20th ICCRTS
“C2, Cyber, and Trust”

Ontology Matching Using Structure and Annotations in XML Schema

Topics
C2-simulation Interoperability
Modeling and Simulation
Data, Information and Knowledge

Name of Author
Samuel Suhas Singapogu
George Mason University C4I Center

Point of Contact
Samuel Suhas Singapogu
George Mason University
4400 University Drive, Fairfax, VA 22030
703-993-3082
ssingapo@c4i.gmu.edu
Abstract

The next generation of command and control to simulation interoperability (C2SIM) technology will use ontologies to capture semantic data in the form of concepts, relationships and axioms. A well-defined and well-populated ontology captures semantic information to model existing knowledge and infer new knowledge effectively. Interoperable systems in the future will be expected to have their own input specification and knowledge representation (ontology). In order to provide interoperability on the semantic level, there is a need to identify how ontologies of different systems compare to each other. Ontology matching is the process of finding concepts that have similar, dissimilar and inheritance relationships between concepts of two ontologies. The output of the matching process is an ontological alignment that can serve as a mapping between the two ontologies/systems.

Existing ontology matching methods largely depend on the edit-distance method of comparing string similarity between concept names in order to compute semantic similarity. XML schemas, often used to model syntax of a system, contain data useful to semantics. The structure and annotations within XML schemas can be used to compute semantic similarity. This paper extends ontology matching using edit-distance similarity measure by identifying and incorporating measures drawn from XML schema structure and annotations. It is shown that this method can be used to compare any two ontologies that have accompanying annotated XML schemas. The newly developed semantic similarity measure is applied to comparing a C2SIM ontology to a similar sample ontology.
1. Introduction

Command and Control to Simulation Interoperability (C2SIM) is a standard under development by the Simulation Interoperability Standards Organization (SISO) to facilitate interoperability between command and control (C2) systems and simulation systems [18]. The current phase of standardization effort is focused on developing a formal semantic model, including an ontology, in order to provide interoperability on the semantic level [1]. An ontology captures semantic definitions of concepts, relationships and axioms. Concepts are the abstraction of related attributes that provide the defining semantic building blocks of a domain. Relationships are defined between two concepts or a concept and a data type. Axioms are assertions expressed in a form of a ‘subject predicate object’ triple that defines the existing knowledge of a domain. Existing software reasoners based on predicate logic can be used to infer new knowledge in the domain using axioms and relationships. An ontology can be used as a formal specification of semantics based on existing W3C standards.

Ontology matching is the process of comparing two ontologies using syntactic and semantic information. The goal of solving the ontology-matching problem is to find correspondences between concepts in the ontologies being compared. A correspondence can have either equality, inequality or subsumption relationship. Two concepts are equal if they have identical semantic properties. Two concepts are unequal if objects belonging to the two concepts are disjoint. Two concepts have the subsumption relationship if one concept semantically inherits from the other concept. Even though a number of ontology matching techniques have been proposed and used, the problem of ontology matching is still considered an “unsolved problem” [3][4]. This is because there does not exist a single technique that fits all problems [3].

Studies have shown that most techniques rely on the Levenshtein method to compare string similarity between concept names. The Levenshtein method calculates the number of operations (add, delete, replace) required to convert one string to another. Semantic similarity is inversely proportional to the conversion cost. While the Levenshtein (edit-distance) method provides insights into semantic similarity, often it is insufficient to fully compute semantic similarity. This is because transformation distance between words does not always capture semantic similarity and the edit-distance method ignores grammar structure, syntactical context and global context of the words [15]. Furthermore, semantic similarity measures are present more in the structure of the domain [5] and its context than in the similarity of concept names. In domains where structure is not captured in a data model, the structure of the ontology has been used to measure semantic similarity [14]. In XML based systems where an XML schema has been developed, the schema provides a valuable base of structural information that can be useful in calculating semantic similarity.

While it has been noted that XML schemas contain semantic data [6], there have not been any methods put forward so far that compute semantic similarity using defined measures found in XML schemas. In this paper we propose a method to identify semantically relevant data in XML schemas. We then define a new semantic measure based on
parameters in XML schemas. We apply this method to comparing a C2SIM ontology to another similar sample ontology. We demonstrate how this method can be used to derive a mapping between two ontologies.

2. Previous Work

Existing research in ontology matching uses “word-similarity” analysis on concept names to compute semantic similarity between concepts in two ontologies. In order to calculate semantic similarity, most methods use Levenshtein method (also known as edit-distance method) to compare concept names [5]. Some methods use syntactic measures [7]; others use structural similarity measures of concepts and concept hierarchy [8].

