2006

A case study in applying semantic web technologies to the XML-based tactical assessment markup language (TAML)

Childers, Candace

FS-6-06-002

A Case Study in Applying Semantic Web Technologies to the XML-Based Tactical Assessment Markup Language (TAML)

Candace Childers*, Don Brutzman*, Curtis Blais*, Paul Young+

* Naval Postgraduate School; + United States Naval Academy
* Monterey, CA; + Annapolis, MD
cmchild1@nps.edu, brutzman@nps.edu, clblais@nps.edu, pyoung@usna.edu

Abstract

The ability to analyze data quickly and transform it into actionable information is vital for information superiority. However, the amount of available data is increasing and the time to make decisions is decreasing. There is too much data for humans to sift through and filter for decision making, so computer automation is necessary.

The Extensible Markup Language (XML) offers a partial solution by providing a syntactic standard for data exchange. The Tactical Assessment Markup Language (TAML) is an XML vocabulary for exchanging undersea warfare tactical data. However, the meaning or semantics of the data is unknown to the machine processing the data.

The Semantic Web is a set of technologies designed to add semantic information to data for machine processing. The technologies consist of several components, including a common syntax for data exchange, common semantic representation, and a common ontology language. Reasoning engines also apply algorithms to the data to infer useful information and present it to decision makers. Sophisticated Semantic Web tools and techniques are rapidly emerging.

This paper provides a case study in adding stronger semantic content through application of Semantic Web technologies to XML-based languages such as TAML. The lessons learned will help enable systems to extract useful, actionable information from a number of distributed, autonomous, heterogeneous information sources and bring the armed forces closer to a knowledge-aware Global Information Grid (GIG).

Problem Overview

In the military, as in business, the ability to analyze data quickly and transform it to obtain actionable information is vital for information superiority. However, the amount of available information is increasing and the time to make decisions is decreasing. There is too much data for humans to sift through and filter for decision making, so computer automation is necessary. Incompatibilities in data models, lack of facilities for capturing the meaning or semantics associated with such data, and a lack of means for processing semantic information hamper the ability to automatically process the quantities of data into information that is vital for conducting military or business operations.

The Semantic Web is a set of technologies designed to add semantic information to data for machine processing. The technologies consist of a common syntax for data exchange, a common semantic representation, and a common ontology language. The Extensible Markup Language (XML) provides the common syntax for data exchange (Hunter et al. 2003). The Resource Description Framework (RDF) provides a common semantic format for representing the relationships between resources or real-world entities (Hjelm 2001). The Web Ontology Language (OWL) is used to build models that explicitly define the concepts in a domain (Lacy 2005). Finally, reasoning engines can be applied to the data in an ontology to infer useful information and present it to decision makers (Pan 2004).

The Global Information Grid (GIG) is the Department of Defense's net-centric information environment of the future (Winters and Tolk 2005). The GIG is expected to provide valuable real-time decision making information on demand to operational commanders to achieve information superiority. The GIG will provide a global net-centric system for processing, storing, managing, and transporting information to support the DoD in peacetime and during times of conflict.

Information superiority is the capability to collect, process and analyze a flow of information while denying an enemy's ability to do the same. Information superiority is vital in current battle spaces. Surveillance and intelligence technologies have made data abundant, but information superiority still faces major challenges in exchanging data among different sources and in analyzing the overwhelming amount of data available to provide a richer picture of the battle space (Hayes-Roth 2005).

In our research (Childers 2006), summarized in this paper, we investigate how available Semantic Web technology can be exploited to address the information superiority challenge. We first select a target domain where Semantic Web technologies can be applied to enhance information superiority. Then we investigate the application of those technologies to the data exchange and data analysis problems, providing examples for converting an existing XML document to RDF, for creating an ontology for the target domain and for applying rules to the ontology for reasoning about the data. Finally, we look at the lessons learned and assess the potential of the Semantic Web for extracting useful, actionable information from a number of distributed, autonomous, heterogeneous information sources.

