Ontology Quality by Detection of Conflicts in Metadata

I. Budak Arpinar
LSDIS Lab, Computer Science Dept.
University of Georgia
Athens, GA, USA 30602-7404
1-706-583-0828
budak@cs.uga.edu

Karthikeyan Giriloganathan
LSDIS Lab, Computer Science Dept.
University of Georgia
Athens, GA, USA 30602-7404
1-706-542-4772
kart@cs.uga.edu

Boanerges Aleman-Meza
LSDIS Lab, Computer Science Dept.
University of Georgia
Athens, GA, USA 30602-7404
1-706-542-4772
boanerg@cs.uga.edu

ABSTRACT

Populated ontologies continue to be an important component in techniques and applications in semantic technologies. Thus, it is necessary to evaluate their quality. Our focus is the detection of conflicting information (within an ontology) as a criterion to improve the quality of an ontology. We describe different types of conflicts and propose a rule-based approach by which human experts can define conditions that signal a conflict in data. These rules (represented using RuleML) are used to automatically detect conflicts in populated ontologies. We describe a prototype application and evaluate the applicability of this approach.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous.

General Terms

Algorithms, Experimentation.

Keywords

Semantic Web, Conflict, Rule, RuleML, RDF, Ontology Quality, Ontology Evaluation.

1. INTRODUCTION

The focus of contemporary data and information retrieval systems has been to provide efficient support for querying and retrieval of data. Due to the increasing move from data to knowledge, and the growing popularity of the Semantic Web vision, ontologies play a significant role for representation of knowledge. The consumer of such knowledge (i.e., facts, semantic metadata), either a human or an application, can be negatively impacted by low quality ontologies. One aspect of quality information is avoidance of conflicting data. Hence, a user should be made aware when s/he is dealing with contradicting information. It is then important to (semi-)automatically identify conflicts within populated ontologies.

Semantic Web languages (like RDF/S [3, 20], and OWL [27]) share the basic concept of triples. That is, subject, predicate, and object, as in "*Anna motherOf John*". An ontology containing a large number of triples could contain sets of triples that are conflicting (or contradictory). For example, the before mentioned triple, together with the triple "*John marriedWith Anna*" is a conflict. We claim that detection of conflicts can help to improve the quality of an ontology. Conflicts can occur between RDF statements as in the previous example, but can also occur among sets of interconnected statements. For example, different sequences of statements might seem correct if analyzed independently of each other but they could be conflicting when considering them as a whole. As relationships continue to play a central role [31] in the Semantic Web, the identification of conflicts gains more relevance. Complex relationships among entities can be conflicting, based on a subjective interpretation of their meaning. For example, a leader of organization x cannot be supporter of an organization y that competes with x. Human experts in the domain of the ontology in question can identify types or sequences of statements that are to be considered in a conflict. However, we cannot expect humans to validate such conflicts in large populated ontologies, such as TAP [16], SWETO [1], and GLYCO [10]. Furthermore, *inconsistency checking* has been stated as an important part of the requirements for the OWL language. The OWL design document[1] justifies the requirement as follows: "The Web is decentralized, allowing anyone to say anything. As a result, different viewpoints may be contradictory, or even false information may be provided. In order to prevent agents from combining incompatible data or from taking consistent data and evolving it into an inconsistent state, it is important that inconsistencies can be detected automatically." Our work can be viewed as a consistency checking approach but not intended to detect logic inconsistencies. Instead, we focus on detecting conflicting relationships or sequences thereof.

The problem we address is improvement of quality of ontologies. There are many aspects of quality. Our method consists of detection of domain-specific, conflicting information. In our approach, we take into consideration a set of conflict-detection rules, which are defined by a domain expert. This set of rules is validated against a populated ontology, and the conflicting statements are shown to the user. Instead of focusing on the vocabulary definition of ontologies, we focus on the population or instance base of ontologies. We claim that by providing the user with the statements that are in conflict (and the cause of conflict) s/he can then take action towards improving the quality of the ontology. Our contributions on identifying conflicts as a criterion towards improving quality are as follows:

- A formalization and classification of conflicts in populated ontologies.
- A rule-based approach (using on RuleML [2]) to define and identify domain-specific conflicts automatically. A prototype implementation of this approach was developed, for which we describe preliminary results.

WWW 2006, May 22–26, 2006, Edinburgh, UK.