The three relationships that are evaluated in ontology matching techniques are:

a) **Equality**: This relationship exists between two concepts that have identical semantic properties and can be used interchangeably in the two considered systems.
b) **Subsumption**: This relationship exists between two concepts that have a subclass-superclass relationship. In this relationship, the superclass can be replaced by the subclass since the subclass is more expressive and has all the elements that the superclass needs.
c) **Disjoint**: This relationship exists between two concepts c1 and c2 when it can be asserted that an object in c1 will not exist as an object in c2.

In this paper, we present a new method that uses semantic data in XML schemas to perform ontology matching.

3. Ontology matching using XML schemas

The proposed method uses two types of parameters to determine semantic similarity:
a) Structural information in XML schemas:
   (i) Parent complex type of concepts: Complex types that are parents to the concepts in the XML schema tree are compared for semantic similarity.
   (ii) Children complex type of concepts: Complex types that are children to the concepts in the XML schema tree are compared for semantic similarity using the edit-distance method.

Structural data in XML schemas contain class hierarchy than can be mapped to conceptual hierarchy in an ontology. In the proposed method, the concepts being compared are mapped to element definitions in the corresponding schemas. Not only are the two concept names compared, but their parent types and child types in the schema are also compared. The rationale being that the context of concepts (parents and children) have a bearing on the semantic similarity between the concepts. As an example consider the comparison between ‘RouteWhere’ and ‘RouteLightType’ in the C2SIM domain. The accompanying schema being used is the C-BML Phase 1 schema that has been balloted through a SISO process [10]. In the absence of the context of “RouteWhere” and “RouteLightType”, the semantic similarity between the two concepts is ambiguous. But when the corresponding parent elements (“OrderTask” and “TaskLightType”) and the child
elements are compared, their semantic similarity is further clarified. Figure 1 illustrates semantic comparison using structure in XML schemas and a detailed description of this process is described below.

Figure 1. Illustrating semantic similarity of concepts using structure in XML schema

b) Levenshtein comparison of:
   (i) Concept name: The two concepts c1 and c2 are compared using the Levenshtein method.
   (ii) XSD Annotations: The annotations in the XML schema corresponding to the two concepts c1 and c2 are compared using the Levenshtein method. Figure 2 shows a snapshot of the XML schema annotations for “RouteWhere” and “RouteLightType”. In the proposed method, these two annotations are compared as a weighted parameter in computing semantic similarity.
The goal is to use these parameters to create an alignment between the two ontologies. An alignment ‘A’ is a set of correspondences where each correspondence is a 4-uple represented as: \{id, c1, c2, r\}

\[
\begin{align*}
Where \\
id &= an \ identifier \ for \ the \ correspondence, \\
c1 &= concept \ from \ ontology \ 1, \\
c2 &= concept \ from \ ontology \ 2, \\
r &= the \ relationship \ between \ c1 \ and \ c2.
\end{align*}
\]

Figure 3 below illustrates the comparison method.
As illustrated in Figure 3, the proposed method uses elements from both the ontology itself and the associated XML schema to evaluate semantic similarity. In order to calculate the semantic similarity between concepts c1 and c2, the steps followed are described below:

1. **Step 1**: Compare the concept names c1 and c2 using the edit-distance method. The formula [9] to calculate the edit-distance value is

   \[
   \text{Lev}_{c1c2}(i,j) = \begin{cases} 
   \max(i, j) & \text{if } \min(i, j) = 0 \\
   \min \begin{cases} 
   \text{Lev}_{c1c2}(i-1,j) + 1 \\
   \text{Lev}_{c1c2}(i,j+1) + 1 \\
   \text{Lev}_{c1c2}(i-1,j-1) + 1 
   \end{cases} & \text{otherwise (if } \min(i,j) \neq 0) 
   \end{cases}
   \]

   If \(\text{Lev}(c1,c2) = 0\) the two concepts have highest semantic similarity. Based on a predefined weight \(\text{CONCEPT_WEIGHT}\) for concept similarity the value

   \[
   \text{BASIC_CONCEPT_SIMILARITY} = \text{CONCEPT_WEIGHT} \times \text{Lev}(c1,c2)
   \]