Selecting a Target Domain for Evaluation of Semantic Web Technologies

In targeting a domain for implementation of Semantic Web technologies within the GIG, ongoing work of the Undersea Warfare XML working group (USW-XML WG) led to the selection of the Undersea Warfare community as the target domain. The USW-XML WG is an open working group registered as a Department of Defense (DoD) Community of Interest (COI) working to improve interoperability in the Undersea Warfare community by designing common XML vocabularies for data exchange (Brutzman and Grimley 2006).

The USW-XML WG produced its first XML vocabulary, the Tactical Assessment Markup Language (TAML), for representing own ship and target tracking information. TAML is expected to benefit command and control systems such as the USW Decision Support System, Anti-Submarine Warfare Tactical Assessment System and the Carrier Tactical Support Center.

Adding Semantic Web technologies to XML languages such as TAML should bring the armed forces closer to information superiority and a knowledge-aware Global Information Grid (GIG). This will be accomplished by providing the mechanism whereby the overwhelming amount of data available can be analyzed more effectively to provide a richer picture of the operational space.

Applying Semantic Web Technologies to TAML

We next look at the process for applying Semantic Web technologies to TAML as a step toward developing applications that will help achieve information superiority. First, we address the issue of providing a common framework for data exchange in the GIG. Then we look at adding Semantic Web technologies to that framework to enable automated analysis of the vast quantities of data to provide actionable information to the operational commander.

Using XML for Data Exchange in the GIG

Central to the strategy for enhancing system interoperability and thereby achieving information superiority is the use of XML for data exchange throughout the GIG. XML provides a framework for describing and structuring data without restricting the terminology. The syntax of XML vocabularies is defined unambiguously which allows many processors to consistently parse and analyze all documents which conform to the associated schema. Data interoperability will support data sharing in the Undersea Warfare domain and is expected to improve command and control, exercise assessment, operational analysis, modeling, simulation and tactical innovation.

The USW community uses various weapon and information systems to gather information about a battlespace to make tactical decisions. Many of these systems are stovepiped systems that were not developed with interoperability in mind. The current approach to achieving interoperability among systems is to hard-code one-to-one mappings between systems, but the number of mappings needed grows geometrically with the number of systems. TAML provides a standard XML tagset for exchanging tactical messages between systems. Each system only needs to map to and from TAML for use as a common exchange language. Using an intermediate language such as TAML as an external model reduces the number of mappings needed from N(N-1) to 2N.

Adding Semantics to TAML to Enable Automated Data Analysis

The TAML schema provides common terminology and a common syntax for representing tactical information but does not provide machine-interpretable semantics for the domain. A TAML Contact Classification Ontology was created to evaluate the expressive capability of OWL-DL (OWL constrained to Description Logic constructs), the use of tools for creating an ontology, and the capability of reasoning engines to use facts contained in the ontology to deduce new information about the domain. During the development and testing of the TAML Contact Classification Ontology, several limitations of OWL-DL were exposed that will be discussed below.

The Semantic Web Rule Language (SWRL) was used to add rules to the ontology in order to overcome some of these limitations (SWRL 2006). SWRL is currently in the World Wide Web Consortium (W3C) Recommendation Phase and thus is not yet a complete specification. Although SWRL rules were defined for the ontology, current Semantic Web reasoners are unable to interpret and execute them due to the relative newness of the language. However, due to the increasing adoption of Semantic Web technologies, reasoner support is improving rapidly.

During the development of the TAML Contact Classification Ontology, we investigated a number of ontology modeling tools and reasoning engines, focusing on Protégé-OWL (Protégé 2006) and RacerPro (RacerPro 2006) for evaluating the expressive capability of the TAML Ontology and the reasoning capability of the reasoning engine when applied to a tactical problem. The OWL ontology modeling tool, Protégé-OWL, provides a framework for explicitly defining the semantics of a domain and provides a plugin architecture for building applications around an ontology. Protégé-OWL hides the specific syntax of the OWL tags and instead presents a graphical user interface for defining concepts and properties. The RacerPro reasoning engine is used to ensure the ontology is consistent and to infer information about the instance data input into the ontology.

