

# Application of Ontology Translation

James Ressler<sup>1a</sup>, Mike Dean<sup>2d</sup>, Edward Benson<sup>2e</sup>, Eric Dorner<sup>1b</sup>, Chuck Morris<sup>1c</sup>

<sup>1</sup> Northrop Grumman Corporation  
1010 Market Street, Suite 1740, St. Louis, Missouri, 63101

<sup>2</sup> BBN Technologies

1300 N. 17<sup>th</sup> Street, Suite 400, Arlington, Virginia, 22209

<sup>a</sup>[james.ressler@ngc.com](mailto:james.ressler@ngc.com), <sup>b</sup>[eric.dorner@ngc.com](mailto:eric.dorner@ngc.com), <sup>c</sup>[chuck.morris@ngc.com](mailto:chuck.morris@ngc.com)  
<sup>d</sup>[mdean@bbn.com](mailto:mdean@bbn.com), <sup>e</sup>[ebenson@bbn.com](mailto:ebenson@bbn.com)

**Abstract.** An ontology provides a precise specification of the vocabulary used by a community of interest (COI). Multiple communities of interest may describe the same concept using the same or different terms. When such communities interact, ontology alignment and translation is required. This is typically a time consuming process. This paper describes Snoggle, an open source tool designed to ease development of ontology translation rules, and discusses its application to geospatial ontologies.

**Keywords.** ontology, alignment, translation, SWRL, geospatial, Snoggle.

## 1 Introduction

An ontology is a valuable resource for making sense of data content because it places information in context by using a formal vocabulary to describe the information. Applied to semantic reasoning, a query against instances of an ontology provides the ability to arrive at relevant responses to otherwise complex questions. However, when data from multiple ontologies are queried together, the context can be lost when the ontologies lack uniformity, i.e. sources written with different vocabularies do not lend themselves to machine-intelligible reasoning. The goal of data interoperability can be both advanced and hindered by the use of ontologies. By providing context to data, interoperability is achieved through the relations expressed between data elements. However, if the relations do not rely upon the same vocabulary, connections between data cannot be inferred. Similar to human reasoning, when a person provides information using a different vocabulary, humans mentally build a translation from terms that are known to the new terms that are provided. A mental translation between similar terms allows experts in related fields to communicate. The same type of translation is needed to make ontologies useful across multiple domains in ontologies.

The mapping of two knowledge representations requires precision to clearly define synonymous concepts and the conditions in which those concept equivalencies and other relationships are valid. A capable mapping tool must also be able to express logical, string manipulation and mathematical functions as part of its rules. The tool

requires the ability to filter the rules and quantify the extent of mapping in order to determine coverage. Once a mapping is defined, the rules used to implement the mapping between the two ontologies become part of the knowledge base and can be used to process queries.

Multiple ontologies arise naturally as different communities of interest address overlapping problem domains. Alignment of these ontologies is required to integrate data from their corresponding data sources. Mapping rules are the product of such alignment.

This paper presents research into a variety of geospatial ontologies, the alignment process and presents a graphical mapping tool based on that research that seeks to simplify the alignment process. We address the topics in the following organization:

- Section 2 summarizes a case study and overview performed among multiple geospatial semantic vocabularies. This study highlights the need to better understand and support the ontology alignment process.
- Section 3 discusses the problem of mapping and alignment as applied to domain ontologies and presents prior work in this area.
- Section 4 presents a graphical mapping language and supporting tool called Snoggle<sup>1</sup> that addresses the ontology mapping problem in an intuitive, visual way.
- Section 5 demonstrates an application of this tool to alignment of geospatial vocabularies from the case study.

## 2 Geospatial Semantic Vocabularies

The W3C Geospatial Incubator Group<sup>2</sup> has defined seven initial categories of basic geospatial ontologies: features, feature types, spatial relationships, coordinate reference systems, toponyms (place names), geospatial metadata, and web services. The authors have recommended four additional categories: geometric, coverage, geopolitical, and temporal. A wide range of existing semantic vocabularies cover most of these categories of geospatial ontologies. Selecting an ontology for a geospatial dataset is based upon many factors, including authorship of the data, required expressivity, user preferences, other data sources to be integrated, and the tools to be used. A study of existing geospatial ontologies [14] categorized the ontologies as shown in Figure 1, where “F” indicates the ontology fully applies to the category and “P” indicates the ontology partially applies.

