# A Validation Technique for Integrated Ontologies

Neli P. Zlatareva

*Abstract*—Ontology validation is an important part of web applications' development, where knowledge integration and ontological reasoning play a fundamental role. It aims to ensure the consistency and correctness of ontological knowledge and to guarantee that ontological reasoning is carried out in a meaningful way. Existing approaches to ontology validation address more or less specific validation issues, but the overall process of validating web ontologies has not been formally established yet. As the size and the number of web ontologies continue to grow, more web applications' developers will rely on the existing repository of ontologies rather than develop ontologies from scratch. If an application utilizes multiple independently created ontologies, their consistency must be validated and eventually adjusted to ensure proper interoperability between them. This paper presents a validation technique intended to test the consistency of independent ontologies utilized by a common application.

*Keywords*—Knowledge engineering, ontological reasoning, ontology validation, semantic web.

## I. INTRODUCTION

ONTOLOGY validation is an important part of web applications' development, where knowledge integration and ontological reasoning play a fundamental role. It is intended to ensure the consistency and correctness of ontological knowledge, as well as to guarantee that ontological reasoning is carried out in a meaningful way. Existing research in ontology validation addresses more or less specific issues primarily concerned with detecting and managing inconsistencies in single ontologies [1], [2], although increasing attention was paid recently to ensuring the interoperability of aligned and merged ontologies [3]-[5]. Testing independent knowledge sources for interoperability and consistency involves two additional tasks compared to validation activities involved in single ontology development and maintenance:

i. Ontology matching, i.e. finding terminological and structural correspondences between cooperating ontologies, and

ii. Ontology integration, i.e. ensuring that the aggregated ontological knowledge is semantically coherent and thus can be utilized in a meaningful way by a common application.

In this paper, we are concerned primarily with the ontology integration task. We present a validation technique which takes two or more ontologies as an input, converts them into a common representation, and evaluates the latter for inconsistencies of the following two types:

Neli P. Zlatareva is with the Central Connecticut State University, New Britain, CT 06050, USA (phone: 860 832-2723; fax: 860 832-2711; e-mail: Zlatareva@ccsu.edu).

- Inconsistencies in concept definitions, and
- Inconsistencies in property definitions.

We assume that the application utilizing independently created and validated ontologies cannot modify them, but their alignment is simplified by a common vocabulary or an established dictionary to translate between their vocabularies is suggested. These assumptions partially address one of the difficulties in ontology integration, namely, that automatic merging is unattainable due to conceptualization and explication mismatches between local ontologies [6]. At the same time, as stated in [7] "… research is required to find ways through which different conceptualization mismatches can be detected and resolved in order to give accuracy to the process of mapping and thus verifying the knowledge being shared." We believe that one difficulty towards pinpointing conceptualization mismatches between ontologies is in the precision discrepancy of underlying representations. For example, a concept in one ontology may be defined in general terms, while the same concept in another ontology may be expressed in a more detailed terms. For example, consider the university domain and assume that there are two separate ontologies defining partially overlapping parts of it, STUDENT and STAFF. Also assume that they share a common vocabulary. Consider the *TeachingAssistant* concept. In STAFF ontology, it can be defined as a subclass of *SupportStaff*, while in STUDENT ontology; it can be defined as a subclass of *GradStudent* class with the property *hasAppointmentTA*. Properties associated with *TeachingAssistant* concept in these ontologies might be inconsistent due to the fact that STUDENT ontology defines a specialization of the *TeachingAssistant* concept defined in STAFF ontology. Although the match between the two concepts is trivial, the discrepancy in their definitions may result in an inconsistency which would require the application to either "make sense" of it, or ignore it together with everything that is associated with this concept. The latter is a common strategy in knowledge engineering [8]. To "make sense" of inconsistent concept definitions, the application is supposed to employ some form of paraconsistent or non-monotonic reasoning such as described in [9]-[11]. An alternative approach is presented in [5], where the so-called "bridge rules" enforce the alignment of independent knowledge sources.