Converting TAML to RDF

The amount of tactical data available in the armed forces is staggering and requires automation in order to be processed and queried quickly. RDF provides a powerful mechanism for cataloging, retrieving and querying data (Powers 2003). RDF enhances XML by representing data with explicit semantics. Serializing TAML as RDF adds semantic information to TAML, enabling increased automation for finding and querying data while maintaining the interoperability advantages of XML.

The serialization of TAML documents as RDF/XML allows for the explicit definition of the relationships between resources in a machine-interpretable format. Once TAML is serialized as RDF/XML the data can be inserted into any document without losing any of its meaning since the meaning is no longer defined by the structure of the document.

The RDF/XML serialization of data enables data from several different domains to be combined and queried as a single data source. A user querying the data needs information about the names of the resources or properties being queried but does not need any knowledge about the ordering of the data within the document.

Representing TAML Resources. TAML documents are tactical messages that describe events like tracking operations and entities like platforms and contacts. The events and entities are described as elements within the TAML instance documents and become resources in the RDF/XML serialization. Within the TAML vocabulary, *id* attributes are used to uniquely identify entities or elements like *Operation*, *Event*, *Platform*, and *Contact* as shown in Figure 1.

```
<Operation id="T001">
   <Name>Trafalgar</Name>
</Operation>
```

Figure 1. Fragment of TAML XML illustrating the use of id attributes.

Unique names must be created for the information contained in the RDF/XML document from the existing information in the TAML XML document. Figure 2 shows a subset of a TAML RDF/XML document which demonstrates how a unique name was created for operations, platforms, and other elements identified in TAML with an id attribute.

```
<rdf:Description
rdf:about="http://usw.xml.wg/
Trafalgar.xml/OperationT001">
   <taml:Name>Trafalgar</taml:Name>
</rdf:Description>
```

Figure 2. Fragment of TAML RDF demonstrating how unique names are created for operations.

The entity type in the subset above is Operation and the TAML *id* is T001. The unique name identifying the entity is a concatenation of the file pathname, the entity type, and the *id* number. The file pathname ensures the name is unique even if an operation in another TAML instance document has the same TAML *id* attribute. This pattern of forming entity names by concatenation is maintained for all elements identified with *id* attributes within the TAML vocabulary.

TAML does not use *id* attributes to uniquely identify all elements. For example, in Figure 3, *Configuration* and *Track* are child elements of element *Platform* and do not contain their own TAML *id* attribute, relying on Platform's *id* instead. Figure 4 shows the convention for creating unique names for elements that do not have *id* attributes.

```
<Platform id="HMSFLAG">
   <Name>Victory</Name>
   <Configuration>
      <ConfigurationItem id="CFGFLAG">
         100 Guns
      </ConfigurationItem>
      <ConfigurationItem id="CMDFLAG">
         Capt. T. M.Hardy
      </ConfigurationItem>
   </Configuration>
   <Track>...</Track>
</Platform>
```

Figure 3. Fragment of TAML XML showing child elements as properties of their parent element.

The patterns used to name the specific resources described within the TAML instance documents ensure the names are unique internally and externally to the document. The patterns also add consistency to the naming convention to make querying resources simpler for applications.

```
<rdf:Description
   rdf:about="http://usw.xml.wg/
      Trafalgar.xml/PlatformHMSFLAG">
   <taml:Track
      rdf:resource="http://usw.xml.wg/
      Trafalgar.xml/TrackPlatformHMSFLAG"
   />
</rdf:Description>
```

Figure 4. Fragment of TAML RDF showing how unique names are created for elements without unique id attributes.

