Automated Ontology Creation using XML Schema Elements

Samuel Suhas Singapogu, Paulo C. G. Costa, J. Mark Pullen
C4I center, George Mason University
Fairfax, VA, USA
[ssingapo,pcosta,jpullen]@c4i.gmu.edu

***Abstract*—Ontologies are commonly used to represent formal semantics in a computer system, usually capturing them in the form of concepts, relationships and axioms. Axioms convey asserted knowledge and support inferring new knowledge through logical reasoning. For complex systems, the process of creating ontologies manually can be tedious and error-prone. Many automated methods of knowledge discovery are based on mining domain text corpus, but current state-of-the-art methods using this approach fail to consider properly semantic data embedded in XML schemata in complex systems. This paper proposes a mapping method for identifying relevant semantic data in XML schemata, automatically structuring and representing it in the form of a draft ontology. Concepts, concept hierarchy and domain relationships from XML schema are mapped to relevant parts of an OWL ontology. A part-of-speech tagging method extracts domain relationships from schema annotations. This mapping method can be applied to any system that has a well-annotated XML schema. We illustrate our process with the preliminary results obtained when creating a command and control to simulation (C2SIM) draft ontology from an XML schema.**

Keywords—OWL, XML Schema, Part of Speech tagging, Command and Control, Interoperability

I. INTRODUCTION

Knowledge discovery is essentially the process of extracting semantic concepts and relationships from domain resources within a particular domain. Ontologies are the *de facto* standard for representing knowledge of a system [1]. The framework and components of ontologies are based on established, yet evolving W3C standards. Ontologies usually are comprised of concepts, relationships and axioms. Concepts are abstractions of related attributes that form the basic building blocks of a semantic model. Relationships can be between two concepts or between a concept and a data-type. Axioms consist of asserted knowledge that can be represented as a `<subject, predicate, object>` triple. Logic-based reasoners can be used to infer new knowledge in an ontology.

Although ontologies are valuable assets in system modeling, testing and analysis, the process of manually creating an ontology for a complex system is inherently tedious and error prone. Existing methods of knowledge discovery and ontology creation usually are based on text mining of a data corpus for that domain. XML-based systems capture the structure and syntax of all necessary and meaningful elements in a XML schema, which therefore is a useful starting point for semantic analysis. In such systems, XML schemata have been shown to be a valuable resource of semantic data [2]. Command and Control systems in the military context support various functions, including commanding of forces and also receiving and interpretation of situational awareness reports. To perform these functions, most C2 systems are modeled using XML schemata e.g. Coalition Battle Management System (C-BML) [3], Military Scenario Definition Language (MSDL) [4], and National Information Exchange Model (NIEM) [5], which represent the systems' structural and syntactic framework. These systems use and exchange XML documents that are based on XML schemata.

XML often is used as the exchange mechanism in the command and control domain [6]. Given the increasing number of XML-based systems in the C2 domain, an automated framework that leverages semantic information in the XML schema and creates a draft ontology would be a useful tool. Domain experts can refine and populate the draft ontology using concept hierarchy and basic domain relationships. For large XML based systems, this process of creating a draft ontology saves valuable time and avoids errors common to the alternative tedious, manual process. In contrast to existing techniques used to map a XML schema to an ontology, the mapping proposed in this paper highlights the need to map schema element (`xs:element` from here forward) to an ontological concept. This avoids the usual approach of mapping XML complex types to concepts that could lead to unnecessary ontological complexity. In addition, this paper proposes a novel Part of Speech tagging

method to extract domain relationships from well-annotated XML schema.

II. Related Work

Most existing work on mapping a XML Schema to an ontology (e.g. [7], [8]) is based on mapping XML schema complexType (henceforth referred to as `xs:complexType`) to an `owl:class`. This mapping can lead to problems for the following reasons:

1. A "simpleType" definition is sufficient to define a semantic concept. In the command and control domain, for example, it is possible to define an element called "UnitName," which is of a "simpleType" string (with string restrictions). There is sufficient semantic information in this definition to create a distinct ontological concept. When this level of abstraction of concepts is ignored and only complex type definitions are considered as semantic concepts, the resulting ontologies will contain significant modeling gaps for any useful analysis.
2. By design, XML parsing only allows elements associated to a complex type to appear in valid XML files. XML schemata can contain complex type definitions that are never associated to an element definition. When concepts are mapped to complex types, the resulting ontology will likely include concepts that will never appear in the XML document. This unnecessary complexity is counterproductive to efficient semantic modeling in design and analysis.