The validation technique described in this paper uses the context-dependent reasoning framework presented in [11] to test integrated knowledge for inconsistencies resulting from the integration process. It is assumed that input ontologies were validated on their own; for the application they are "closed" domains, which cannot be modified from the outside. The goal of the validation analysis is to identify incoherencies between independent created ontologies that may affect the

application's problem solving process. The advantages of using an independent inference engine to perform validation analysis, in this case the context-dependent reasoning framework, are:

i. Common representation that input ontologies can be converted to.
ii. Explication of inconsistencies is more easily attainable.
iii. Detected inconsistencies are associated with explicit contexts (or justifications) that help explain the type and the source of each inconsistency.

An extended example is followed throughout the paper to illustrate the proposed validation technique. Section II introduces some basic definitions and the notation used throughout the paper. Section III provides some background work that the proposed validation technique is based on. Section IV presents the conversion between ontology representation and context-dependent rules, and the algorithm for testing integrated ontological knowledge for consistency is presented in Section V.

## II. DEFINITIONS AND NOTATION

**Definition 1**: Ontology, $O$, is a tuple $<Tbox, Abox>$, where:

- The *Tbox* contains:
i. Concepts A1, A2, …, B1, B2 … and their properties a1, a2,…, b1, b2, …;
ii. Relations between concepts (subsumption, equivalence, disjointness);
iii. Relations between concepts and their properties. These can be:
a) *Firm relations*, $A1 \Rightarrow a1.T$ (property a1 holds universally for all instances of A1).
b) *Possible relations*, $A2 \Rightarrow a2$ (property a2 holds for some instances of A2).
iv. Relations between concept properties, which can also be:
a) *Firm*, $a1.T \Rightarrow a2.T$ (whenever property a1 holds, property a2 holds).
b) *Possible* $b1.T \Rightarrow b2$ (property b2 commonly holds when property b1 holds).
- The *Abox* contains:
i. Facts about individual instances, such as $<name>: C$, where C is a concept and $<name>$ is an instance of C or a property of an instance of C.
ii. Relations between individual instances, such as $<name1, name2>: R$, where R is a role and $<name1>$ and $<name2>$ are instances of the same or different concepts.

To illustrate, consider the following example ontology, STUDENT.

- **Concepts**: Person (denoted as S0 for brevity), Student (S1), GradStudent (S2), UndergradSudent (S3), TuitionPayer (S6), ParkingPayer (S8), TeachingAssistant (S13).
- **Concept properties**: takesCourses (s4), hasFreeTuition (s5), hasFreeParking (s7), hasPaymentDue (s9), hasHabitatCampus (s10), hasGPA≥3.5 (s11), has AppointmentTA (s12).

- **Relations between concepts**: $S2 \equiv \neg S3$, $S1 \subseteq S0$ $S2 \subseteq S1$, $S3 \subseteq S1$.
- **Relations between concepts and concept properties**:
- $S1 \Rightarrow s4.T$ (Students take courses)
- $(S0 \Rightarrow s4.T) \cap (S0 \Rightarrow \neg s5) \subseteq S6$ (Tuition payers are people who take courses and are commonly not granted free tuition)
- $(S1 \Rightarrow \neg s7) \subseteq S8$ (Parking payers are students who commonly are not granted free parking)
- $S6 \Rightarrow s9$ (Tuition payers typically have payment due)
- $S8 \Rightarrow s9$ (Parking payers typically have payment due)
- $(S1 \Rightarrow s10) \cap (s10.T \Rightarrow s7.T)$ (Some students leave on campus, and all those who leave on campus are granted free parking)
- $S3 \Rightarrow s10.T$ (All undergraduate students leave on campus)
- $(S1 \Rightarrow s11) \cap (s11.T \Rightarrow s5.T)$ (Some students have GPA $\geq 3.5$, and all those who do get free tuition)
- $(S2 \Rightarrow s12) \subseteq S13$ (Some graduate students have appointments as teaching assistants and those who have such appointments are teaching assistants)
- $S13 \Rightarrow s5.T$ (All teaching assistants are granted free tuition)
- $(S1 \Rightarrow s5) \equiv \neg S6$ (Students who are granted free tuition are not tuition payers)
- $(S1 \Rightarrow s7) \equiv \neg S8$ (Students who are granted free parking are not parking payers)
- $\neg (S6 \cup S8) \Rightarrow \neg s9.T$ (Those who are neither tuition payers, nor parking payers do not have payment due)

It is easy to see that if the *Abox* contains semantically correct instance definitions, this ontology will correctly classify them.