Representing TAML Predicates. Many of the elements within TAML are used to describe their parent elements. Any element that describes its parent becomes a predicate in the RDF/XML file. Figure 5 illustrates a series of child elements describing the element *Contact*.

```
<Contact id="x20" sensorCode="Active">
   <DateTimeGroup>
      <DateTime>
         2005-05-12T14:00:00Z
      </DateTime>
   </DateTimeGroup>
   <Position>
      <AbsolutePosition>
         <Latitude>
            22.12345
         </Latitude>
         <Longitude>
            -121.123456
         </Longitude>
      </AbsolutePosition>
   </Position>
   <Course>0</Course>
   <Speed>0</Speed>
</Contact>
```

Figure 5. Fragment of TAML XML document showing child elements which describe element Contact.

The elements *DateTimeGroup*, *Position*, *Course*, and *Speed* are characteristics of the Contact with an *id* of *x20*, so each of these elements becomes the predicate of a triple describing the resource *ContactIDx20* as shown in Figure 6. The *taml* prefix references the TAML namespace which is used to uniquely identify the TAML vocabulary elements. The prefix precedes each property, since the property terms are defined in the TAML vocabulary. The XML also includes the attribute *sensorCode*, which describes *ContactIDx20*, so it also becomes a predicate in the RDF/XML serialization.

The above steps are duplicated to create the predicates for each resource within the RDF/XML document. The TAML vocabulary acts as a prototype for the RDF semantics. Since the semantics of the TAML vocabulary are not defined by the schema, the meaning of the elements has to be interpreted by a human and translated to a set of RDF triples. The RDF/XML file states the parent/child relationships of the TAML hierarchy as explicit property relationships, which are understandable by machines. Predicates or properties are the linking mechanism used to illustrate relationships between objects and other objects or between objects and literal values.

```
<rdf:Description
   rdf:about="http://usw.xml.wg
   /TAMLExample.xml/ContactIDx20">
   <taml:sensorCode>
      Active
   </taml:sensorCode>
   <taml:DateTime>
      2005-05-12T14:00:0Z
   </taml:DateTime>
   <taml:Position>
      Absolute Position
   </taml:Position>
   <taml:Latitude>
      22.12345
   </taml:Latitude>
   <taml:Longitude>
      -121.123456
   </taml:Longitude>
   <taml:Course>0</taml:Course>
   <taml:Speed>0</taml:Speed>
</rdf:Description>
```

Figure 6. Fragment of TAML RDF showing child elements as properties of their parent class *Contact.*

Representing TAML Objects. The object of an RDF/XML triple is either a literal value or another resource with a unique identifier. The RDF/XML serialization of TAML contains objects that are literals and objects that are resources. The objects that are also resources are further described within the RDF document by their own triples. When the document is parsed, these objects are linked to the statements describing them by creating a graph structure of logical chains where the object of one triple is the subject of another triple.

The TAML Schema defines data types for the data stored within certain elements. The elements that store text or data are redefined as properties describing their parent element in the TAML RDF/XML serialization. The data stored within the element is redefined as the object of the property with a literal value. Figure 7 illustrates converting TAML XML element data to RDF literal objects. The objects are highlighted in bold text to illustrate the consistency within the change. In the RDF/XML serialization below, the triple object remains a literal value.

```
<Event id="x2">
   <StartTime>
      2005-05-12T14:00:00Z
   </StartTime>
   <EndTime>
      2005-05-13T14:00:00Z
   </EndTime>
   <Name>Event x2</Name>
</Event>
```

```
<rdf:Description
rdf:about="http://usw.xml.wg/
   TAMLExample.xml/Eventx2">
   <taml:StartTime>
      2005-05-12T14:00:00Z
   </taml:StartTime>
   <taml:EndTime>
      2005-05-13T14:00:00Z
   </taml:EndTime>
   <taml:Name>Event x2</taml:Name>
</rdf:Description>
```

Figure 7. Comparison of XML serialization (above) and RDF serialization (below) of literal objects.