---

[1] http://www.w3.org/TR/webont-req/#goal-inconsistency

2. MOTIVATION

Data on the Web is typically perceived differently by humans. Hence, integrating data from different sources can have conflicting information. As the amount of available data continues to increase, it is becoming more important to identify conflicts that typically occur when data is integrated from multiple sources. Data on the Semantic Web also has the problem of conflicting information, such as "wrong" emails in FOAF data (www.foaf-project.org) that are detected when multiple FOAF documents are aggregated [8]. In addition, the identification of conflicts can potentially ensure that retrieved information is reliable or trusted.

Semantic metadata can be described as content enriched with semantic annotations using classes and relationships from an ontology [32]. Tools that generate semantic metadata from the Web are responsible for making sure that the ontology does not contain contradicting or conflicting information which will affect its applicability.

The type of conflicts discussed in this work is different from the term 'semantic conflict' in some literature (e.g., [23]). For instance, 'semantic conflict' has been used to refer to the usage of the same term with different meanings resulting in ambiguity in understanding the information. For example, one source may use the term 'rate' as charges after taxes and another source may use the term 'rate' as charges before taxes. In our discussion of conflicts we do not consider this kind of ambiguity conflicts.

3. CONFLICTS

3.1 Definition of Conflicts

Before presenting the conflict definitions, the terminology we used is as follows:

t	a single triple
T	a set of triples
S	A function denoting the process of *simplification*
s	The result of *simplification* (S(T)→s), could be a single triple or again a set of triples
U	Constraints expressed in an ontology, e.g., the property '*biologicalMother*' is unique
E	Constraints supplied by an expert, e.g., person(x) can never do action(y)

Definition 1: A *simplification* s is the result of any function S that reduces the complexity of a set of triples but still preserves some meaning (from human point of view). The goal is to take into consideration the granularity of information (a set of triples or a single triple) because some conflicts may occur at the level of triples (i.e., subject, predicate, and object). Other conflicts may be between complex relations instead of triples where a complex relation may span several triples. Thus, a *simplification* reduces the granularity level of information to be compared.

Definition 2: Two sets of triples T_1 and T_2 are said to be in conflict if their simplifications $S_x(T_1) \rightarrow s_1$ and $S_y(T_2) \rightarrow s_2$ are mutually non-agreeable.

Definition 3: Two simplifications s_1 and s_2 are mutually non-agreeable if when taken together they are in violation of constraints in either of the sets U or E.

The definitions of the simplification functions, the constraints U, and E enable flexibility in conflict definitions given that they are created by domain experts. A choice of constraints from U, or E is typically subjective and even arbitrary as it depends upon the domain of interest or purpose of the analysis. We use the term 'mutually agreeable' (instead of 'mutually exclusive') to allow for levels of disagreement. For example, triples about a person that state that he is a champion in both gymnastics and wrestling are mutually non-agreeable. It is intuitively non-agreeable for us that a person can excel at both sports. Therefore by signaling a certain level of disagreement it is possible that hidden inconsistencies in an ontology can be detected and then action can be taken to improve its quality.

In a simplification, a set of information (i.e., statements) is reduced to simpler knowledge units. For example, given a set of relationships involving a person in particular, we can arguably state his/her family relationships such as those involving cousins, uncles and siblings. We use the idea of simplification to identify conflicts among complex relations by reducing them to more manageable units. The goal of doing this is to facilitate a domain expert in the definition of conflicting information. In terms of RDF, we consider three types of simplifications:

First, an RDF triple is by definition a simplification (i.e., basic piece of knowledge).

Second, we might be able to compose relations [35] to a single relation between a subject and an object. Let F denote the set of entities and P denote the set of relations in a set of statements of an RDF graph where Q is the set of entities therein, $Q = \{q_1, q_2, ..., q_n\}$, and $P = \{p_1, p_2, ..., p_m\}$. Let P^+ be the power set of P. That is, $P^+ = \{(p_1), (p_2), ..., (p_m), (p_1, p_2), ..., (p_1, ..., p_m)\}$. Let C be a subset of P^+ consisting of only groups of relations that can be composed into a single relation, that is, $C = \{(p_1, p_k), ..., (p_a, p_b, p_c, ...)\}$. Let R be the set of relations obtained by substituting the composed relation for the composable relations, then $R = \{r_1, r_2, ..., r_n\}$, where $r_1, r_2, ..., r_n$ are results of the composition. The statement $<q_i \; r_k \; q_j>$ is a simplification if $r_k \in R$ and $q_i, q_j \in Q$. In the example shown in Figure 1 the statement "*ChrisRock supporterOf RepublicanParty*" is a simplification because the relation '*supporterOf*' is a result of composition of the relations '*votedFor*' and '*memberOf*.'