Consider a second ontology, STAFF, which for brevity is described in more concise terms:

- **Concepts**: Person (F0), Staff (F1), Professor (F2), Administrator (F3), TuitionPayer (F6), ParkingPayer (F8), TeachingAssistant (F13), SupportStaff (F14).
- **Concept properties**: takesCourses (f4), hasFreeTuition (f5), hasPaymentDue (f9).
- **Relations between concepts**: $F2 \equiv \neg F3$, $F1 \equiv \neg F14$, $F1 \subseteq F0$, $F2 \subseteq F1$, $F3 \subseteq F1$, $F14 \subseteq F1$, $F13 \subseteq F14$.
- **Relations between concepts and concept properties**:
- $(F1 \Rightarrow f4) \cap (F1 \Rightarrow f5) \subseteq \neg F6$ (Staff who takes courses and is granted free tuition is not tuition payer)
- $F3 \Rightarrow f5.T$ (Administrators get free tuition)
- $(F14 \Rightarrow f4) \subseteq F6$ (Tuition payer is support staff who takes courses)
- $F6 \Rightarrow f9$ (Tuition payers typically have payment due)
- $F8 \Rightarrow f9$ (Parking payers typically have payment due)

Again, it is easy to see that the *Tbox* is a consistent set of axioms as long as the *Abox* contains semantically correct concept and property assertions.

**Definition 2:** Let $O_1 = <Tbox_1, Abox_1>$ and $O_2 = <Tbox_2, Abox_2>$. $O_1$ and $O_2$ are called **fully compatible** iff:

a. Syntactic definitions of the concepts and concept properties defined in their Tboxes are either identical, or match according to some agreed upon dictionary.

b. Transitive closures of $O_1$ and $O_2$ contain only semantically equivalent sets of concepts and concept properties, i.e. derivation paths of those are exactly the same. We shall say that such concepts and concept properties **strongly agree** [12].

**Definition 3:** $O_1$ and $O_2$ are called partially compatible iff:

a. A subset of concepts and concept properties match.

b. Transitive closures of $O_1$ and $O_2$ contain only subsets of concepts that strongly agree.

As stated in [12], if two ontologies are fully or partially compatible, their complete or partial alignment is possible. Following the alignment procedure described there, we can establish the following partial match between example ontologies. F0 ↔ S0, F6 ↔ S6, F8 ↔ S8, F13 ↔ S13, f4 ↔ s4, f5 ↔ s5, and f9 ↔ s9.

**Proposition 1**: Integration of fully compatible semantically consistent ontologies results in a consistent aggregated ontology $O_\Sigma = <Tbox_1 \cup Tbox_2, Abox_1 \cup Abox_2>$.

**Proof.** The correctness of Proposition 1 is trivial and follows from the fact that if a concept definition $A$ is entailed by $Tbox_1$, it must be entailed by $Tbox_1 \cup Tbox_2$.

**Proposition 2**: Integration of partially compatible consistent ontologies is not guaranteed to result in a consistent aggregated ontology $O_\Sigma$. However, the aggregated ontology may still be useful in a specific application setting if inconsistent concept definitions are known in advance and can be accommodated by the application's problem solving process.

It is important to note that in some cases explicating the inconsistencies between independent knowledge sources may be a goal of its own. Consider, for example, two independent knowledge bases representing the same domain. Finding inconsistencies between these alternative domain models may be an important part of the problem solving process intended to find everything that is dependent on detected inconsistencies, or everything that can be derived in spite of them.