Within the TAML vocabulary, many elements describe other elements. In RDF the relationship between these elements is explicitly defined by a triple. The parent element is the subject of the triple and the child element is the object of the triple. The property of the triple is the relationship between the two elements.

The name of the child element describes the relationship; therefore, it becomes the predicate of the triple as shown in Figure 8. The triple in Figure 8 states that EventX2 has a PlatformRef with the name http://usw.xml.wg/TAMLExample.xml/Platformx5. The object of the triple is the specific entity that is related to the subject. Since the object is a specific entity with a unique identifier, it is further defined within the document. The ability of RDF to define resources as objects of statements allows for linkage of entities throughout the document and creates complete semantics. When the document is parsed, the linkages are mapped out to allow quick processing, understanding, and querying of the document.

```
<Event id="x2">
   <PlatformRef idref="x5"/>
</Event>
```

```
<rdf:Description
   rdf:about="http://usw.xml.wg/
      TAMLExample.xml/Eventx2">
   <taml:PlatformRef
   rdf:resource="http://usw.xml.wg/
      TAMLExample.xml/Platformx5" />
</rdf:Description>
```

Figure 8. Illustration of a resource as an object in TAMLExample.rdf.

TAML documents serialized as RDF/XML can also provide instance data for a knowledge base which can be queried or reasoned upon. An ontology is used to add context to the data explicitly defined in the RDF/XML document.

Creating a TAML Ontology

Once data is explicitly defined, machines still need context information about the domain in order to understand how to process the data to obtain valuable information. An ontology was created for a subset of the TAML vocabulary focusing on contact information. The TAML Contact Classification Ontology provides a semantic model of a *Contact* where each statement about a *Contact* has only one explicit interpretation, thus reducing ambiguity and enhancing machine interpretability. The ontology, along with the RDF/XML TAML documents, increases automation by providing a machine-interpretable knowledge base.

The original TAML XML Schema provided the concept and property names which are explicitly defined by the TAML Contact Classification Ontology. Human interpretation was applied to the schema to determine which elements represented concepts and which elements represented the properties defining those concepts. The ontology focuses on explicitly defining the concept of a *Contact* by defining the properties a *Contact* may have and by dividing the concept *Contact* into five different subclasses: *HostileContact*, *FriendlyContact*, *NeutralContact*, *SuspectContact* and *UnknownContact*.

After defining the major concept, *Contact*, each of its subclasses was defined as shown in Figure 9. For example, one definition of *HostileContact* is any *Contact* which is a military aircraft and has a hostile country code. If a Semantic Web reasoner encounters an instance of *Contact* which meets the criteria of any definition it will sub-classify the *Contact* as a member of the corresponding class. The subclassification of a *Contact* is inferred information about the data which is added automatically to the asserted information. Machines are able to sift through high volumes of data and classify instances more quickly than humans. Similar definitions are added for each subclass of *Contact* to create a proof of concept for classification reasoning using the Semantic Web.

The ontology was tested by populating individual instances with data that met the criteria for each definition written. An example test case defines the country code of a

Contact as *CA* (a friendly country code) and the target type as submarine. The classification of the test case was correctly inferred by RacerPro as *FriendlyContact*.