Bohring et al. [9] recognize the value of semantic concepts being mapped to `<xs:element>` definition. However, this mapping is done only for `<xs:element>` definitions that are not leaf nodes and have at least one attribute definition. The approach in [9] fails to consider valid semantic concepts that are simple literal definitions. In addition, ontologies have been designed so that datatype properties can be mapped to XML schema datatypes [10]. Therefore, the resulting mapping of simple `xs:element` to `owl:DatatypeProperty` using that approach would be inconsistent with standard practice of ontology design. Yang, Steele, and Lo [11] describe an ontology-based mapping between XML and ontology (bi-directional) that focuses on limiting loss of information in the bi-directional mapping.

Existing work ignores semantic information pertaining to domain relationships that are present in well-annotated XML schema annotations. By design, the purpose of XML annotations is to capture description of elements, which are often described in relation to other elements. For instance, consider the XSD annotation for the element `<xs:EventStatus>` of the C2 domain in Figure 1.

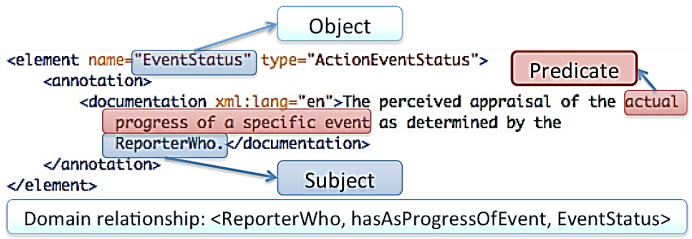

Fig. 1. Illustrating the presence of domain relationships in XSD annotations

This pattern of linking elements in annotations is common in domains with well-annotated XML schemata. Our approach leverages this pattern by employing a mapping from `xs:element` to `owl:class` and uses Part of Speech tagging of XSD annotations to extract domain relationships.

III. System Design

The mapping process takes as input a sufficiently annotated XML schema, which we define as any XML schema that contains the following:

a) The schema provides annotations for most elements using descriptive domain terminology
b) Annotations referencing elements defined in the schema use a consistent naming convention.

As an example of the latter, if the schema defines an element as "ReporterWho" then any annotation referring to this element must do so in a consistent way, i.e., the reference can be extracted by simple operations (e.g., removing spaces, pruning special characters, etc.)

The system components and relationships between the components are illustrated in Figure 2 below.

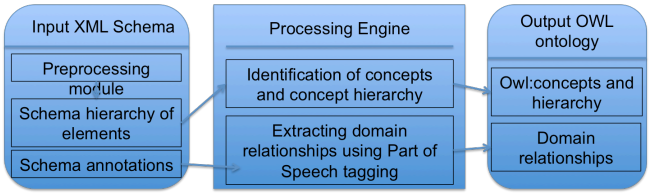

Fig. 2. Illustrating system components and relationships

Pre-processing the XML schema: Our approach maps the ontological concept name to the name of the element in the XML schema. XML schemata allow for multiple elements to have the same name. More often than not, the same name is used (e.g. `<xs:element name="ID"/>`), specially when defining identifiers and other common elements. Even though the same name may exist in different element definitions, as long as their context is different they are semantically different concepts. In order to disambiguate between elements with the same name, we propose a pre-processing step that concatenates the parent complex type name to the name of the element using a delimiter. For elements that do not have a parent complex type, an iterating place-holder name can be used (e.g. "Parent$_i$") Because XML schema rules require that all `xs:complexType` have unique names in the schema, this pre-processing step ensures that element names (and therefore concept names in the ontology) are unique in the XML schema. The pre-processing step performs the disambiguation technique to all elements, and not only to the redundant elements. This is specifically convenient because capturing the schema structure in the ontology could be useful to other ontological processes (e.g. ontology matching that uses XML schema structure). The pre-processing step is illustrated in Figures 3 and 4 below.

```
<complexType name="GDCLightType">
    <annotation>
        <documentation> Provides the coordinates for a point using the GDC system.
        </documentation>
    </annotation>
    <sequence>
        <element minOccurs="0" name="OID" type="OIDType"/>
```