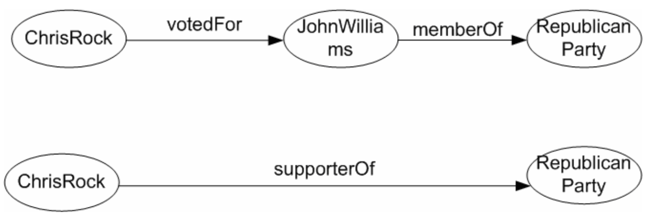


Figure 1. Example of a composition

Third, there could be background knowledge based simplifications of the form $statement_1 \wedge statement_2 \wedge \ldots statement_n \rightarrow statement_t$. In this case, $statement_t$ is a simplification. This type of simplification will depend on expert knowledge. A money laundering example is shown in Figure 2 to illustrate *simplification*. This set of triples tries to capture, from the knowledge base, instances of a person making multiple deposits in a financial organization and working for a business organization that is owned by somebody well known to the owner (who is an immigrant) of another business organization that has employees who are under investigation by a judicial organization. The dotted lines show some possible simplifications that can be done on this set of triples such as "*funded-by*". This simplification is possible only through an expert's knowledge involving these subjects. Note that this type of simplification is different than the relationship composition in the previous item where a series of nodes are assembled and the end points do not change in the composition. However, in this third case, the simplification can result in totally new statements with potentially new nodes (e.g., '*MoneyLaundering*' was not part of initial set of nodes). In terms of RDF, these new nodes are not expected to become part of the instance base; they only exist during the conflict-detection process.

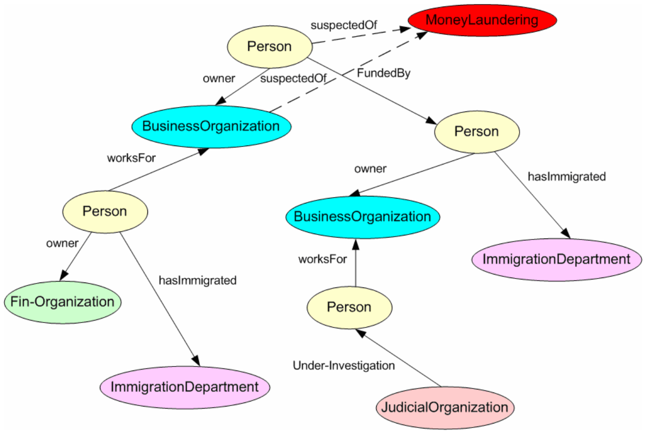


Figure 2. Simplification of a statement

Now consider a conflict in terms of simplification from two sets of statements. By composing the relations, '*marriedTo*' and '*motherOf*' into the relation '*fatherOf*' we get the simplification "*John fatherOf Bill*" (dotted line in Figure 3(top)). The resulting simplification "*John fatherOf Bill*" and an existing simplification "*John fatherInLawOf Bill*" are mutually non-agreeable (Figure 3(bottom)). Therefore, they are considered to be in conflict and their detection enables a refinement and validation of the metadata.

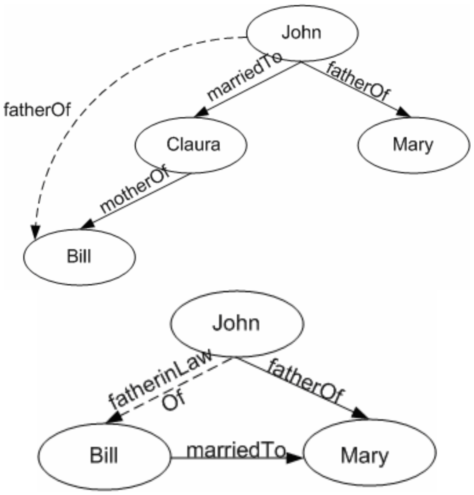


Figure 3. Example of simplification and conflict

3.2 Types of Conflicts

We classify conflicts based on the type of assertion that the simplifications violate. In the following subsections we use of prefixes rdf, rdfs, and owl to refer to their respective namespaces.

3.2.1 Property Assertion Conflicts

Property Assertion Conflicts occur when constraints placed on a property *p* are violated. For existing constrains of owl properties, warnings can be provided to the user based on the semantics of their intended usage/interpretation (i.e., owl:FunctionalProperty, owl:InverseFunctionalProperty, owl: SymmetricProperty, etc). However, our work does not intend to replace existing reasoners. We focus on domain-dependant conflicts, which in many cases cannot be stated (or their specification is not practical) using existing constraints of Semantic Web languages. For example, a constraint that our system supports is that of *asymmetric* relations. If *p* is said to be *asymmetric*, then $<e_1\ p\ e_2>$ and $<e_2\ p\ e_1>$ are in conflict. If a property has an *asymmetric* constraint, then it cannot connect a subject to an object and vice versa (i.e., in both the forward and the reverse directions). For example, assume '*situatedSouthOf*' is specified as an *asymmetric* property. Then, the triples "*Canada situatedSouthOf USA*" and "*USA situatedSouthOf Canada*" are in conflict. A knowledge expert is the person that defines which properties are *asymmetric*.