In Sections IV and V, we describe a validation technique that aims to identify inconsistencies between two partially compatible ontologies and evaluate their suitability for integration towards solving a common task. But first, we discuss a pre-processing step that is required to convert the input ontologies into a common rule-based representation which is more easily amenable to validation analysis.

## III. CONTEXT-DEPENDENT RULES: AN OVERVIEW

If two ontologies are only partially compatible, integration of their Tboxes may result in an inconsistent set of axioms. There are two types of inconsistencies that can be identified at an application level:

- Inconsistencies in matching concept or property definitions.
- Inconsistencies in matching concept or property relations.

Inconsistencies of the first type can be handled by the application if the contexts associated with inconsistent concept and/or property definitions are known. The second type of inconsistencies is generally unresolvable by the application, because this would require changes in one or both ontologies.

The idea behind the proposed validation technique is similar to operationalization-based verification for knowledge-based systems [13]. It utilizes an independent representation to carry out validation analysis, which serves two purposes:

1. Provides a framework where both ontologies can be expressed in a common format.

2. Provides an independent inference capability that can identify and explain detected inconsistencies between the Tboxes without considering the specific contents of their Aboxes. Note that Aboxes may not be relevant to the specific application context.

Next, we briefly describe the representation format that input ontologies are converted to, and the rules that utilize this format to carry out validation analysis (see [11] for more details).

Consider the following data structure representing a statement, $\alpha$, in terms of the context where it holds:

$$\alpha \; LV: (T_1,...,T_n)(P_1,...,P_m)$$

where:

- $\alpha$ is a concept, a concept property, an instance of a concept, or an instance of a concept property;
- **LV** defines the logical value of $\alpha$. It can be:

a. $T$ (true), which states that $\alpha$ is supported unconditionally by all known evidence associated with it, and

b. $D$ (default), which states that the validity of $\alpha$ is defined with respect to the associated partial evidence;

- $(T_1,...,T_n)(P_1,...,P_m)$ states the evidence associated with $\alpha$, where $T_1,...,T_n$ defines the existing evidence that supports $\alpha$, while $P_1,...,P_m$ defines the potential evidence, that is the one that has not been established yet, but if established would increase the truthfulness of $\alpha$. As more evidence is derived, some $P_i$ will be transferred to the T-set.

Relations between statements are expressed by the following two types of rules:

- *Monotonic rules* of the form $R_i \; (T_1,...,T_n)( ) \rightarrow \alpha : T$, where $R_i$ is a reference to the rule that states "if $T_1,...,T_n$ are true, then $\alpha$ is necessarily true."

- *Default rules* of the form $R_j \; (T_1,...,T_n)(P_1,...,P_m) \rightarrow \alpha : D$, where $R_j$ is a reference to the rule that states "if $T_1,...,T_n$ are true, then $\alpha$ is possibly true; its truth will be further strengthened if additional evidence from the set $P_1,...,P_m$ becomes available in which case the corresponding $P_i$ will become part of the T set resulting in a rule that is a stronger version of the original one.

Because rules' syntax is dependent on the current context (the state of the knowledge) at the time the rule fires, we call such rules *context-dependent*. Also, the evidence upon which $\alpha$ was derived defines the *context* with respect to which the validity of $\alpha$ is determined.

Inference procedure supporting context-dependent reasoning can be used as a meta-framework to reason about interrelations between independent knowledge sources, in this case, ontologies to be evaluated for consistency towards solving a specific application task. This requires input ontologies to be converted into the described representation format.

## IV. CONVERTING ONTOLOGY REPRESENTATION INTO CONTEXT-DEPENDENT RULES

A lot of research has been devoted to establishing the relations between Description Logics and rule-based formalisms [14], [15]. Here we address a special case of this problem, which is concerned with converting some generic Description Logics representation as defined in Section II into context-dependent rules.