Fig. 3. XML schema before pre-processing

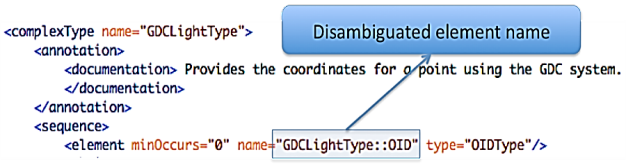

Fig. 4. XML schema after pre-processing

After pre-processing, the following mappings are established while parsing through the XML schema:

Mapping 1: 'xs:element' to owl:class: Commonly, each definition of an element contains a name and an associated complex type. An `owl:class` is created with the class name equal to the name of the element. Attribute definitions for the element are mapped to the `owl:datatype` property. Cardinality of concepts is defined according to the `xs:minOccurs` and `xs:MaxOccurs` in the XML schema as explained in [9]. Therefore, if an element "element1" is defined as having type "complexType1" and the definition of "complexType1" includes an "element2" with `max-occurs=unbounded`, then the cardinality between the concepts "element1" and "element2" is 1…∝

Mapping 2: Element hierarchy to concept hierarchy: Nayak and Wina [12] have noted that XML schema contains element definitions in a hierarchical structure. They employ structural information in XML schemata to define clusters based on semantic similarity. Varlamis and Michalis [13] note that XML schema relationships support inheritance relationships between elements. The most common method to create an inheritance relationship in a XML schema is to use `xs:extension` so that one element can extend another. This mapping step also identifies all occurences of "`abstract=true`" in the definition of an element, computing it as indication of a inheritance relationship. It should be noted that existing mapping methods ignore the presence of inheritance using of "`abstract=true`" because `xs:element` is not mapped to ontological concept.

Mapping 3: Schemata composition to 'partOf' and 'kindOf' relationships: In XML schemata an element defined as a complex type is composed of other elements. Elements can be composed using '`All`', '`Sequence`' and '`Choice`' indicators. All elements in '`All`' and '`Sequence`' groupings are mapped to a '`isPartOf`' OWL object property and all elements in the '`Choice`' composition are mapped to a '`isKindOf`' OWL object property. The following two examples illustrate the mapping from schema composition to OWL object properties.

Example 1 - Consider the definition of a Task below:

```
<xs:element name="Task" type="taskType">
```

The definition of 'taskType' is composed of other elements as follows:

```
<xs:complexType name="taskType">
 <xs:sequence>
   <xs:element  name="Who"
                type="whoType"/>
  <xs:element   name="When"
                type="whenType"/>
  <xs:element   name="Where"
                type="whereType"/>
 </xs:sequence>
</xs:complexType>
```

In the definition above, the element 'Task' is composed of 'Who', 'When' and 'Where' using the sequence composition. The mapping proposed in the paper will leverage the sequence composition to establish the following properties:

- 'Who' isPartOf 'Task'
- 'When' isPartOf 'Task'
- 'Where' isPartOf 'Task'

Example 2 - Consider the definition of a Where below:

```
<xs:element name="Where"
type="whereType>
<xs:complexType name="whereType">
 <xs:choice>
   <xs:element  name="AtWhere"
                type="AtWhereType"/>
  <xs:element   name="RouteWhere"
                type="routeType"/>
 </xs:choice>
</xs:complexType>
```

The element 'Where' is composed of 'AtWhere' and 'RouteWhere' using the choice composition. The mapping proposed in this paper will leverage the choice composition to establish the following properties:

- 'AtWhere' `isaKindOf` 'Where'
- 'RouteWhere' `isaKindOf` 'Where'

Mapping 4: Mining XSD annotations for domain relationships: Annotations in XML schemata are designed to provide documentation in the form of free text for elements being defined. It is common in the C2 domain to have annotations in XML schemata describing an element often in relationship to other elements. Existing published research in XML schemata to ontology mapping does not check for semantic relationships in XSD annotations. We propose the novel use of Part of Speech (POS) tagging to extract domain relationships from XSD annotations. Part of Speech tagging is a well-developed natural language technique that parses text and determines the part of speech for each word in the text. The common process is to determine the tag based on a probabilistic modeling of the word and its context (preceding and succeeding words). The current standard involves use of the Penn Treebank tokenization that categorizes into thirty-six possible parts of speech [14]. Extensive work has been done to identify how POS tagging can be used to determine relationships embedded in text, such as those described in [15][16][17][18]. Wang, Ting, et al. [19] use a support vector method and accompanying relationship ontology to determine semantic relationships embedded in text. The following steps are used to map XSD annotations to domain relationships (`owl:objectProperty`):

