologies treated holistically.

In [2], it is noticed that contemporary ontology mapping solutions frequently have difficulties with obtaining accurate background information. Eventually, an approach using WordNet as a bridge between two mapped ontologies is proposed. The topic of using lexical information (about hypernymy, hyponymy, holonymy, and meronymy relations connecting words) was also explored in [13] where authors described a Super Word Set tool, which is built on the top of extracting a concept and property mappings between two ontologies. It eventually produces additional concept mappings by finding property mappings using WordNet similarity. Such an approach entails more mappings of concepts taken from mapped ontologies, than other matching methods, which was experimentally proved using already mentioned OAEI datasets and standard precision and recall quality measures. The biggest downside of this research is its experimental verification, which concentrated only on a cardinality of designated alignments, and not their accuracy.
In other words, authors favor the completeness above the precision. Another idea for enhancing ontology alignments is described in [12]. Authors claim that incorporating WordNet, among other lexical databases, as a resource for searching related terms in concept matching procedure, may give promising results.

What differentiates the previous research ([2], [12], [28], [30]) and our work is the separation of WordNet from ontologies and their expressivity. We propose an approach utilizing WordNet to fill the gap that appears when applying the theoretical framework in practical situations.

A similar approach to that presented in this paper can be found in [26], where authors describe a procedure of coupling ontology concepts with corresponding WordNet entries. Such correspondences are further used to generate a set of virtual documents, representing different ontology concepts, which can be eventually incorporated during similarity calculations.

A different perspective is described in [10], where a concept alignment method called DeepAlignment can be found. It is based on refining pre-prepared vectors of terms that are further used in deriving additional concepts descriptions. The proposed solution was evaluated using standard ontology matching benchmarks.

To the best of our knowledge, the research presented in [10] and [26] are the closest to our work. The aforementioned virtual documents ([26]) and term vectors ([10]) can be identified with the concepts’ context from (5) and similarity calculations are close to methods presented in [25]. However, our ideas are independent of WordNet and can be used without any lexical database. Moreover, our framework focuses mainly on attributes and meaning they may carry. Concepts semantics are derived based on their structure – see (2), and (6).

Ontology mapping and merging are not the only WordNet applications in ontology-related issues. Some papers are devoted to automatic creation of ontologies using lexical databases. A comprehensive survey of this research can be found in [4]. This approach was further extended in [3], which presents algorithms that improve interoperability of generated ontologies in a fast lexicon-based integration algorithm.

Issues covered in [3], [4], [23] are somehow similar to the tool described in this paper – the main difference lies in using WordNet. Unlike the overviewed research from [3] and [23], our framework is based not only on WordNet, but also on zero-order logic. In conjunction, they can overcome the downsides that may appear in practical applications of ontologies.

Many ontology matching/integration tools integrated with WordNet are aimed at solving some practical applications. One of them is the SMART tool ([22]) – an implementation of a semi-automatic ontology merging and alignment. It is also capable of detecting potential semantic inconsistencies introduced to maintained ontologies. Authors extended their ideas and created the PROMPT system ([24]), which provides a unified framework allowing to compare, align, and eventually integrate ontologies. This tool aims at being a holistic framework for ontology management, covering a wide array of requirements. Ideas presented in this paper are more focused on one application, which is enhancing a purely theoretical framework with practical capabilities. Neither SMART, nor PROMPT utilize attributes of concepts as foundations for further considerations but concentrate on both concepts’ labeling and their taxonomy.

In [27], authors concentrate on analyzing a utility of WordNet-based solutions in the context of ontology alignment, highlighting the issue of their performance compared to alignment methods that only use syntactic matching techniques. It is proved that using an external lexical database in an ontology matching may improve the quality of designated alignments. However, in many cases, it is also more time-consuming.

The performance issues are also addressed in [1], where a tool called Falcon-AO is presented. It is built around the so-called enrichment step, which extends correspondences between concepts, determined using standard matching approaches, with their semantic type. The method presented by the developers of Falcon-AO incorporate WordNet in calculating lexical similarities between names of concepts. The authors claim that their method is sufficient to designate ontology mappings (which definition is purely based on OWL format and, unlike the approach presented in this paper in Section 2, not on solid mathematical foundations). However, these claims are supported by good results of the experimental evaluation conducted using real benchmark ontologies taken from the Ontology Alignment Evaluation Initiative (OAEI) in the conference Ontology track.

In [16] and [17], a framework called Chimaera is described. It is a tool addressing two problems: (i) merging multiple ontologies and (ii) diagnosing the influence of evolution on ontologies. Beside considering issues related to the diagnosis of coverage and correctness of ontologies or merging different terminologies from varied sources, authors also covered the topic of the performance and calculation of the complexity of implemented procedures. External, lexical databases are used to detect whether internal naming conventions are consistent.