3.2.2 Class Assertion Conflicts

Class Assertion Conflicts occur when constraints placed on classes are violated. We consider here the type of assertions possible using OWL, in particular, *disjoint* classes. If class c_1 has a relationship '*owl:disjointWith*' to c_2, then "*x subclassOf* c_1" and "*x subclassOf* c_2" signal a conflict. Similarly, the relation *rdf:ype* used with the same entity over disjoint classes signals a conflict. For example, if class '*Citizen*' and '*Immigrant*' are disjoint then the statements "*Bill type Citizen*" and "*Bill type Immigrant*" are in conflict. We acknowledge that existing reasoners can take care of detecting this type of conflicts yet we considered this type of assertions for completeness.

3.2.3 Statement Assertion Conflicts

For the case of Statement Assertion Conflicts, an assertion indicates that under specific conditions the given statements are in conflict. These assertions are defined by a knowledge expert given that the conflicts to be detected are domain-specific. For example, assume that we want to say that a person cannot be both a supervisor and a friend of someone at the same time. Thus, the statements "*x superiorOf John*" and "*x friendOf John*" are in conflict. We use a '?' mark on 'x' to show that 'x' can be replaced by an instance from the knowledgebase. For expressing this kind of conflict, we define rules in RuleML.

Some of the conflict types can be detected automatically based on the ontology. Other conflict types require a human that provides the constraints. The basic idea for identifying conflicts is to convert assertions into rules and to signal a conflict if these rules are violated. We use RuleML [2] as an intermediate step in identifying conflicts, and translate assertions in RDF(S), or OWL to RuleML rules (e.g., in an intuitive If-Then format). We used reification to automatically assign an id to each statement in order to have rules about statements. Thus, when two statements are in conflict, we syntactically represent the conflict as a statement about the two statements.

The notion of simplification allows indicating that given certain condition(s), a new statement can be added. Simplifications are syntactically written as RuleML rules. Figure 3 provides an example of simplification whereby relations '*marriedTo*' and '*motherOf*' result in the addition of a new statement with the relation '*fatherOf*.'

4. PROTOTYPE APPLICATION

We implemented a prototype to evaluate our approach of using conflict detection towards improving ontology quality. We have taken a rule-based approach to identify conflicts. Rule languages such as RuleML [2] and SWRL (Semantic Web Rule Language) [17] have been developed to specify rules on semantic metadata. Figure 4 provides an overview of the system architecture of the prototype. Detection of conflicts involves the following steps:

- RDF documents are the input to the system.
- The simplification rules are based on expert knowledge and added either through a user interface or by means of RuleML files.
- Simplifications are enumerated and added as triples into the knowledgebase. This step is repeated until no more simplifications can be added.
- Assertions/constraints provided by user are translated into rules and placed in the rule-base.
- A rule engine identifies conflicting statements by querying the knowledgebase.