Step1: Identifying all concepts in the annotation: This is done by identifying all words tagged as nouns or proper nouns (NN, NNS, NNP, NNPS) by the POS tagger. The concepts are added to a vector as follows:

$$V_{concepts} = \{C_i \mid C_i \text{ is the } i^{th} \text{ concept in the annotation}\}$$

Step 2: Identifying predicates for the relationship: Starting at the beginning of the annotation, a concatenation of adjectives (JJ), Pronouns (WP), and prepositions (IN) is created until concept C_i is encountered. This concatenation forms the predicate of the domain relationship. These predicates are added into the vector as:

$$V_{predicates} = \{pred_i \mid pred_i \text{ is the concatenated predicate before the } i^{th} \text{ concept in the annotation}\}$$

Each concept C_i will now have an accompanying predicate. If $pred_i$ has only one word and is a coordinating conjunction (e.g. "and") then $pred_{i-1}$ is assigned to $pred_i$. This is due to the presence of a coordinating conjunction between

two concept names C1 and C2, meaning that whatever predicate applied to C1 also applies to C2.

Step 3: Creating the domain relationship: If the annotation is for element E1, then for each concept in $V_{concepts}$ the following domain relationship (`owl:objectProperty`) is created:

```
<subject, predicate, object> =
< C_i, str_concat(hasAs,pred_i), E1  >
```

```
<element name="TaskWhatRef" type="TaskWhatRefType">
    <annotation>
        <documentation xml:lang="en">Specifies a reference to task.</documentation>
    </annotation>
</element>
```

Fig 5. An example schema element with annotation

Based on the technique described above, the following domain relationship is created:

```
<Task hasAsReferenceTask TaskWhatRef>
```

Note: For the sake of clarity of illustration the schema as it appears before pre-processing is shown in Figure 5.

The mappings described in the steps above are illustrated in Figure 6.

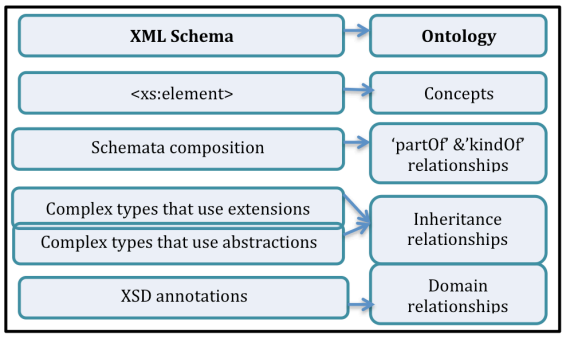

Fig. 6. Mapping from XML schema components to ontology components

IV. Application to C2SIM.

C2-to-simulation interoperability (C2SIM) is a standard under development by the Simulation Interoperability Standards Organization (SISO) to facilitate interoperability between C2 and simulation systems [20]. The current phase of standardization effort involves work on developing a formal semantic model as part of a model-driven framework. The goal is to use C2SIM to interoperate between multiple C2 and simulation systems.

In order to provide interoperability on the semantic level, C2SIM will require ontological support for formalizing semantics and for the design and analysis of networked C2 and simulation systems. Early work on the need for and future of semantic C2SIM is described in [21]. C2SIM development is based on complex XML schemata that have been developed in Phase 1 standardization effort of C-BML [3] and MSDL [4]. These XML schemata have been found to be complex [22] because the design intended to capture the full expressivity of the underlying sophisticated data model. Adopting a manual process to create an ontology from these schemata can be tedious and error-prone. The method proposed in Section 3 has been applied to the C-BML Phase1 schema [3]. Statistics of the XML schema used to create the draft ontology are presented in Table 1.

TABLE 1. STATISTICS REGARDING C2SIM XML SCHEMA

Metric	Value
Number of complex type definitions	531
Number of element definitions	1115
Number of annotations	1604
Number of unbounded elements(Number of elements that have "maxoccurs=unbounded")	40
Fanning Index [23](Number of relationships/number of elements)	9.67
ComplexityMeasure (based on the formula in [23]):	738

V. Preliminary Results and Discussion

A software prototype was built, based on the proposed method using OWL-API [24] to create the draft ontology.