---

<sup>1</sup> <http://snoggle.projects.semwebcentral.org/>

<sup>2</sup> <http://www.w3.org/2005/Incubator/geo/Wiki/FrontPage>

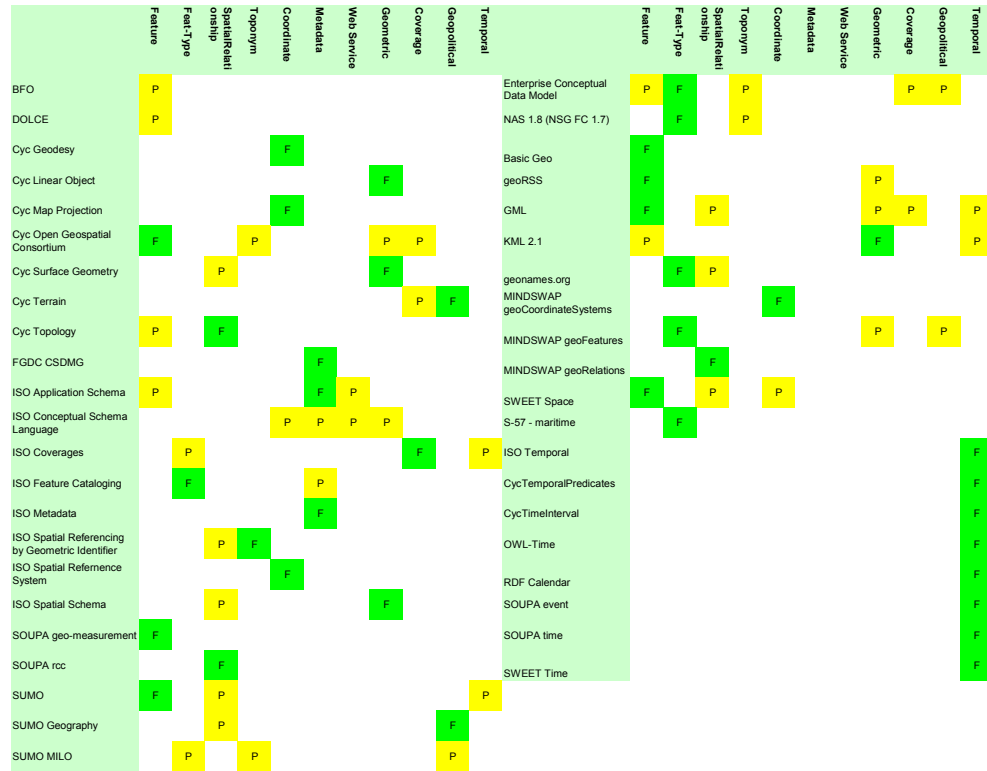


Figure 1: Ontology Coverage of Core Geospatial Categories

As this figure shows, many ontologies cover the majority of the core spatial domains. The Cyc, ISO and NGA based ontologies are most prevalent in the geospatial domain, while other niche models possess unique and useful roles in middle and domain specific ontologies. This diversity demonstrates the need for ontology alignment.

### 2.1 Maritime Case Study

The rest of this paper considers two specific domain ontologies targeted at different communities of interest. The domain ontologies draw classes from several of the ontologies referred to earlier, including those in the Features, Feature Types, Coordinates, and Geometry categories. The Digital Nautical Chart® (DNC®) is produced by the US National Geospatial-Intelligence Agency (NGA) and is a vector-based, digital database containing significant nautical features essential for safe maritime navigation. DNC® is based on the MIL-PRF-89023 specification [11]. The Electronic Navigational Chart (ENC) is produced by the US National Oceanic and Atmospheric Administration (NOAA) and supports the marine transportation infrastructure and coastal management. NOAA ENCs use the International Hydrographic Office (IHO) S-57 exchange format [3]. To illustrate the application of