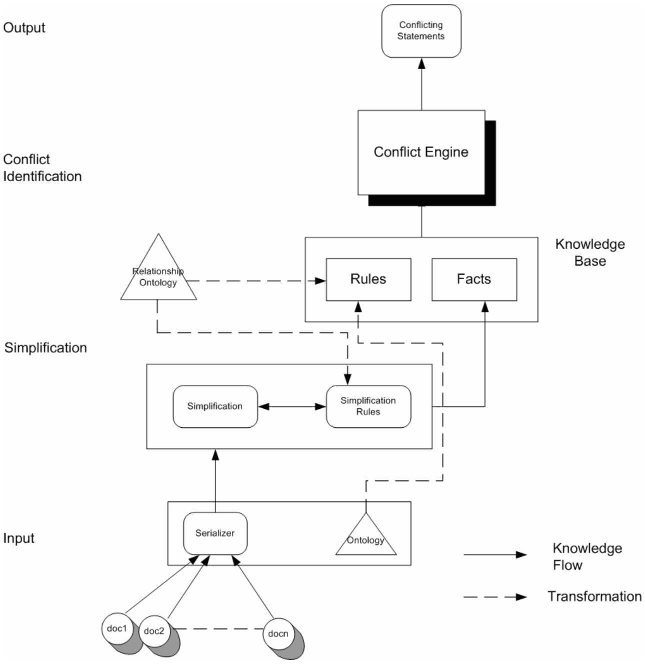


Figure 4. Overview of system architecture

We included predefined rules for detecting conflicts in OWL, in particular, *disjointWith*, *FunctionalProperty*, and *InverseFunctionalProperty*. This was our first step on evaluating the applicability of using a rule-based approach before moving to detection of domain-specific conflicts. A conflict engine uses rules and statements to identify the conflicts and generates a list of the conflicting pairs of statements as output. Figure 6 illustrates how the knowledgebase can be queried for conflicts. The queries are evaluated using a backward-reasoning algorithm implemented using the Mandarax API [5, 6]. Mandarax is an open source Java class library for deduction rules. It provides an infrastructure for defining, managing and querying rule bases. Mandarax contains a reference implementation of a very flexible inference engine (i.e., it provides unification algorithm, loop checking algorithm and selection policy). The result of executing a query is a set of ids of statement pairs that are in conflict. The GUI shows the components of the concerned statements as well as the particular conflict rule(s) that detected the statements to be in conflict. Additionally, a derivation tree provides an answer of "why" a pair of statements is in conflict.

The simplification process is executed as a query and the resulting statements are stored back in the knowledgebase but discarded after the conflict-detection process terminates. The derivation is stored along with the simplification. The derivation in this case would be the simplification rule used together with the statements that were brought together to generate a simplification. Thus, when there is a conflict involving a simplification, it is possible to provide detail of how that simplification was achieved using the associated derivation.

4.1 Experimental Evaluations

For the evaluation of our approach, we used a subset of SWETO ontology [1] containing over 6,000 entities and more than 11,000 explicit relations among them. This subset contains bibliography information such as journals, conferences, authors, and papers. Figure 6 illustrates the statements detected by a conflict-rule that verifies that no two papers are published in different journals. In this case, two statements (with ids 'ID_51' and 'ID_55') were in conflict because they have the same subject ('sweto:SWEET_1667893') connected to different objects ('sweto:SWEET_1666006' and 'sweto:SWEET_1666007') through the same property 'sweto:Published_In'. In this dataset, 'SWEET_1667893' was the resource Id of a publication. 'SWEET_1666006' and 'SWEET_1666007' were the resource ids of different journals.

4.2 Scalability Evaluation

With respect to scalability, Mandarax has a parameter 'maxsteps' that specifies the maximum number of derivation steps that it should perform before it gives up. The value of this parameter determines the depth of the tree that the rule engine uses. As the number of triples increases, the time taken to construct the evaluation tree increases. Figure 5 illustrates execution time (y-axis) versus number of triples (x-axis). In this figure, it can be seen that a threshold exists where the tree is saturated and so the time taken to detect the conflicts almost becomes a constant regardless of increasing the number of statements. When the number of triples is more than what the tree can handle, then the inference engine is not able to detect the conflicts.

For the case of multiple rules, each of them is evaluated individually and the results accumulated. With a large set of facts and a relatively limited number of rules this methodology will be efficient. However, when the number of rules increases there will be scalability issues because each rule has to be evaluated over the entire set of facts. We plan address scalability issues in future work. Additionally, when a triple is represented as a fact, we use four predicates (statement, subject, property, and object). This increases the amount of memory required to hold the knowledgebase in memory and may also limit the scalability of the approach. One design choice that could have helped is to represent the triple as a single predicate 'triple (id, subject, property, object)'. We made a choice to use binary predicates (predicates with two parameters), which resembled triples closely.

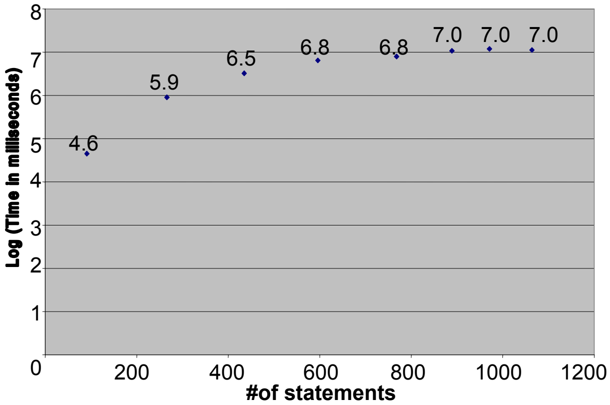


Figure 5. Performance with increase in number of triples

5. RELATED WORK

Evaluating the quality of ontologies has been addressed along many dimensions. Quality metrics have considered statistical aspects [38]; content, language and usage costs [22]; syntactic, semantic, pragmatic, and social aspects [4]; and improvements through transformations [24]. The characteristics of ontology vocabularies (i.e, ontology schema) have been reviewed by [26]. However, evaluation of the content of ontologies is different than evaluating tools for their development [34]. Detection of (logic) inconsistencies is also an important issue [18]. Maintaining quality while ontologies change has also been studied [36]. The value of ontologies benefits various fields where ontology quality has been noted, such as in biology [33]. Additionally, guidelines to create semantically sound ontologies have been proposed [13].