The pre-processing step, described in section 2, disambiguated element names so that concepts can be accurately mapped to elements in the XML schema. The draft ontology created by this method captures a conceptual hierarchy consistent with an intuitive understanding of C2SIM. The domain relationships are descriptive and useful for capturing business rules. The statistics of the ontology created are shown in Table 2. At the time of this writing, we are conducting an evaluation of the draft ontology to validate these preliminary results. The evaluation involves the use of subject matter experts (SMEs) to evaluate the draft ontology by checking it against domain documents and their own expertise. The initial results, while still anecdotal, suggest that the resulting ontology is consistent with SME evaluation of domain documents.

TABLE 2. C2SIM ONTOLOGY METRICS

Ontology metric	Value
Number of Concepts	1115
Number of Inheritance relationships	765
Number of domain relationships	73

Figures 7, 8 and 9 provide snapshots of the ontology as viewed in the ontology editor Protégé 4.3 [25].

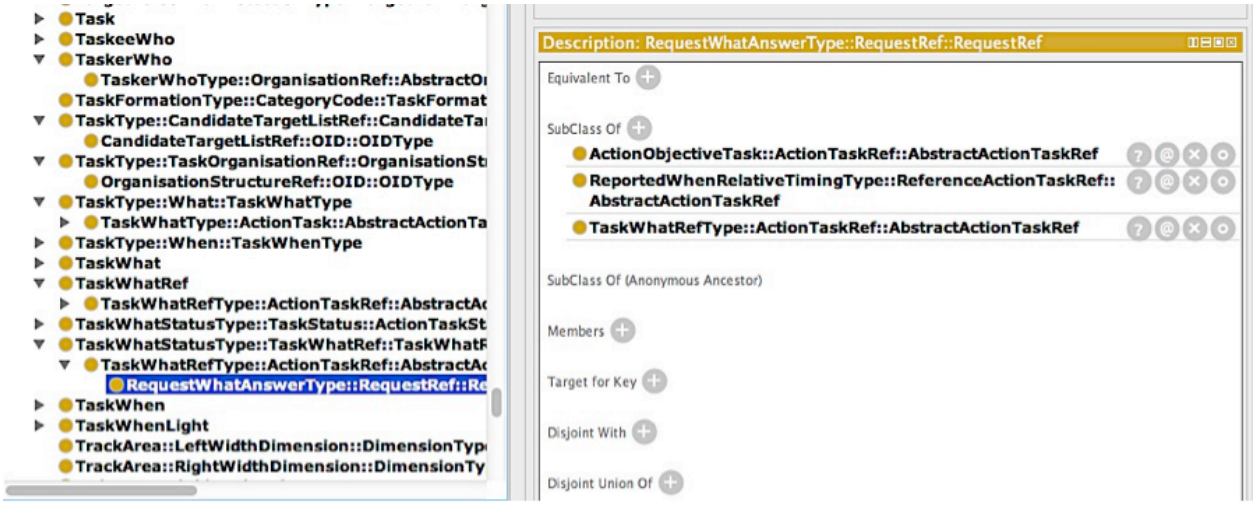

Fig. 7. Snapshot of class hierarchy in C2SIM ontology

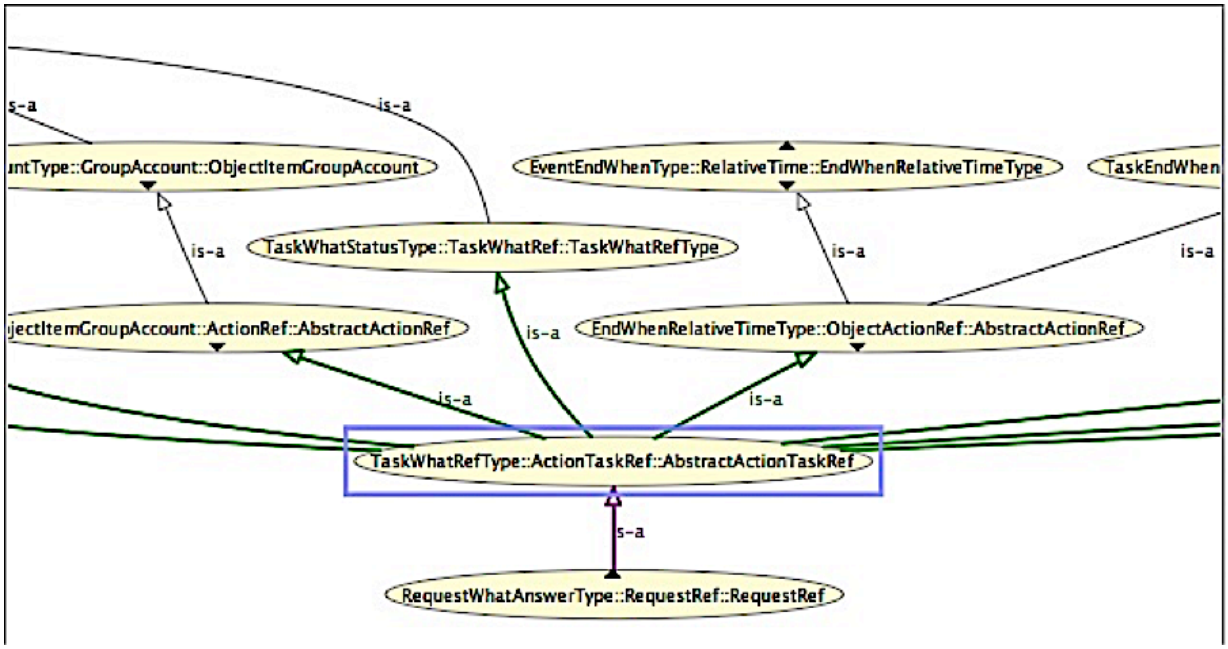

Fig. 8. Snapshot of C2SIM ontology subset visualized in OWL Viz.

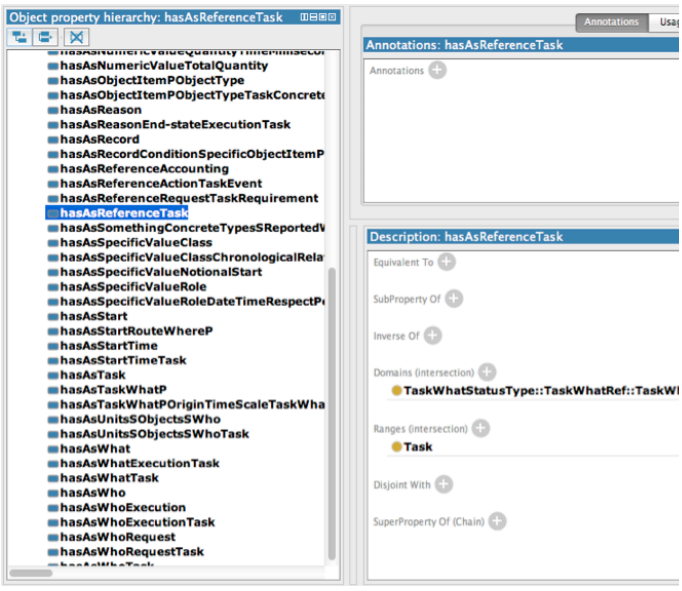

Fig. 9. Snapshot of domain relationships in C2SIM ontology extracted through POS tagging of XML schema annotations

VI. Future Work

The proposed method extracts domain concepts and relationships from well-annotated XML schema. Potential approaches to improve the quality and resolution of the resulting ontology include the use of domain synonym tables as a means to support identifying concept names in schema annotations. We believe this can account for variants of a name that may be used in the schema annotation. Future work improving our methodology also includes capturing C2 doctrine in the form of axioms, as well as evaluating the use of reasoning to both infer and hypothesize knowledge. In ongoing work [26], we are also investigating the use of structural information in XML schemata to perform ontology matching between two XML based ontologies. Ontology matching of two complex ontologies that have accompanying complex schemata suffers from high computational cost. Finally, we are exploring a tighter coupling between ontology creation and ontology matching by embedding basic XML schema structure in auxiliary ontology artifacts (e.g. annotations).

REFERENCES

[1] Adda, Mehdi. "A Pattern Language for Knowledge Discovery in a Semantic Web context." *Models for Capitalizing on Web Engineering Advancements: Trends and Discoveries: Trends and Discoveries* (2012): 59.

[2] Li, Luochen, et al. "Discovering semantics from data-centric XML." Database and Expert Systems Applications. Springer Berlin Heidelberg, 2013.