2. **Step 2**: Using the associated XML schemas, identify complex types that are parents to concepts c1 and c2. The number of levels in traversing the XML schema tree is regulated using the predefined variable \(\text{PARENT_LEVEL_HEIGHT}\). If the value of \(\text{PARENT_LEVEL_HEIGHT}\) is ‘2’, the similarity of the parent and the grandparent of c1 are taken into consideration. Iteratively traversing through the XML schema tree, the measure is calculated as:

   \[
   \text{PARENT_SUBTREE_SIMILARITY} = \sum_{i=1}^{PARENT_LEVEL_HEIGHT} (\text{PARENT_LEVEL}_i\_\text{WEIGHT} \times \text{Lev}({\text{parent(i,c1)}}, {\text{parent(i,c2)}}))
   \]
Where parent(i, C) is the $i^{th}$ level parent of the concept ‘C’ e.g. parent(2, C) is the
grandparent of C.

Step3: Similar to the process outlined in step2, the XML schema is traversed to calculate
the similarity of children complex types as:

$$\text{CHILD\_SUBTREE\_SIMILARITY} = \sum_{i=1}^{\text{CHILD\_LEVEL\_HEIGHT}} (\text{CHILD\_LEVEL\_i\_WEIGHT} \ast \text{Lev}(\text{child}(i, C1), \text{child}(i, C2)))$$

Where child(i, C) is the $i^{th}$ level child of the concept ‘C’ e.g. child(2, C) is the grandchild
of C.

Step4: The annotations associated with concepts c1 and c2 are compared and the
similarity measure is calculated as:

$$\text{ANNOTATIONS\_SIMILARITY} = \text{ANNOTATIONS\_WEIGHT} \ast \text{Lev}(\text{annon1}, \text{annon2})$$

Where ‘annon1’ is the XML schema annotation associated with c1 and ‘annon2’ is
XML schema annotation associated with c2.

The total weighted similarity measure is calculated as:

$$\text{WEIGHTED\_SIMILARITY\_MEASURE} = \text{BASIC\_CONCEPT\_SIMILARITY} + \text{PARENT\_SUBTREE\_SIMILARITY} +$$
$$\text{CHILD\_SUBTREE\_SIMILARITY} + \text{ANNOTATIONS\_SIMILARITY}$$

Depending on the design of the ontology and the XML schema, the user can chose a
threshold value for ‘WEIGHTED\_SIMILARITY\_MEASURE’ under which the concepts
are considered semantically similar.

4. Comparing a C2SIM ontology to a similar sample ontology

The proposed method is used to compare a C2SIM ontology with a similar sample
ontology. The C2SIM ontology has been derived from the balloted C-BML Phase 1 XML
schema [10]. The technique used creates concepts corresponding to <xs:element> definitions in the schema. Inheritance relationships are based on structural information in the schema and domain relationships are based on text mining schema annotations associated with <xs:element> definitions. Figure 4 shows the mapping from XML schema elements to ontology elements.
Concepts randomly selected from a C2SIM ontology and new sample concepts unrelated to C2SIM are used to create a new sample ontology. Table 1 below captures statistics regarding the two ontologies and Table 2 captures statistics regarding the XML schemas associated with the two ontologies:

Table 1. Statistics regarding C2SIM ontology and sample ontology

<table>
<thead>
<tr>
<th>Metric</th>
<th>Subset of C2SIM ontology</th>
<th>Created Sample Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of concept definitions</td>
<td>175</td>
<td>160</td>
</tr>
<tr>
<td>Number of inheritance relationship definitions</td>
<td>97</td>
<td>76</td>
</tr>
<tr>
<td>Number of domain relationship definitions</td>
<td>27</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 2. Statistics regarding the XML schemas associated with C2SIM ontology and sample ontology

<table>
<thead>
<tr>
<th>Metric</th>
<th>XML schema associated with C2SIM ontology</th>
<th>XML schema associated with sample ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of complex type definitions</td>
<td>531</td>
<td>489</td>
</tr>
<tr>
<td>Number of element definitions</td>
<td>1115</td>
<td>1079</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>Number of annotations</td>
<td>1604</td>
<td>967</td>
</tr>
<tr>
<td>Number of unbounded elements (Number of elements that have &quot;maxoccurs=unbounded&quot;)</td>
<td>40</td>
<td>37</td>
</tr>
<tr>
<td>Fanning Index[11] (Number of relationships/number of elements)</td>
<td>9.67</td>
<td>8.64</td>
</tr>
<tr>
<td>Complexity Measure (based on the formula in [11]):</td>
<td>738</td>
<td>598</td>
</tr>
</tbody>
</table>