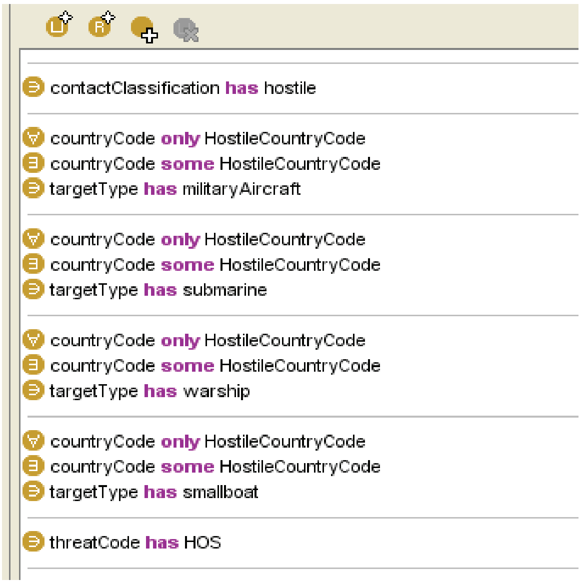

Figure 9. Definition of a Hostile Contact. The above screenshot shows the definition of a HostileContact within the Protégé ontology editor.

The design and testing of the TAML Contact Classification Ontology exposed limitations of OWL-DL. The OWL-DL language limits the type of restrictions that can be used to define classes in order to guarantee decidability. The current version of OWL-DL only supports two XML Schema datatypes: *xsd:string* and *xsd:int*. Therefore, the range value of datatype properties cannot be further restricted through the use of user-defined datatypes. For example, there is no OWL construct for stating the speed of the contact is greater than five knots using the current version of OWL-DL. Future versions of OWL-DL are expected to support user-defined datatypes.

Another significant limitation of OWL-DL is the lack of support for defining properties through constraints and relationships. This area of domain definition is addressed by rule languages. Thus SWRL rules were added to the TAML Contact Classification Ontology in order to overcome some of the limitations of OWL-DL.

Using SWRL to Add Rules to the TAML Contact Classification Ontology

SWRL adds the ability to write Horn-like rules for a domain in terms of the concepts and properties defined in the ontology. Rules add new capabilities to ontologies including rule-based reasoning and expressive-instance querying. The main goal of SWRL is to provide a semantically coherent way to create rule-bases that allow machines to reason about a domain defined by an ontology (Grosof 2003). Thus the upper levels of the Semantic Web stack interoperate in complementary ways.

The SWRL rules added to the TAML Contact Classification Ontology overcome some of the limitations of OWL and enhance the ontology by adding the capability to prioritize instances of *HostileContact*.

Datatype Range Restriction Rules. The lack of support for user-defined datatypes is a serious limitation of OWL-DL when building an ontology of the TAML domain. The TAML XML Schema includes restrictions on the values of certain elements and there is a need to enforce these restrictions within the knowledge base.

SWRL provides a method for enforcing range restrictions on properties. A new Boolean property, *rangeViolationError*, was created to highlight when an instance property value is outside the allowed range. Rules defining unacceptable values were defined and when the rules are executed by a rule engine, the *rangeViolationError* property field for all violating instances is set to "true". For example, if an instance has a course value which is greater than 360, then the *rangeViolationError* property for that instance is set to "true".

Once the rules have been executed, a query can be written to return all of the instances with unacceptable values since this may indicate an error in the instance data.

Checking Semantic Consistency with SWRL. The TAML XML schema is limited to validating the syntactic components of TAML documents, thus there is a need for using the ontology of a domain to validate their semantic consistency.

A new boolean property, hasSemanticError is defined and SWRL rules set the property to true when certain conditions indicating semantic inconsistencies exist in the knowledge base. For example, the following SWRL rule:

```
Contact(?x) ^ threatCode(?x, ?fc)
^ contactClassification(?x, ?cc)
^ swrlb:equal(?fc, "HOS")
^ swrlb:notEqual(?cc, "hostile")
→ hasSemanticError(?x, true)
```

indicates a semantic inconsistency since a contact cannot have a *threatCode* value of "HOS" (hostile) and a *contactClassification* value equal to something other than "hostile". If this condition exists within an instance, then the *hasSemanticError* property for that instance is set to "true". A query can then be used to extract all instances with semantic errors.