Similarly, ASMOV (Automated Semantic Mapping of Ontologies with Validation) is an ontology mapping tool aimed at facilitating the integration of distributed systems by processing their data source ontologies ([9]). It combines lexical, structural, and extensional approaches to designate ontology mappings, which are further subjected to semantic verification using a lexical database (WordNet). Unlike our approach, in solutions described in [9], [16] and [17], the content extracted from WordNet is not directly included within ontologies. Such utilization of WordNet addresses problems related to the verification of maintained ontologies that are going to be eventually mapped. It asserts no difficulties during the core process of both ASMOV and the earlier described Chimaera.
A concept of “bridging axioms” understood as an auxiliary knowledge source allowing to conduct merging of two related ontologies is given in [5], where authors propose to use a notation which is a strongly typed first-order logic language that is eventually translated to the OWL – the most common ontology representation format. These axioms may somehow resemble attributes semantics (3), however, they are not a part of processed ontologies, but are inferred using a dedicated reasoning engine during the runtime of an ontology integration process, which strongly differs the presented approach from the ideas proposed in the following paper.

Very few ontology management tools address the matter of user interaction when performing tasks which involve ontologies. One of such tools is Potluck ([8]), which provides a user interface that allows ordinary users to perform a difficult task of data integration. Another example is described in [14], where aiding users during even more complex task of integrating biomedical ontologies is addressed. The described system (SAMBO) conducts knowledge merging and reasons about logical consequences. Similar issues are covered in [6], where authors propose to involve users in the process of validating designated ontology alignments. Our utilization of WordNet during enhancement of the theoretical ontology management framework is semi-automatic. The process involves users’ interaction to verify suggestions extracted from WordNet and also to provide any missing information that was not found. The exact users’ workflow is illustrated in Fig. 1.

As easily seen, previous works focus on utilizing WordNet (and in general, lexical databases) as a source of background knowledge, and eventually, an auxiliary tool in a variety of different tasks like the ontology integration or the ontology alignment. To the best of our knowledge, there is no research focused on using WordNet to “fill in the blanks” within ontologies and extend their content, which may be defined independently. We claim that incorporating WordNet in our theoretical framework (introduced in Section 2) may highly influence the quality and expressivity of maintained ontologies, which in result may impact the quality of results of different tasks related to ontology applications.

V. SEMI-AUTOMATIC METHOD OF ESTABLISHING ATTRIBUTE SEMANTICS

As the motivating example has demonstrated, the manual definition of attribute semantics is error-prone and time-consuming. Semantics definition requires conciseness, completeness, and consistency, both internal – among definitions of attributes in ontologies to be merged, and external – among definitions of attributes and the real world. These goals can be at least partially addressed by a semi-automatic method to establish attribute semantic. Such a method, to be effective, has to fulfill several critical assumptions:

- It should minimize required expert involvement but be parametrized if possible.
- It should deliver objective (repeatable) semantics of attributes when possible.

It should allow integration of any number of ontologies with the same results unless the choices of experts are different; the order of integration should influence neither the final integration results nor the calculated knowledge increase metric.

A. WordNet DESCRIPTION

The semantics of an attribute can be defined by its reference to WordNet [7]. WordNet is an online lexical database, which keeps words (grouped into categories) together with their meanings [19] in the so-called lexical matrix. It happens that the same representations (word forms) have different meanings and that the same meaning is represented by different words (synonyms). Among the existing categories, only nouns are considered in the paper, as it is assumed that names of attributes are represented in that way. The synonym sets are called synsets, and they are used to represent meanings (each synset has a short definition, and usage examples). WordNet defines a mapping between written words and synsets. Synsets are connected with the use of semantic relations, e.g., for nouns [19]:

- **Hypernymy** – a semantic relation between a more specific and more general concept, e.g., {maple} is a hypernym of {tree}. It is a transitive and asymmetrical relation, which forms an acyclic graph. According to [19]: “Hypernym inherits all the features of the more generic concept and adds at least one feature that distinguishes it from its superordinate and from any other hypernyms of that superordinate.” {entity} is the root parent (direct or indirect) of every other concept. The inheritance hierarchy rarely exceeds a dozen levels [20], and it will be used for the representation of attribute semantics.

- **Hyponymy** – a semantic relation between a more general and more specific concept, opposite to hypernymy, e.g., {tree} is a hyponym of {maple}.

B. METHOD DESCRIPTION

This section presents a method used for ontology integration at the concept level, which uses WordNet as a knowledge base.
1) METHOD OVERVIEW
The integration method (at the concept level) consists of 3 main stages presented in Fig. 1 with the use of different colors:

1) Preliminary stage – in which an expert selects ontologies to be integrated (action 1) and mappings between ontologies’ concepts and/or their attributes (action 2).

2) Semantics definition stage – in which an expert decides about the expected support level (action 3) and next she or he actively takes part in attribute semantics definition (action 4); this involves communication with WordNet tools.

3) Integration stage – in which the tool – from previously defined semantics – iteratively integrates ontologies (action 5); the final result together with the knowledge increase metric is presented to the user (action 6) who can translate it to OWL format and save it for further processing (action 7).

Actions 3-5 are described in detail in the following sub-chapters.

2) DEFINITION OF ATTRIBUTE SEMANTICS
This subsection answers the question of how WordNet helps in establishing attributes semantics, and in what way this process is influenced by the support level selected by a domain expert. Two levels of support are offered: high and low. High support means minimal expert involvement; the tool determines automatically attributes semantics in every place it is possible. Low support means that the expert has better control over the semantics definition. In both cases, the goal is the same – to map an attribute to a specific WordNet synset or to tell that such mapping does not exist or is not precise enough.