A software prototype was written to perform the ontology matching using weighted similarity measure described in section 3. An ontology alignment API [11] based on [12] was used to perform the ontology matching and alignment between a C2SIM ontology and sample ontology. An alignment between the two ontologies is a set of correspondences with each correspondence of the form:

```
<map>
  <Cell>
    <entity1 rdf:resource="">Concept1</entity>
    <entity2 rdf:resource="">Concept2</entity>
    <measure rdf:datatype="">1.0</measure>
    <relation>=</relation>
  </Cell>
</map>
```

Figure 5. Example correspondence between two concepts

Each cell contains the two compared entities, a graded measure of the confidence of the mapping and the relationship between the two entities. Using the API, the “ObjectAlignment” class was extended to implement the weighted similarity measure described in section 3. The parameter values used in this experiment are:

Table 3. Parameter values for ontology matching between a C2SIM ontology and sample ontology

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONCEPT_WEIGHT</td>
<td>0.75</td>
</tr>
<tr>
<td>PARENT_LEVEL_HEIGHT</td>
<td>2</td>
</tr>
<tr>
<td>PARENT_LEVEL_1_WEIGHT</td>
<td>0.45</td>
</tr>
<tr>
<td>PARENT_LEVEL_2_WEIGHT</td>
<td>0.30</td>
</tr>
<tr>
<td>CHILD_LEVEL_HEIGHT</td>
<td>2</td>
</tr>
<tr>
<td>CHILD_LEVEL_1_WEIGHT</td>
<td>0.30</td>
</tr>
<tr>
<td>CHILD_LEVEL_2_WEIGHT</td>
<td>0.25</td>
</tr>
<tr>
<td>ANNOTATIONS_WEIGHT</td>
<td>0.55</td>
</tr>
<tr>
<td>THRESHOLD_LIMIT</td>
<td>25.75</td>
</tr>
</tbody>
</table>
Based on the above values, if the weighted similarity measure has a value less than 25.75, the two concepts are concluded to be similar. The results of the comparison process are as follows:

Table 4. Experimental test results

<table>
<thead>
<tr>
<th>Metric</th>
<th>Simple edit-distance matching</th>
<th>Weighted XML schema based matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of equality relationships</td>
<td>78</td>
<td>334</td>
</tr>
</tbody>
</table>

As captured in the results, the weighted similarity measure based on semantic parameters incorporated from XML schemas have out performed basic edit-distance comparison of concepts. This supports the fact that XML schemas contain semantically relevant elements and can be used in ontology matching and alignment.

The alignment output, which is a list of correspondences, can be used as a semantic mapping between the two systems/ontologies. The mapping identifies classes that are equivalent, classes that can be interchangeable (subclass for a superclass) and classes that are dissimilar. This mapping can now be used to translate input documents of one system to another. For example, using this method, a mapping can be obtained between C2SIM and C2core (which is another framework in the C2 domain). That mapping can be used to translate between C2SIM and C2core. Since the mapping is in the form of an XML document, the translation can be automated. This process can be a useful tool in semantic mapping of XML based systems, which is crucial to semantic interoperability. In addition, the alignment output can also be used in analysis and design to determine the extent of semantic alignment between two systems. There have been coalition experiments where interoperation between different command and control and simulation systems have been tested [16,17]. While making design decisions in such multi-nation, multi-system interoperating environments, this analysis can be a useful tool to determine the extent of semantic alignment between systems.

5. Conclusions and Future work

In this paper, we propose a novel method to perform ontology matching and alignment that leverages semantic data in XML schemas. Although it has already been noted that XML schemas contain semantic data [12][13], a technique to incorporate semantics from XML schemas into the problem of ontology matching and alignment has not yet been developed. The proposed method provides a viable basis to create a semantic mapping between two systems based on ontologies and XML schemas. When applied to comparing a C2SIM ontology to a sample similar ontology, the results demonstrate the value of semantic data in XML schemas. Mapping between C2 systems has been found to be a complex problem [14] and this method can be a useful tool to bridge the gap between XML based C2 and simulation system interoperability. Future work can focus on upgrading the accuracy of the similarity measure by using a synonym table in the C2
domain as a parameter to this method. Additionally, it would also be beneficial to create a Graphical User Interface (GUI) to capture the ontology alignment as an interactive mapping diagram. This may then be used to create interfaces between interoperating C2 and simulation systems for interoperability on the semantic level.

6. **References**