Prioritizing Hostile Contacts with SWRL. The SWRL language includes a set of predefined predicates or built-ins which increase the expressivity of the SWRL language (SWRL BIS 2006). The built-ins are divided into several different types of functionality including comparison, math, boolean, string, date/time/duration, URIs and list built-ins.

SWRL built-ins are used in the TAML Contact Classification Ontology to further prioritize contacts based on their speed and their distance from the detecting platform. The closer a contact is and the faster it is moving, the higher the threat level assigned. The comparison predicates are used to specify which speed and range values indicate certain levels of threat. For example, the following rule:

```
HostileContact(?x) ^ speed(?x, ?speed)
^ range(?x, ?range)
^ swrlb:greaterThanOrEqual(?speed, 10)
^ swrlb:lessThanOrEqual(?range, 10000)
→ isImmediateThreat(?x, true)
```

indicates that a contact with a speed greater than or equal to 10 (knots) and within 10000 (yards) is an immediate threat. Therefore, if instance A has a *speed* value of 11 and a *range* value of 8000, then when the rule is executed, the *isImmediateThreat* property for A is set to "true". The addition of prioritization rules to the ontology enhances the type of automated reasoning that can be accomplished. More semantic information can now be added to the knowledge base and presented to the warfighter.

Same as/different from rules. SWRL includes predicates for indicating that two instances refer to the same real world entity and for indicating that two instances refer to different entities (O'Conner 2005). These predicates are useful for determining if two instances detected from different platforms are referring to the same Contact. For example, the following rule:

```
Contact(?x) ^ Contact(?y)
^ position(?x, ?posx)
^ position(?y, ?posy)
^ swrlb:equal(?posx, ?posy)
^ dateTime(?x, ?dtx)
^ dateTime(?y, ?dty)
^ swrlb:equal(?dtx, ?dty)
→ sameAs(?x, ?y)
```

indicates that if two contact instances occupy the same position at the same time, then the two instances represent the same real world entity. The rule indicates this in the knowledge base by using the *sameAs* construct to define the two instances as the same entity. This rule can be used to consolidate instances and reduce the data presented to the warfighter.

The current rules defined in the ontology require the positions to be exactly the same. However, the rules can be enhanced by taking errors and small position/time differences into account if the accuracy of the detecting systems is known. The SWRL math built-ins provide the capability to write rules which take into account certain amounts of error. This is a valuable task for future work.

Lessons Learned and Conclusions

In this paper we investigated how available technology might be exploited to address the information superiority challenge. We first selected a target domain where the application of Semantic Web technologies can be used to enhance information superiority. Then we investigated the application of those technologies to the data exchange and data analysis problems, providing examples for converting an existing XML document to RDF, for creating an ontology for the Target Domain and for applying rules to the ontology for reasoning about the referenced data.

We chose the Tactical Assessment Markup Language (TAML), used for representing own ship and target tracking information, as the target for applying Semantic Web technology to enhance information superiority. TAML provides a common syntax for the exchange of tactical data within the Undersea Warfare community. We added semantics to TAML by converting the TAML XML to RDF in order to extract information useful to the operational commander from the data being exchanged between systems.

RDF enhances XML by representing data with explicit semantics. RDF provides an explicit language for representing data or facts but does not provide any means for adding context to the data. Machines need context about information in order to process and infer information from the facts presented by the data.

The TAML Contact Classification Ontology explicitly defines the concept of a Contact within the TAML domain. The goal of the ontology is to accept Contact instance data from a TAML document and then use a reasoner to infer the classification of the Contact. The reasoner was able to correctly classify instance data based on the definitions; however, several limitations of OWL and implemented reasoners were exposed during this development.

The current version of OWL-DL only supports XML schema *xsd:int* and *xsd:string* datatypes. OWL-DL does not support other XML schema (xsd) datatypes or user-defined datatypes (Pan 2004). This lack of support creates a problem when trying to input TAML documents into the ontology, since TAML documents do contain other datatypes including user-defined datatypes.