Let’s assume someone wants to define the semantics of last_name attribute in the Person concept. This term is defined in WordNet as a noun (surname.n.01), and it has only one definition [29]: “the name used to identify the members of a family (as distinguished from each member’s given name).” Its synonyms are: family name, cognomen, and surname. In the case of high support, the first meaning (surname.n.01) will be automatically assigned to any attribute named: last_name, family_name, cognomen, surname or to an attribute which is mapped from another ontology to such attribute. In the case of low support, an expert will be shown the definition first, and he can decide whether to use it or not. One can try to find out in WordNet another term with definition better reflecting the original attribute meaning. If this is impossible, the assignment to the WordNet wouldn’t be defined.

If there are many mappings of an attribute name to a WordNet synset, the expert is shown all the definitions to select one of them. For example, the name word has six potential noun meanings in WordNet (1: “a language unit by which a person or thing is known”, 2: “a person’s reputation”, 3: “family based on male descent”, 4: “a well-known or notable person”, 5: “by the sanction or authority of”, 6: “a defamatory or abusive word or phrase”). In the case of high support, the expert should select one meaning or tell none of the presented definitions is valid, and – in the case of low support, the expert is also allowed to find out another word in WordNet, which will be used later for integration purposes.

If there is no meaning of an attribute name in WordNet, independently of the selected type of support an expert can find another word, which will be used for integration purposes or to tell the tool the mapping doesn’t exist. For example, there is no direct mapping of the attribute birthdate, however, there are two for birthday: 1: “an anniversary of the day on which a person was born (or the celebration of it)”, 2: “the date on which a person was born”. In the considered motivating example, the second one was selected.

Mapping attribute names to WordNet synsets is the first step in determining attribute semantics. The second one is providing values of the $S_A$ function for particular attributes. Implementation of this function in the proposed approach is context dependent – the explanation why, along with some examples, is given in the next subsection.

3) ONTOLOGY INTEGRATION PROCESS
The proposed method is prepared for the integration of any number of ontologies: $O_1, \ldots, O_n$ for $n \geq 1$. The process is performed iteratively according to (12):

$\emptyset \oplus O_1 = O_1^* = O_1$

$O_{i-1}^* \oplus O_i = O_i^*$ for $i \geq 2$ (12)

where:

- $\emptyset$ – an empty ontology,
- $\oplus$ – merge operator,
- $O^*$ – integration result.

It is assumed that the first ontology is integrated with an empty ontology. Therefore, it is the first integration result. As it was proven in [11], the ordering of ontologies to be integrated does not influence the integration result.

The integration takes place at the concept level. It is assumed that either a direct mapping between ontologies’ concepts or their attributes is given. For example, the mapping can define that the Person concept from $O_1$ is equivalent to the Person concept from $O_2$ (see Motivating example). The integration algorithm at the concept level, being a slight modification of the one presented in [11], [21], is given below.

The algorithm refers to the WordNet relations between synsets, i.e., hyponymy and synonymy. If the attribute $b$ is a child of another attribute $a$, it is removed from the resulting set (the more general attribute is left), and its domain – if necessary – is added to the domain of its parent. This enables to represent all instances of both attributes $a$ and $b$ even if their domains are different. It should be noted that the relations are checked between synsets assigned to attributes (represented by the function synset) not directly to the attributes. Let demonstrate it by an example. Assume that the name attribute has assigned the synset name.n.01, and the surname
Algorithm 1 Integration at a Concept Level

Input: \((A^1, V^1), (A^1, V^1)\) where \(A^1, A^1\) – the attribute set of concept \(C_i\), and \(C_i\) respectively, \(V^1, V^1\) – the set of attributes domains; concept \(C_i\) is mapped to concept \(C_j\)

Output: \((A^*, V^*)\) – attributes and their domains of the merged concept \(C^*(integration of \(C_i\) and \(C_j\))

begin
\[ A^* \rightarrow A^1 \cup A^1 \]
for \((a, b) \in A^* \times A^*(a \neq b)\) do

if synset\((a) \leftarrow hypersyn \leftarrow synset(b)\)

or synset\((a) \leftarrow synonymy \leftarrow synset(b)\) then

\[ A^* = A^* \setminus \{b\} \]
\[ V^*_a = V_a \cup V_b \]
end
end

attribute – the synset surname.n.01. In WordNet, the first synset is a hypernym of the second one, and that is why the surname attribute will be removed. The same operation is performed for synonyms – only one is left as the integration result.

To calculate the increase knowledge metric, the function \(S_A\) must be defined for each attribute (see Algorithm 2).

The algorithm assumes the existence of some auxiliary functions. The first is unique_id which returns a globally unique identifier for a specific attribute \(a\), e.g., unique_id(end_date) could return the 3#end_date value when 3 is the ordering number of ontology the attribute comes from. Therefore, \(S_A(\text{end_date}) = \text{entity.n.01} \land 3\#\text{end_date}\). The next auxiliary function is shortest_path, which returns the synsets’ names (connected with and operator) between two function arguments laying on the shortest path between them according to hypernymy relation. For example shortest_path(entity.n.01, name.n.01) = entity.n.01 \land abstraction.n.06 \land relation.n.01 \land part.n.01 \land language_unit.n.01 \land name.n.01.