The transformations used to carry out this conversion are the following:

- $A \subseteq B$ becomes $(A, R_i)$ ( ) $\rightarrow$ B : T
- $A \equiv B$ becomes $(A, R_i)$ ( ) $\rightarrow$ B : T and $(B, R_j)$ ( ) $\rightarrow$ A : T
- $A \cap C \subseteq B$ becomes $(A, C, R_k)$ ( ) $\rightarrow$ B : T
- $A \Rightarrow a.T$ becomes $(A, R_i)$ ( ) $\rightarrow$ a : T
- $A \Rightarrow a$ becomes $(R_i)$ (A) $\rightarrow$ a : D
- $a \Rightarrow b.T$ becomes $(a, R_i)$ ( ) $\rightarrow$ b : T
- $a \Rightarrow b$ becomes $(R_i)$ (a) $\rightarrow$ b : D
- $(A \Rightarrow b) \cap (b \Rightarrow c.T)$ becomes $(A, b)$ ( ) $\rightarrow$ c : T

Applying these transformations to our example ontologies, STUDENT and STAFF, results in the following context-dependent rule sets:

**STUDENT set:**
$(S1, R1)$ ( ) $\rightarrow$ S0 : T
$(S2, R2)$ ( ) $\rightarrow$ S1 : T
$(S3, R3)$ ( ) $\rightarrow$ S1 : T
$(S1, R4)$ ( ) $\rightarrow$ s4 : T
$(S0, s4, R5)$ $(\neg s5)$ $\rightarrow$ S6 : D
$(S1, R6)$ $(\neg S7)$ $\rightarrow$ S8 : D
$(R7)$ $(S6)$ $\rightarrow$ s9 : D
$(R8)$ $(S8)$ $\rightarrow$ s9 : D
$(S1, s10, R9)$ ( ) $\rightarrow$ s7 : T
$(S3, R10)$ ( ) $\rightarrow$ s10 : T
$(S1, s11, R11)$ ( ) $\rightarrow$ s5 : T
$(S3, s12, R12)$ ( ) $\rightarrow$ S13 : T
$(S13, R13)$ ( ) $\rightarrow$ s5 : T
$(S1, s5, R14)$ ( ) $\rightarrow$ $\neg$S6 : T
$(S1, s7, R15)$ ( ) $\rightarrow$ $\neg$S8 : T
$(\neg S6, \neg S8, R16)$ ( ) $\rightarrow$ $\neg$s9 : T

**STAFF set:**
$(F1, R1^*)$ ( ) $\rightarrow$ F0 : T
$(F2, R2^*)$ ( ) $\rightarrow$ F1 : T
$(F3, R3^*)$ ( ) $\rightarrow$ F1 : T
$(F14, R4^*)$ ( ) $\rightarrow$ F1 : T
$(F13, R5^*)$ ( ) $\rightarrow$ F14 : T
$(F1, R6^*)$ $(f4, f5)$ $\rightarrow$ $\neg$F6 : D
$(F3, R7^*)$ ( ) $\rightarrow$ f5 : T

$(F14, f4, R8^*)$ ( ) $\rightarrow$ F6 : T
$(F1, R9^*)$ ( ) $\rightarrow$ F8 : T
$(R10^*)$ $(F6)$ $\rightarrow$ f9 : D
$(R11^*)$ $(F8)$ $\rightarrow$ f9 : D
$(\neg F6, \neg F8, R12^*)$ ( ) $\rightarrow$ $\neg$f9 : T

Testing the combined set of rules for inconsistencies will reveal the incoherencies between underlying ontologies.

## V. TESTING THE CONSISTENCY OF INTEGRATED REPRESENTATION

Assume that a billing application needs both ontologies, STUDENT and STAFF, to assemble a list of people who have payments due for tuition and/or parking fees. To ensure that integrated knowledge $\{Tbox_1 \cup Tbox_2\}$ allows for a meaningful processing of such a query, consider all inferences implied under the complete set of inputs for both ontologies. Let this set be {S2, S3, s11, s12, F2, F3, F13, f4}. Note also that that there are three special axioms (S2 $\equiv \neg$ S3, F2 $\equiv \neg$ F3, F1 $\equiv \neg$ F14) that have been ignored in the conversion process. These axioms defining restrictions on the instances of classes can be treated as semantic constrains which, if violated, will automatically block certain inferences. For example, inferences involving both S2 and S3 will be disregarded.