Work in the field of electronic commerce has lead to several important ideas about conflicts. They deal with the priority of one business rule over other in the case that both are applicable. They are more concerned with resolving the conflicts than identifying it. An example would be the prioritized conflict handling from IBM [12]. They introduce a term called overrides to indicate which rule has priority over the other.

Trust management has been discussed comprehensively in the context of semantic Web [9, 11, 14, 25, 30]. Our work can be used to first detect conflicting information, and then existing trust techniques can be applied to take appropriate action. For example, selection of a recommendation for a product could utilize a trust network whenever there are conflicting recommendations from various users.

Recent efforts (i.e., Semantic Web Rule Language [17]) have tried to realize the logic layer of the Semantic Web by combining RuleML and OWL where rules of RuleML are written using vocabulary from OWL. Our work does not try to bridge the gap between rules and Semantic Web Languages. Rather it is an effort in utilizing some of the OWL constraints as rules that can be evaluated towards detection of conflicts. We believe that improvements on evaluation of ontologies will lead to more efficient search and ranking of ontologies (e.g., [7, 28, 37]). The types of conflicts that we believe are more important to be detected are those involving sequences or groups of relationships. Such constraints are not easily expressed using OWL constraints not easily identified during the creation of the ontology. Thus, we believe our approach is complementary to existing constrains of semantic web languages as it provides capabilities to specify and detect domain-specific conflicts leading to improvements on the quality of an ontology.

The issue of conflict in data appears in problems other than ontology quality. For example, when aggregating data from different sources, a human-defined context can be used to reconcile conflicting information [15]. The main difference with our approach is that our technique is performed once the data is in a populated ontology. This has the advantage on flexibility because rules can be defined to detect the existence of conflicts that might have not being thought of while harvesting or integrating the data. It is also worth mentioning that even when the same vocabulary (i.e., FOAF) is utilized by different people, conflicting information still appears when integrating data from multiple sources [8].

Detection of conflicts within data encoded utilizing Semantic Web languages has been addressed for data interoperability and integration [21]. They consider seven cases of conflicts yet these are different to the ones we address. Their work is along the lines of schematic and structural conflicts that have been studied in database interoperability [19, 25, 32], including recent techniques using ontologies [29]. An important difference with our work is that our approach is able to detect conflicts occurring on sequences (or groups) of relationships.