The function is calculated independently for each integration step (iteration), and the way it is built for a particular attribute depends on the context, i.e., the ontologies merged (the left and right side of merge operator). The integration process is performed according to Algorithm 1, with no use of the results of the \(S_A\) function. However, the function which keeps its original semantics could be applied for the ontology integration as it was defined in [11], giving the same results. Obviously, we assumed that all attribute semantics are defined as mappings to WordNet synsets.

Theorem: It should be shown that for any two attributes \(a, b\) it holds: \(S_A(a, A) \leftrightarrow S_A(b, B)\) if attributes are synonyms, and \(S_A(b, B) \rightarrow S_A(a, A)\) if the attribute \(a\) is more general than \(b\). Otherwise none of these is true.

Proof: Let assume that attributes are synonyms. According to Algorithm 2, the same path will be generated for both attributes – it will be the shortest path between entity.n.01 and the synset representing the attribute’s semantics. It is worth mentioning that the NLTK library, for different lemmas (e.g. last_name, family_name), always returns the same synset (here: surname.n.01). So, the formula for the attribute \(a\) will be the same as for the attribute \(b\).

Now, let us assume that the attribute \(a\) is more general than the attribute \(b\). The formula generated for the child \(b\) will be a concatenation of two paths: the shortest path from entity.n.01 and the parent \(a\), and the shortest path between the parent \(a\) and the child \(b\). The formula generated for the parent \(a\) will contain all the elements for the child extended with a unique identifier. This unique identifier is necessary to distinguish the case when the parent (attribute \(a\)) has only one child (attribute \(b\)). So, if the formula for \(b\) holds then its extension (formula for \(a\)) also holds.

If there are no hypernymy or synonymy relation between any two attributes in the merging classes, but the attribute’s semantics is established, the attribute is represented as the shortest path between its synset and entity.n.01. Such a path must be unique. Similarly, when the attribute wasn’t mapped to any WordNet synset, its semantics is represented by the formula: entity.n.01 and a unique identifier. These unique identifiers are free variables in \(S_A(a, A) \leftrightarrow S_A(b, B)\) and \(S_A(b, B) \rightarrow S_A(a, A)\), which causes that they can’t be always true.

Algorithm 2 Calculation of \(S_A\) function

Input: \(A^1, A^1\) – the attribute set of concept \(C_i\), and \(C_j\) respectively; concept \(C_i\) is mapped to concept \(C_j\)

Output: \(S_A\) – the set of logical sentences representing attributes semantics for the knowledge increase metric calculation

begin
\[ S_A \leftarrow \emptyset \]
for \(a \in A^1 \cup A^1\) do

if synset\((a) = \text{null}\) then

\[ S_A = S_A \cup \{\text{entity.n.01} \land \text{unique_id}(a)\} \]
else if \(\exists b \in A^1 \cup A^1\): synset\((a) \leftarrow \text{hyperonymy} \leftarrow synset(b)\) then

\[ S_A = S_A \cup \{\text{shortest_path(entity.n.01, synset(a))}\} \]
else if \(\exists b \in A^1 \cup A^1\): synset\((a) \leftarrow \text{synonymy} \leftarrow synset(b)\) then

\[ S_A = S_A \cup \{\text{shortest_path(entity.n.01, synset(a))}\} \]
end
end
C. METHOD IMPLEMENTATION

The proposed method was implemented in a prototype tool. The tool is a Java application with a Graphical User Interface. It reads OWL2 ontologies to be integrated (the owlapi library is used to parse OWL files, http://owlapi.sourceforge.net/) and files with mapping definitions – see Fig. 2. The runnable version of the tool can be downloaded from [32].

The tool communicates via REST API with a set of web services written in Python. These services manipulate WordNet with the use of the NLTK library (http://www.nltk.org).

Fig. 3 shows an example of service application – meanings of *age* word limited to nouns are displayed, from which an expert has to select one.

An expert is allowed to find out the matching synset in WordNet by himself. Such example is shown in Fig. 4 where the meaning for term *birthdate* is established.

The integration results are presented in a textual manner (see Fig. 5), however, the integrated ontology can also be saved as an OWL2 file. The result’s format is described in the next subsection.

One of the problems to be solved was the definition of two-directional mappings between OWL2 syntax and internal definition of ontology (see Section 2). For example, a domain for a data property can be defined as a union of two or more classes in an OWL ontology. In such a case the attribute is copied to be a member of each concept separately, however, its semantics is defined only once (and is the same for all copies). As the attribute can disappear in some classes after merging, the translation of internal ontology representation to OWL2 does not use union axiom. Instead, unique version numbers are added to the attribute names, and the attributes are stored as separate elements. A similar mechanism is used to distinguish classes/attributes with the same name defined in two different integrated OWL ontologies, which are not mapped to each other (e.g. ontology 1 introduces *Person* class, and ontology 2 does the same). The resulting ontology will contain two *Person* classes, with different prefixes, e.g. 1_Person, 2_Person.

Translation to OWL works correctly for classes and data properties; the other elements are neglected at that moment.

D. METHOD APPLICATION EXAMPLE

The motivating example presented in Section 3 will be used to demonstrate the way the integration method works. The ontologies to be integrated (*Person_1.owl, Person_2.owl,* see
Fig. 2) are identified by two subsequent numbers: 1 and 2. The first ontology is integrated with an empty one, giving the ontology 3 (merged_0.owl) as a result, which is further integrated with the ontology 2, giving the ontology 4 (merged_1.owl). The final integration result is presented in Fig. 5.