The inference procedure performed on the combined context-dependent rule set results in the following transitive closure set, $TC(O_{\Sigma})$.

$TC(O_{\Sigma})$ = {S1: T (S2, R2) ( ), S1: T (S3, R3) ( ),
s4: T (S1, R4) ( ), S0: T (S1, R1) ( ),
S6: D (S0, s4, R5) $(\neg s5)$, S8: D (S1, R6) $(\neg s7)$,
s9: D (R7) (S6), s9: D (R8) (S8), s10: T (S3, R10) ( ) ,
s5: T (S1, s11, R11) ( ), S13: T (S3, s12, R12) ( ),
S5: T (S13, R13) ( ), $\neg$S6: T (S1, s5, R14) ( ),
S7: T (S1, s10, R9) ( ), $\neg$S8: T (S1, s7, R15) ( ),
$\neg$s9: T ($\neg$S6, $\neg$S8, R16) ( ), F1: T (F2, R2*) ( ),
F1: T (F3, R3*) ( ), F14: T (F13, R5*) ( ),
$\neg$F6 : D (F1, R6*) (f4, f5), F5: T (F3, R7*) ( ),
F6: T (F14, F4, R8*) ( ), F8: T (F1, R9*) ( ),
f9: D (R10*) (F6), f9: D (R11*) (F8), F0: T (F1, R1*) ( ),
F1: T (F14, R4*) ( ), $\neg$f9 : T ($\neg$F6, $\neg$F8, R12*) ( ) }

Note that according to the semantics of context-dependent rules, both ontologies are internally consistent, because if there is enough information to derive $\neg$S6, then the evidence for $\neg$S6 overrides the evidence for S6, and thus S6 can be ignored. The same is applicable to the other seemingly inconsistent pairs within STAFF and STUDENT ontologies.

To test the integrated set for consistency between the statements of STUDENT and STAFF ontologies do:

**Step1.** Identify the subsets of matched concepts and concept properties. Include both positive and negative instances in the same subset. Ignore all members of the subset marked as D.

**Step2.** For all subsets containing contradictory statements do:

i. Compute the transitive closure of the statements comprising the support for the contradiction. We call resulting lists grounded explanations.

ii. Identify grounded explanations containing matching concepts or concept properties. If such cannot be found, the two contradictory statements are not agreeable and thus the contradiction can be dismissed. Otherwise, explain the contradiction in terms of the associated evidence.

In our example, the test for consistency results in the following subsets of related statements:

$$\{\neg S6: T\ (S1, s5, R14)\ (\ ),\ F6: T\ (F14, F4, R8^*)\ (\ )\}$$
$$\{F8: T\ (F1, R9^*)\ (\ ),\ \neg S8: T\ (S1, s7, R15)\ (\ )\}$$

This result suggests that there are two possible contradictions: between $\neg S6$ and F6, and between F8 and $\neg S8$. For the former, the following grounded explanations are computed:

- For $\neg S6$: (S2, S13, R2, R13, R14) ( ), (S2, S11, R2, R11, R14) ( ), (S3, S11,R3, R11,R14) ( ), (S3, S12, R3, R12, R13, R14) ( )
- For F6: (F13, f4, R5*, R8*) ( )

Notice that the first grounded explanation for $\neg S6$ can be matched with the grounded explanation for F6 because S13 and F13 match. This result points to an inconsistency in the definitions of the *TeachingAssistant* concept in STUDENT and STAFF ontologies. Further evaluation of the associated explanations may suggest a preference to be given to one definition and thus the contradiction may be resolved at the application level with respect to a particular query. Although very interesting, this issue is beyond the scope of this paper but we plan to address it in our future work.