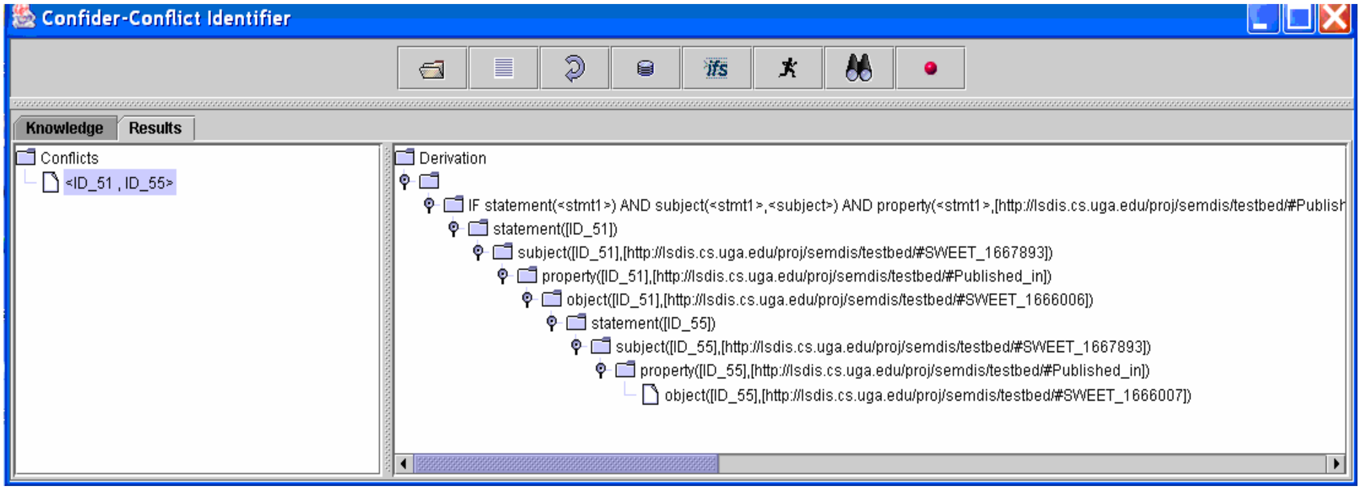
6. CONCLUSION AND FUTURE WORK

In this work, we have defined different types of conflicts that can appear in populated ontologies. We claim that detection of domain-specific conflicts within data of an ontology is an aspect for evaluating its quality. That is, our work can help in maintaining/improving ontology quality by identifying conflicts. The approach presented utilizes rules created by a human domain expert that are used to identify conflicts. A prototype implementation was developed to detect conflicts within populated ontologies (in RDF or OWL).

The evaluation of this approach seems promising yet scalability issues remain to be addressed. Thus, future work directions include development of more scalable conflict identification techniques for large amounts of semantic metadata and conflict rules. Figure 4 includes a component named 'relationship ontology' which is work in progress towards a repository of common constraints for conflict detection. We believe that detection of domain-specific conflicts will gain importance in areas identified as likely early adopters of Semantic Web technologies such as life-sciences, health-care and e-government (panel at the 2006 Semantic Web and Databases Workshop). Lastly, a ranking method for different types of conflicts can be developed to provide ranked results to user.

7. ACKNOWLEDGMENTS

This work is funded in part by NSF-ITR-IDM Award#0325464 titled 'SemDIS: Discovering Complex Relationships in the Semantic Web' and NSF-ITR-IDM Award#0219649 titled 'Semantic Association Identification Knowledge Discovery for National Security Applications.'



```
Conflicts
  <ID_51 , ID_55>
```

```
Derivation
  
    IF statement(<stmt1>) AND subject(<stmt1>,<subject>) AND property(<stmt1>,[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Publish
      statement([ID_51])
        subject([ID_51],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893])
          property([ID_51],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Published_in])
            object([ID_51],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1666006])
              statement([ID_55])
                subject([ID_55],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893])
                  property([ID_55],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Published_in])
                    object([ID_55],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1666007])
```

Figure 6. Results of conflict detection

8. REFERENCES

[1] Aleman-Meza, B., Halaschek, C., Sheth, A., Arpinar, I.B. and Sannapareddy, G., SWETO: Large-Scale Semantic Web Test-bed. In *16th International Conference on Software Engineering and Knowledge Engineering (SEKE2004): Workshop on Ontology in Action*, (Banff, Canada, 2004), 490-493.

[2] Boley, H., Tabet, S. and Wagner, G., Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In *The first Semantic Web Working Symposium*, (Stanford University, California, USA, 2001).

[3] Brickley, D. and Guha, R.V. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 2000.

[4] Burton-Jones, A., Storey, V.C., Sugumaran, V. and Ahluwalia, P. A Semiotic Metrics Suite for Assessing the Quality of Ontologies. *Data & Knowledge Engineering*, *55* (1). 84-102.

[5] Dietrich, J. The Mandarax Manual, http://mandarax.sourceforge.net/docs/mandarax.pdf, 2003.

[6] Dietrich, J., Kozlenkov, A. and Schroeder, M. Rule-based Agents for the Semantic Web. *Journal on Electronic Commerce Research Applications*, *2* (4). 323-338.

[7] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V. and Sachs, J., Swoogle: A Search and Metadata Engine for the Semantic Web. In *International Conference on Information and Knowledge Management*, (Washington, DC, USA, 2004).

[8] Ding, L., Zhou, L., Finin, T. and Joshi, A., How the Semantic Web is Being Used: An Analysis of FOAF Documents. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, (Big Island, HI, USA, 2005).

[9] Finin, T. and Joshi, A. Agents, Trust, and Information Access on the Semantic Web. *Sigmod Record*, *31* (4). 30-35.

[10] Glyco-Ontology. http://lsdis.cs.uga.edu/Projects/Glycomics.

[11] Golbeck, J., Parsia, B. and Hendler, J. Trust networks on the semantic web. *Cooperative Information Agents VII, Proceedings*, *2782*. 238-249.

[12] Grosof, B.N. Courteous Logic Programs: Prioritized Conflict Handling for Rules *IBM Research Report RC 20836*, 1997.

[13] Guarino, N. and Welty, C. Evaluating Ontological Decisions with OntoClean. *Communications of the ACM*, *45* (2). 61-65.