Each class/attribute has its traces to the elements in the merged ontologies. Attributes also have information about the selected (by an expert or automatically) synset in WordNet, e.g. name.n.01, and additional Boolean value describing if the selected meaning fits the expert intention. The expert had to select meanings for two attributes (age, name), and to find out the meaning for one (birthdate). The obtained integrated ontology is the same as for Expert 3 in Table 1.

Finally, the knowledge increase metric is calculated – here 0.53. This value – compared to the manual integration – lies in the middle; it is about 20% higher than the one obtained by the Expert 1, and about 20% less than the one obtained by Experts 2 and 3.

**VI. EXPERIMENT DESCRIPTION**

**A. GOALS**

A controlled experiment was conducted to check the usefulness and applicability potential of the proposed method. It aimed to answer three research questions:

1) Is the integration result obtained using the proposed method similar to the result obtained by a manual approach? Is the distance between integration results acceptable?

2) Is the proposed method more efficient than the manual one?

3) Is the definition of attribute semantics obtained using the proposed method more reusable than in the case of the manual approach?

To answer the first question, a simple measure \( Sim \) to calculate ontologies similarity is used – see (13). It is based on Petrikas at al. S metrics [15], in which A, and B originally represents either synsets or term description sets. The \( Sim \) metric returns values from 0 (two ontologies have nothing in common at the level of attributes) to 1 (two ontologies are identical).

\[
Sim(O_1, O_2) = \frac{\sum_{<s,t> \in map(O_1, O_2)} S(s, t)}{|map(O_1, O_2)|}
\]

\[
S(s, t) = \max_{A \cap B} \frac{A \cap B}{A \cup B}
\]  

where:

- \( map(O_1, O_2) \) – the set of tuples \(<s, t>\) where \( s, t \) are classes in mapped ontologies \( O_1, O_2 \)
- \( A, B \) – the set of attributes in class \( s \), and \( t \), respectively.

It is assumed that the distance between ontologies is acceptable if it is less or equal to 0.2 – in other words, their similarity is above 80%. This assumption is consistent with Pareto’s rule.

### Table 2. Experiment plan.

| Nr | Merged ontologies | Starts with | Nr of concepts in \( O^* \) | Nr of mapped classes with attributes /Nr of mapped classes | Nr of attributes in mapped classes |
|----|-------------------|-------------|----------------------------|-------------------------------------------------|----------------------------------|
| 1  | (Edas, Sigkdd)    | M           | 143                        | 2/11                                            | 21 + 11 =                         |
| 2  | (Edas, ConfTool)  | A           | 129                        | 6/14                                            | 21 + 32 =                         |
| 3  | (Edas, ConfTool)  | A           | 150                        | 5/14                                            | 21 + 22 =                         |
| 4  | ConfTool          |             | 83                         | 2/6                                             | 11 + 32 =                         |
| 5  | ConfTool          |             | 98                         | 3/12                                            | 11 + 22 =                         |
| 6  | Sofsem            |             | 88                         | 6/11                                            | 32 + 22 =                         |

The auxiliary \( S \) function (13) is also used in question 3 to access the similarity in the definition of attribute semantics given by various experts.

**B. MATERIALS**

The experiment was performed on ontologies taken from the OAEI set [30], for which also reference alignments are defined [31]. The selected ontologies belong to the conference track. They are \{Edas, Sigkdd, ConfTool, and Sofsem\}. From the alignments, irrelevant mappings (not referring to classes or data attributes) were removed (see [32]).

**C. PLAN**

The experiment aimed to define attribute semantics defined in separate ontologies with the purpose of their integration. The definition was either supported by the developed tool, which refers to Wordnet, or done manually with the use of first-order logic formulas. The experiment was split into 12 mini-experiments, 2 for each possible pair of ontologies from the set \{Edas, Sigkdd, ConfTool, and Sofsem\}. The difference between them was in the ordering of the methods used – the first started with the manual (M) and the second with the automatic approach (A), see Table 2.

Each mini-experiment was run under the supervision of one of the authors whose responsibility was to give a proper introduction and take measurements. The supervisor measured times (in minutes) necessary to assign the semantics to the attributes both manually and with the implemented tool, and to collect the results.

**D. PARTICIPANTS**

The experiment plan consisted of 12 mini-experiments, each one conducted by another person. Most of the experiment participants (10) were the faculty employees of different ages (24-65, average about 40) with strong mathematical foundations, however not necessarily knowledge with the concept of ontologies and their integration (3 identified as novice, 2 – advanced beginners, 4 – competent, 3 – proficient). The
last two people are working outside the university but are connected with IT. All are familiar with object-oriented programming, and that metaphor was used to explain the concept of ontology if required.

### E. DATA ANALYSIS

The data collected during the experiment include: the time of semantics definition performed manually (M) and automatically (A), as well as the values of “growth of knowledge” metric (Δ) for both approaches (M, A), and the integrated ontologies similarity Sim (13). The results are presented in Table 3 and all of the experiment files are available in [32].

Experiment participants introduced a few inconsistencies in the attribute semantics independently of the method used. It happened that many attributes in the same class were given exactly the same definition (what made them indistinguishable). Another problem, mostly in the automatic approach, was a different semantic definition for attributes that obviously represent the same phenomenon. We also observed that the interface of the implemented tool could be more user-friendly, e.g., offering a kind of context help. Without it, the experts, after a few attempts of finding the proper meaning, gave up.