OWL also does not support placing restrictions on datatype properties (Pan 2004). In addition, current DL reasoners and the interface used between Protégé and current reasoners have limitations which affect the level of reasoning which is possible using the TAML Contact Classification Ontology. Cardinality restrictions on datatype properties are ignored during reasoning due to a limitation in the Description Logic (DL) Implementation Group (DIG) interface between Protégé and RacerPro. OWL has several limitations when used to model a problem without the use of SWRL rules. The use of OWL alone is limited to simple classification problems which do not require restrictions on datatype properties and do not use mathematical functions to determine results. In the case of the TAML Contact Classification Ontology, SWRL is needed to overcome these limitations of OWL.

SWRL effectively overcomes some of the limitations encountered in the OWL-only TAML Contact Classification Ontology, but software-implementation support for SWRL remains limited. SWRL provides logic to the Semantic Web but there are some types of reasoning which are not yet possible such as associative thinking, spatial reasoning, recognition of images, and complex decision procedures (Herman 2004).

Increased machine automation for the processing and analysis of data is vital to net-centric warfare. The Semantic Web provides a methodology for machine automation which can significantly enhance the information available to warfighters for making decisions.

Semantic Web languages and tools are still being developed and refined. However, the Semantic Web benefits from the widespread adoption of XML and a growing number of early adopters such as Oracle, Sun Systems, Hewlett Packard and Adobe. All of the Semantic Web languages are platform-independent, license-free standards that are being developed by a well-known standards organization, the W3C. A variety of other Knowledge Representation (KR) languages such as the Knowledge Interchange Format (KIF) are complete and available for use, but they do not benefit from the common format for data exchange provided by XML or from the broad, international support of the W3C.

References

Berners-Lee, T. 2000. XML-2000 Conference Presentation. http://www.cs.umd.edu/~hendler/AgentWeb.html

Brutzman, D. and Grimley, M. 2006. USW-XML: Next-Generation USW Interoperability using XML. Retrieved on April 30, 2006 from http://usw-xml.cs.nps.navy.mil/usw-xml.html

Childers, C. 2006. Applying Semantic Web Concepts To Support Net-Centric Warfare Using The Tactical Assessment Markup Language (TAML), Masters Thesis, Naval Postgraduate School, Monterey, California.

Hayes-Roth, R. 2005. Model-based Communication Networks and VIRT: Filtering Information by Value to Improve Collaborative Decision Making. 10th International Command and Control Research and Technology Symposium.

Hjelm, J. 2001. *Creating the Semantic Web with RDF*. New York: John Wiley & Sons, Inc.

Hunter, D., Cagle, K., Dix, C., Kovack, R., Pinnock, J., & Rafter, J. 2003. *Beginning XML (3rd ed.)*. Indianapolis: Wiley Publishing Inc.

Lacy, L. 2005. *OWL: Representing Information Using the Web Ontology Language*. Canada: Trafford

Pan, J. 2004. Description Logics: Reasoning Support for the Semantic Web. Retrieved on March 1, 2006 from http://dlweb.man.ac.uk/~panz/Zhilin/pubc.php?sort=paper&id=thesis

Powers, Shelley, 2003. *Practical RDF*. Sebastopol, CA: O'reilly & Associates Inc.

Protégé: Retrieved on June 15, 2006 from Protégé Web Site http://protege.stanford.edu

RacerPro: Retrieved on June 15, 2006 from RacerPro Web Site http://www.racer-systems.com

SWRL: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 21 May 2004. Retrieved on June 15, 2006 from http://www.w3.org/Submission/SWRL

SWRL BIS: SWRL Built-in Specification: Retrieved on June 15, 2006 from http://www.daml.org/rules/proposal/builtins.html

Winters, L., and Tolk, A. 2005. The Integration of Modeling and Simulation with Joint Command and Control on the Global Information Grid. http://www.stormingmedia.us/17/1783/A178334.html