[3] Blais, C., Brown, D., Chartrand, S., Diallo, S., Heffner, K., Levine, S., Singapogu, S., St-Onge, M., and Scolaro, D.: "Coalition Battle Management Language (C-BML) Phase 1 Information Exchange Content and Structure Specification," Paper 10S-SIW-002, Proceedings of the Spring Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, Orlando, FL, April 2010.

[4] Simulation Interoperability Standards Organization. 2008. Standard for: Military Scenario Definition Language. SISO-STD-007-2008, 14 October.

[5] https://www.niem.gov/ (last viewed: September 10 2015)

[6] Beardsworth, Robert, et al. "XML standards as the basis for data interoperability among military C2 systems and beyond." MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010. IEEE, 2010.

[7] Matthias Ferdinand, Christian Zirpins, and D. Trastour. Lifting XML Schema to OWL. In Nora Koch, Piero Fraternali, and Martin Wirsing, editors, Web Engineering - 4th International Conference, ICWE 2004, Munich, Germany, July 26-30, 2004, Proceedings, pages 354–358. Springer Heidelberg, 2004.

[8] Bedini, I., Gardarin, G., and Nguyen, B. Deriving Ontologies from XML Schema. In Proceedings 4émes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2008). Invited paper. 5 - 6, June 2008 Toulouse, France

[9] Bohring, Hannes, and Sören Auer. "Mapping XML to OWL Ontologies." Leipziger Informatik-Tage 72 (2005): 147-156.

[10] http://www.w3.org/TR/2004/REC-owl-guide-20040210/#Datatypes1 (last viewed: September 10, 2015)

[11] K. Yang, R. Steele, A. Lo, '"An ontology for XML schema ontology mapping representation"' *in Proceedings of the 9th International Conference on Information Integration and Web-based Applications & Services (iiWAS 2007)* (2007)

[12] Nayak, Richi, and Wina Iryadi. "XML schema clustering with semantic and hierarchical similarity measures." Knowledge-Based Systems 20.4 (2007): 336-349.

[13] Varlamis, Iraklis, and Michalis Vazirgiannis. "Bridging XML-schema and relational databases: a system for generating and manipulating relational databases using valid XML documents." Proceedings of the 2001 ACM Symposium on Document engineering. ACM, 2001.

[14] http://www.comp.leeds.ac.uk/ccalas/tagsets/upenn.html (last viewed: September 10 2015)

[15] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

[16] Ruiz-Casado, Maria, Enrique Alfonseca, and Pablo Castells. "Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia." Data & Knowledge Engineering 61.3 (2007): 484-499.

[17] Medelyan, Olena, et al. "Mining meaning from Wikipedia." International Journal of Human-Computer Studies 67.9 (2009): 716-754.

[18] Sánchez, David. "A methodology to learn ontological attributes from the Web."Data & Knowledge Engineering 69.6 (2010): 573-597.

[19] Wang, Ting, et al. "Automatic extraction of hierarchical relations from text." The Semantic Web: Research and Applications. Springer Berlin Heidelberg, 2006. 215-229.

[20] Pullen, J. Khimeche, L. "Advances in Systems and Technologies towards Interoperating Operational Military C2 and Simulation Systems", 19th International Command and Control Research and Technology Symposium (ICCRTS). Alexandria, VA, 2014.

[21] Singapogu, Samuel. "Opportunities for Next Generation BML: Semantic C-BML". 19th International Command and Control Research and Technology Symposium (ICCRTS). Alexandria, VA, 2014.

[22] Abbott, Jeff, J. Pullen, and Stan Levine. "Answering the Question: Why a BML Standard Has Taken So Long to Be Established?." *IEEE Fall Simulation Interoperability Workshop, Orlando FL*. 2011.

[23] McDowell, Andrew, Chris Schmidt, and Kwok-bun Yue. "Analysis and Metrics of XML Schema." Software Engineering Research and Practice. 2004.

[24] http://owlapi.sourceforge.net/ (last viewed: September 10 2015)

[25] http://protege.stanford.edu/ (last viewed: October 25 2015)

[26] Singapogu, Samuel Suhas. "Ontology Matching Using Structure and Annotations in XML Schema". 20th International Command and Control Research and Technology Symposium (ICCRTS), 2015.