Sometimes, the subjects were confused with the real meaning of attributes, e.g. Edas ontology introduced two dateTime attributes for ConferenceEvent, namely, hasStartDateTime, hasEndDateTime, whereas ConfTool ontology two string attributes for Event (mapped to ConferenceEvent) entitled ends_on, starts_on. Some experts decided that, probably because of different types, the attributes have nothing in common, when others deduced that they represent the same thing.

Additionally, similarities between the attribute semantics definitions, made by different experts (E₁, E₂) for the same pair of ontologies (O₁, E₁), (O₂, E₂), (O₂, E₁), (O₂, E₂) were calculated (13) – see Table 4.

The data allow the following hypotheses to be made:

1. Time for the manual definition of attribute semantics is significantly longer than time for automatic definition.
2. The increase of knowledge measure (Δ) is independent of the method used.
3. The similarity of the integrated ontologies prepared manually and semi-automatically is at the acceptable level (above 90%).
4. The automatic approach produces a more reusable definition of attributes’ semantics.

The first hypothesis is directly confirmed in Table 3. For each expert, time spent on manual semantics definition is longer than for automatic approach.

For the next analysis, we assumed that the significance level is equal to 0.05. First, we checked how the method (manual, semi-automatic) affects the value of the knowledge increase measure. The Shapiro-Wilk test confirmed that samples do not come from the normal distribution. We obtained statistical values equal to 0.773 (for p-value = 0.0047) and 0.841 (for p-value = 0.029) for manual and automatic samples, respectively. Then, the result of the non-parametric Wilcoxon test (T-statistic equal 14.5 and p-value equal 0.74) allows us to conclude that the method does not affect the value of the knowledge increase measure.

The integrated ontologies similarity is bigger than 96% (the Wilcoxon statistic value for hypothetical median 0.96 was equal to 13 and p-value 0.042, the positive rank sum was bigger than negative rank sum).

The last hypothesis is also directly confirmed by the data from Table 4. As one can easily observe, the probability that two experts define the same attribute semantics in the same way is very low. Additionally, for the automatic approach, we checked the average similarity of attribute semantics produced by different experts.

Since the sample does not come from the normal distribution (Shapiro-Wilk test value equal 0.83 and p-value 0.019), we calculated by using the Wilcoxon test that the hypothetical median is equal to 0.68. The analysis showed that the similarity between the semantics definition given by different experts is at least 68%. It demonstrated that the automatic approach produces a good quality definition of attribute semantics that other experts can reuse.

### F. CONCLUSIONS

The experiment results proved the usability of the proposed semi-automatic approach to attribute semantics definition. The method allows determining definitions in a shorter time than the fully manual procedure. Additionally, semantics produced by the automatic approach can be reused by other users.
(the accordance of definitions produced by two experts was at least 68%). In a situation where semantics were defined manually, we have obtained always different answers. Each expert differently understood a particular attribute and proposed a solution with his own intuition. Moreover, the choice of automatic or manual procedure for attribute semantics definition does not statistically influence the ontology integration results.

G. THREATS TO VALIDITY

To assure construct validity, we reused the OAEI set of ontologies [30] with reference alignments [31]. The problem we encountered, but couldn’t address, was a relatively small number of concepts mapped between ontologies. The available alignments contain only from 6 to 14 mappings, which is about 10% of the total number of concepts in the integrated ontology. The number of concepts with attributes was even smaller - from 2 to 6, which is from 1% to 4% of the total number of concepts in the output ontology. In consequence, the Δ (increase knowledge metrics) is almost the same independently of the approach used. Moreover, attributes in merged ontologies, in most cases, represent entirely different features, so only in a few cases, some of them were removed from the result. It means that the chance to obtain a different set of attributes for mapped concepts is rather low.

We identified the following threats to internal validity:

- Potential low quality of semantics definition. We tried to mitigate the risk by involving mostly university staff members, aware of the importance of experiment organization. We can conclude that the maturity of experts in this area was similar.
- The same pair of ontologies was a subject of two mini-experiments performed by two different participants. These two mini-experiments differed in the ordering of the method used first for attributes’ semantics definition to reduce the learning effect.

We identified one threat to the external validity. The actual training and the background knowledge of the subjects could influence the experiment results. We believe that the level of expertise of the participants did not affect anything other than the time used to solve tasks.

In all cases, manual definition of attribute semantics took longer, despite the actual experience of the person in using ontologies and WordNet. However, we believe that even a very brief training with Wordnet and a better user interface of the implemented tool may result in better effectiveness of the automatic approach.

VII. SUMMARY

Ontologies in recent years are widely used not only for storing data but also for processing their content or reasoning about it. The popularity of ontologies becomes both a motivation and a challenge for ontology developers to provide the most useful and correct products. However, the process of creating an ontology by human experts is highly vulnerable to errors. Thus, the motivation of our work is to reduce the human participation through support in this process.

This paper was devoted to the problem of assigning semantics to attributes within concepts. To the best of our knowledge there is no research addressing the mentioned problem. We have proposed an easy to apply approach utilizing WordNet and incorporated it into our framework for the ontology integration process and estimation of the potential growth of knowledge.