[14] Guha, R., Kumar, R., Raghavan, P. and Tomkins, A. Propagation of Trust and Distrust *Proceedings of the 13th International Conference on World Wide Web*, ACM Press, New York, NY, USA, 2004.

[15] Guha, R., McCool, R. and Fikes, R., Contexts for the Semantic Web. In *International Semantic Web Conference*, (Hiroshima, Japan, 2004), 32-46.

[16] Guha, R.V. and McCool, R. TAP: A Semantic Web Test-bed. *Journal of Web Semantics*, *1* (1). 81-87.

[17] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B. and Dean, M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2003.

[18] Kalyanpur, A., Parsia, B., Sirin, E. and Hendler, J. Debugging Unsatisfiable Classes in OWL Ontologies. *Journal of Web Semantics*, *3* (4). 268-293.

[19] Kim, W. and Seo, J.Y. Classifying Schematic and Data Heterogeneity in Multidatabase Systems. *IEEE Computer*, *24* (12). 12-18.

[20] Lassila, O. and Swick, R.R. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, W3C, 1999.

[21] Li, C. and Ling, T.W. *OWL-Based Semantic Conflicts Detection and Resolution for Data Interoperability*, 2004.

[22] Lozano-Tello, A. and Gomez-Perez, A. ONTOMETRIC: A Method to Choose the Appropriate Ontology. *Journal of Database Management*, *15* (2). 1-18.

[23] Lu, H., Fan, W. and Goh, C.H., Discovering and Reconciling Semantic Conflicts: A Data Mining Perspective. In *Proceedings of the 7th IFIP 2.6 Working Conference on Data Semantics (DS-7)*, (Leysin, Switzerland, 1997).

[24] Mostowfi, F., Fotouhi, F., Improving Quality of Ontology: An Ontology Transformation Approach. In *Third Semantic Web and Databases Workshop*, (Atlanta, Georgia, USA, 2006)

[25] Naiman, C.F. and Ouksel, A.M., A Classification of Semantic Conflicts in Heterogeneous Database Systems. *Journal of Organizational Computing*, *5* (2). 167-193.

[26] Noy, N.F. and Hafner, C.D. The State of the Art in Ontology Design - A Survey and Comparative Review. *AI Magazine*, *18* (3). 53-74.

[27] Patel-Schneider, P.F., Hayes, P. and Horrocks, I. OWL Web Ontology Language Semantics and Abstract Syntax. W3C Technical Recommendation.

[28] Patel, C., Supekar, K., Lee, Y. and Park, E.K. OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification *Proceedings of the 5th ACM International Workshop on Web Information and Data Management*, ACM Press, New Orleans, Louisiana, USA, 2003.

[29] Ram, S. and Park, J. Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflicts. *IEEE Transactions on Knowledge and Data Engineering*, *16* (2). 189-202.

[30] Richardson, M., Agrawal, R. and Domingos, P. Trust Management for the Semantic Web. *Semantic Web - ISWC 2003*, *2870*. 351-368.

[31] Sheth, A.P., Arpinar, I.B. and Kashyap, V. Relationships at the Heart of Semantic Web: Modeling, Discovering and Exploiting Complex Semantic Relationships. In Nikravesh, M., Azvin, B., Yager, R. and Zadeh, L.A. eds. *Enhancing the Power of the Internet Studies in Fuzziness and Soft Computing*, Springer-Verlag, 2003.

[32] Sheth, A.P. and Kashyap, V. So Far (Schematically) yet So Near (Semantically). *IFIP Transactions a-Computer Science and Technology*, *25*. 283-312.

[33] Soldatova, L.N. and King, R.D. Are the current ontologies in biology good ontologies? *Nat Biotech*, *23* (9). 1095-1098.

[34] Staab, S. Why Evaluate Ontology Technologies? - Because it Works! *IEEE Intelligent Systems*, *19* (4). 74-81.

[35] Staab, S. and Maedche, A., Axioms are Objects too - Ontology Engineering beyond the Modeling of Concepts and Relations. In *Workshop on Ontologies and Problem-Solving Methods*, (Berlin, Germany, 2000).

[36] Sugiura, N., Shigeta, Y., Fukuta, N., Izumi, N. and Yamaguchi, T. Towards On-the-fly Ontology Construction Focusing on Ontology Quality Improvement. *Semantic Web: Research and Applications*, *3053*. 1-15.

[37] Supekar, K., Patel, C. and Lee, Y. Characterizing Quality of Knowledge on Semantic Web *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*, AAAI Press, Miami Beach, Florida, USA, 2004.

[38] Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P. and Aleman-Meza, B. OntoQA: Metric-Based Ontology Quality Analysis *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, Houston, TX, USA, 2005.