Grounded explanations for the statements in the second subset above are:

- For F8: (F2, R2*, R9*) ( ), (F3, R3*, R9*) ( ), (F14, R4*, R9*) ( ), (F13, R4*, R5*, R9*) ( )
- For $\neg S8$: (S3, R3, R10, R15) ( )

None of the explanations for F8 can be matched to the explanation for $\neg S8$, which is why this inconsistency can be dismissed.

To summarize the results of the proposed validation technique relative to our example: The billing application will be required to resolve an inconsistency in the definitions of the *TeachingAssistant* concept. With respect to the specific query, with the populated $Abox_1$ and $Abox_2$, instances of the *ParkingPayer* class will be correctly classified, while instances of the *TuitionPayer* class may not be correctly classified.

## VI. Conclusion

In this paper, we have presented a validation technique intended to test the consistency of independently created ontologies to address the needs of an application utilizing them. The validation process is performed on models of input ontologies represented as context-dependent rule sets. This provides a framework where both ontologies can be expressed in a common format, and allows for an independent inference procedure to not only detect inconsistencies based on structural mismatches of identical concepts or concept properties, but also to derive explanations to potentially resolve such inconsistencies at the application's level.

### References

[1] M. Gueffaz, P. Pittet, S. Rampacek, C. Cruz, and C. Nicolle, "Inconsistency Identification in Dynamic Ontologies Based on Model Checking" , in *Proc. 8th Int. Conf. on Web Information Systems and Technologies*, April 2012, Porto, Portugal, pp. 418–421.
[2] M. Horridge, B. Parsia, and U.Sattler, Explaining inconsistencies in OWL ontologies, in *Scalable Uncertainty Management*, Lecture Notes in Computer Science, vol. 5785, SpringerLink, 2009, pp. 124-137.
[3] P. Shvaiko and J. Euzanat, Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, 2013, 25 (1) 00. 158-176.
[4] B. Grau et al., "Results of the Ontology Alignment Evaluation Initiative 2013," http://oaei.ontologymatching.org
[5] T. Eiter, M. Fink, P. Schuller, and A. Weinzierl, "Finding explanations of inconsistency in multi-context systems", *Artificial Intelligence*, 216, Elsevier Publ., 2014 pp. 233-274.
[6] K. Kotis, G. Vouros, and K. Stergiou, "Towards automatic merging of domain ontologies: The HCONE-merge approach", *Journal of Web Semantics*, 4(1), 2006.
[7] N. Anjum, J. Harding, B. Young, K. Case, "Analysis of Ontology Mapping tools and Techniques". *Enterprise Interoperability IV*, Springer, 2010.
[8] D. Bell, G. Qi, W. Liu "Approaches to inconsistency handling in description-logic based ontologies", In *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems* - Volume Part II, 2007, Springer-Verlag, pp. 1303-1311.
[9] Y.Ma, P. Hitzler and Z.Lin, "Algorithms for Paraconsistent Reasoning with OWL," In *Proc. ESWC'2007*.
[10] T. Lukasiewicz, "Expressive probabilistic description logics," *Artificial Intelligence Journal*, vol. 172, no. 6-7, 2008, pp. 852-888.
[11] N. Zlatareva, "Context-dependent reasoning for the Semantic Web", *Journal of Systemics, Cybernetics and Informatics*, vol. 9, number 4, 2011, IIIS Press.
[12] N. Zlatareva and M. Misheva, "Alignment of Heterogeneous Ontologies: A Practical Approach to Testing for Similarities and Discrepancies. In *Proc. 21st International FLAIRS Conference*, AAAI Press, 2008.
[13] A. Ginsberg, "Theory reduction, theory revision, and retranslation", In *Proc. 8th National Conference on Artificial Intelligence*, vol. 2, 1990, MIT Press, pp. 777-782.
[14] A. Krisnadhi, F. Maier, and P. Hitzler, "OWL and Rules", In *Proc. 7th International Conference on Reasoning Web: Semantic Technologies for the web of data*, 2011, Springer-Verlag, pp. 382-415.
[15] B. Motic and R.Rosati, "Reconciling Description Logics and Rules", *Journal of the ACM*, vol.57, No.5, 2010.