The experiments confirmed the usefulness of the proposed approach to attribute semantics for ontology integration. The WordNet hierarchy of synsets is consistent and reflects faithfully the reality. Therefore, the mapping of attributes to synsets can be used instead of manually defined formulas. Such an approach is less error-prone and less sensitive to a different understanding of the domain (semantics of attributes defined by one expert is likely to be reused by another expert without any changes). However, the solution is not ideal since WordNet lacks some specific words. In these cases, a mixed approach seems to be a cure. The semantics (represented, e.g., by logical formulas) for new terms can be stored in a catalog, extended on demand, from which the proper meaning in specific context could be selected. Alternatively, one can locally extend WordNet to represent a specific domain, and store new terms together with their definitions, synonyms, and relations to existing synsets.

In the future, we plan to support the tool with such a catalog for gain the ability to perform quick search and to implement integration rules at the relation and instance levels. That should result in a mature enough solution for practical applications. Furthermore, we would like to prepare a tool for creating, storing, and processing ontologies which supports or even replaces human experts in each mentioned task. This requires to relate our ontology model with OWL and to solve the problem of semantics definition at the relational level. The implemented framework for ontology integration should be further verified using benchmark datasets provided by OAEI. We also plan to address an idea of extending the presented approach to concept’s meta data expressed in, for example, rdfs:label and rdfs:description tags in OWL, and incorporate them in the process of designating concepts’ semantics.

REFERENCES

[1] B. Alhassan, S. B. Junaidu, and A. A. Obiniyi, “Extending an ontology alignment system with a lexical database,” Sci. Res. J., vol. 3, no. 1, pp. 12–17, 2015.
[2] S. Amirian and S. Mohammadi, “Ontology alignment using WordNet method,” Int. J. Comput. Sci. Netw. Secur., vol. 17, no. 7, pp. 161–167, 2017.
[3] T. Boiński and H. Krawczyk, “Dynamic OWL ontology matching using lexical Wordnet-based measures,” Stud. Proc. Polish Assoc. Knowl. Manage., vol. 60, pp. 4–15, Apr. 2012.
[4] M. Cristani and R. Cuel, “A survey on ontology creation methodologies,” Int. J. Semantic Web Inf. Syst., vol. 1, no. 2, pp. 49–69, Apr. 2005.
[5] D. Dou, D. McDermott, and F. Qi, “Ontology translation on the semantic Web,” in Journal on Data Semantics II, S. Spaccapietra, E. Bertino, S. Jajodia, R. King, and D. McLeod, eds. Berlin, Germany: Springer, 2005, pp. 35–57.
[1] Z. Dragisic, V. Ivanova, P. Lambrix, D. Faria, E. Jiménez-Ruiz, and C. Pesquita, “User validation in ontology alignment,” in Proc. Int. Semantic Web Conf., P. Groth, Ed. Cham, Switzerland: Springer, 2016, pp. 200–207.

[2] C. Fellbaum and K. Brown, “WordNet and wordnets,” in Encyclopedia of Language and Linguistics, K. Brown, Ed. 2nd ed. Oxford, U.K.: Elsevier, 2005, pp. 665–670.

[3] D. F. Huyhn, R. C. Miller, and D. R. Karger, “Potluck: Data mash-up tool for casual users,” in The Semantic Web K. Aberer, Ed. Berlin, Germany: Springer, 2007, pp. 239–252.

[4] Y. R. Jean-Mary, E. P. Shironshtia, and M. R. Kabuka, “Ontology matching with semantic verification,” J. Web Semantic, vol. 7, no. 3, pp. 233–251, Sep. 2009.

[5] P. Kolyvakis, A. Kalousis, and D. Kiriatis, “DeepAlignment: Unsupervised ontology matching with refined word vectors,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Long Papers), vol. 1, 2018, pp. 787–798.

[6] A. Kozierkiewicz and M. Pietranik, “The knowledge increase estimation framework for ontology integration on the concept level,” J. Intell. Fuzzy Syst., vol. 32, no. 2, pp. 1161–1172, Jan. 2017.

[7] A. A. Krizhanovsky and F. Lin, “Related terms search based on WordNet and wiktionary and its application in ontology matching,” 2009, arXiv:0907.2209. [Online]. Available: http://arxiv.org/abs/0907.2209

[8] J. Kwak and H.-S. Yong, “Ontology matching based on hypernym, hyponym, holonym, and meronym sets in WordNet,” Int. J. Web Semantic Technol., vol. 1, no. 2, pp. 1–14, Apr. 2010.

[9] P. Lambrix and H. Tan, “SAMBO—A system for aligning and merging biomedical ontologies,” J. Web Semantic, vol. 4, no. 3, pp. 196–206, 2006.

[10] F. Lin and K. Sandkuhl, “A survey of exploiting wordnet in ontology matching,” in Artificial Intelligence in Theory and Practice II, vol. 276, D. L. McGuinness, Ed. Berlin, Heidelberg, Germany: Springer, 2004, pp. 341–350.

[11] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, “An environment for merging and testing large ontologies,” in Proc. 7th Int. Conf. Princ. Knowl. Represent. Reasoning, A. G. Cohn, F. Giunchiglia, and B. Selman, eds. San Francisco, CA, USA: Morgan Kaufmann, 2000, pp. 483–493.

[12] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, “The chimera ontology environment,” in Proc. AAAI/IAAI, 2000, pp. 1123–1124.

[13] I. Meng, R. Huang, and J. Gu, “Measuring semantic similarity of word pairs using path and information content,” Int. J. Future Gener. Commun. Netw., vol. 7, no. 3, pp. 183–194, Jun. 2014.

[14] G. A. Miller, R. Beckwith, C. Fellbaum, K. Brown, and F. Lin, “Introduction to WordNet: An on-line lexical database,” Int. J. Lexicography, vol. 3, no. 4, pp. 235–244, 1990, doi: 10.1093/ijl/3.4.235.

[15] G. A. Miller, “Nouns in WordNet: A lexical inheritance system,” Int. J. Lexicography, vol. 3, no. 4, pp. 245–264, 1990, doi: 10.1093/ijl/3.4.245.

[16] N. T. Nguyen, Advanced Methods for Inconsistent Knowledge Management. Springer, 2008.

[17] N. F. Noy and M. A. Musen, “SMART: Automated support for ontology merging and alignment,” in Proc. 12th Workshop Knowl. Acquisition, Modelling, (KAW), Oct. 1999, pp. 1–24.

[18] N. F. Noy and D. L. McGuinness (2001), Ontology Development 101: A Guide to Creating Your First Ontology. Knowledge Systems Laboratory, Stanford Univ., Stanford, CA, USA. Accessed: Jun. 8, 2019, [Online]. Available: http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mguinness.html

[19] N. F. Noy and M. A. Musen, “The PROMPT suite: Interactive tools for ontology merging and mapping,” Int. J. Hum. Comput. Stud., vol. 59, no. 6, pp. 983–1024, Dec. 2003, doi: 10.1016/j.jhcs.2003.08.002.

[20] M. Pietranik and N. T. Nguyen, “A multi-attribute based framework for ontology aligning,” Neurocomputing, vol. 146, pp. 276–290, Dec. 2014, doi: 10.1016/j.neucom.2014.03.067.

[21] F. C. Schadd and N. Roos, “Coupling of wordnet entries for ontology mapping using virtual documents,” in Proc. 7th Int. Conf. Ontology Matching, vol. 946, 2012, pp. 25–36.

[22] U. Thayasivam and P. Doshi, “On the utility of WordNet for ontology alignment: Is it really worth it?” in Proc. IEEE 5th Int. Conf. Semantic Comput., Sep. 2011, pp. 267–274.

[23] M. Yatskevich and F. Giunchiglia, “Element level semantic matching using WordNet,” in Proc. Meaning Coordination Negotiation Workshop at ISWC, 2004, pp. 37–48.

[24] WordNet. Accessed: Jun. 8, 2019. [Online]. Available: http://wordnetweb.princeton.edu/perl/webwn

[25] [DataSet] Ontology Alignment Evaluation Initiative OAEI-2015 Campaign. Conference Track. Accessed: Jun. 8, 2019. [Online]. Available: http://oaei.ontologymatching.org/2015/conference/index.html

[26] [DataSet] Ontology Alignment Evaluation Initiative OAEI-2015 Campaign. (2015). Results of Evaluation for the Conference Track Within OAEI. Crisp reference alignments (reference-alignment.zip). Accessed: Jun. 8, 2019. [Online]. Available: http://oaei.ontologymatching.org/2015/conference/eval.html#crisp-ra

[27] Ontology Integration Framework Source Code and Experimental Data. [Online]. Available: https://github.com/bhnatkowska/OntologyIntegrationFramework

BOGUMIŁA HNATKOWSKA received the M.Sc. and Ph.D. degrees in computer science from the Wrocław University of Science and Technology, Poland, in 1992 and 1997, respectively. Her Ph.D. dissertation was associated with the use of formal methods in software engineering. Since 1998, she has been working as an Assistant Professor. She has over 100 publications in international journals and conference proceedings from different areas of software engineering. Her main scientific interests include but are not limited to software development processes, modeling languages, model driven development (also with the use of ontologies), model transformations, and quality of the software products. She is a member of program committees of several international conferences.

ADRIANNA KOZIERKIEWICZ received the M.Sc. and Ph.D. degrees in computer science and the Ph.D. degree in 2007 and 2011, respectively. She is currently an Assistant Professor at the Wrocław University of Science and Technology. Her research interests include ontologies, knowledge management, recommendation systems, consensus theory, and other artificial intelligence applications. She is an author or coauthor of over 55 publications in prestigious journals and proceedings of international conferences. She is also an Editor of a book titled “Modern Approaches for Intelligent Information and Database Systems.” She has also been the chair of several conferences, and a member of program committee of many conferences and review boards of some journals. For her research projects, she has been awarded with a few scholarships and grants. She is a Managing Editor of International Journal of Intelligent Information and Database Systems.

MARCI N PIETRANIK received the M.Sc. and Ph.D. degrees in computer science from the Wrocław University of Science and Technology, in 2008 and 2014, respectively. Since 2016, he has been an Assistant Professor with the Wroclaw University of Science and Technology. His scientific interests include knowledge integration and topics related to ontology management (especially ontology evolution and ontology alignment). He has authored or coauthored over 40 articles and co-organized several conferences. He also has a strong experience in modern web technologies used within medical projects.