multiple data sets using similar ontologies, these two maritime product ontologies were mapped. The S-57 ontologies were created by Malyankar during his work on maritime text markup [8] and the DNC® ontology was created for this paper from MIL-PRF-89023. For example, classes exist for bridges, berths, mooring facilities, and pontoons in ENC which are equivalent to bridges, berths, and piers (equivalent to pontoons) in the DNC®. Their associated properties are also structurally equivalent but named differently. By mapping these two ontologies, queries can be made using either with equivalent results. The results of a mapping from ENC to DNC® are shown in Figure 4 on page 9.

### 3 Ontology Mapping and Alignment

Kolas [6] describes a frequently used Semantic Web data integration design pattern. A user domain ontology is defined by subject matter experts and used to express queries within a community of interest. This ontology is mapped to one or more data source ontologies. When it's sufficiently complete, a data source ontology may also be used as the domain ontology. Queries expressed in a domain ontology are translated into queries on the individual sources and the results are translated back into the domain ontology. We applied this design to two maritime data sources as described in Section 5

#### 3.1 SWRL Rules

Ontology mapping can be viewed at a practical level as an applied case of the larger field of rule creation. Mapping between OWL ontologies is typically done using SWRL [13], the Semantic Web Rule Language, which combines OWL and RuleML<sup>3</sup> concepts.

SWRL rules are an implication between an antecedent and a consequent, both comprised of multiple atoms. When the atoms in the antecedent hold, then the conditions in the consequent also hold. Atoms may be of various types that encompass the possible entities and relationships expressible in OWL. Specifically, atoms can take the form  $C(x)$ ,  $P(x,z)$ ,  $\text{sameAs}(x,y)$ , or  $\text{differentFrom}(x,y)$ , where  $x$  and  $y$  are either variables or OWL individuals,  $z$  is a variable, OWL individual, or an OWL data value,  $C$  is an OWL class, and  $P$  is an OWL property.

Mapping from a source ontology to a target ontology using SWRL is thus accomplished by partitioning the source ontology into atomic components, using these components as the antecedents in a set of rules, and placing the equivalent structures in the target ontology as the consequents of such rules. If a source:SalariedEmployee in the source ontology maps to a target:Employee in the target ontology, the following rule would accomplish this mapping:

---

<sup>3</sup> <http://www.ruleml.org/>

source:SalariedEmployee(var) => target:Employee(var)

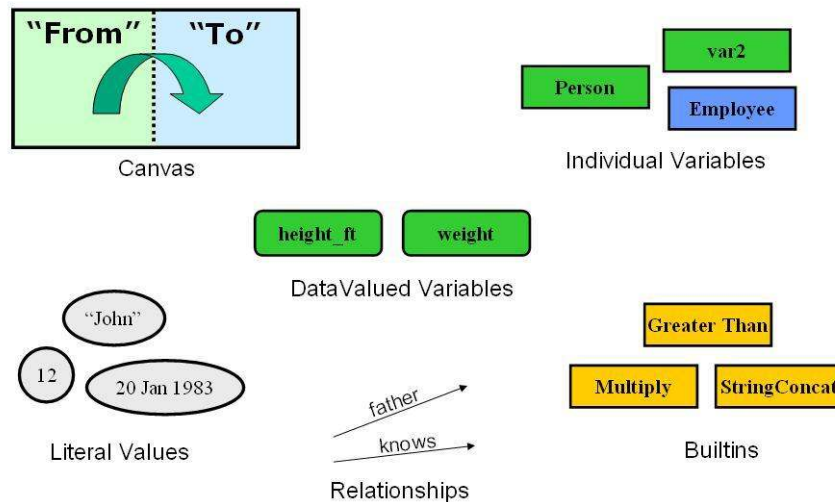
Additional atoms or rules would be used to map the properties associated with each class.

## 4 A Graphical Mapping Tool

Previous literature on the subject of alignment and the maritime case study presented in Section 5 make it clear that mapping rules are critical to the ontology alignment process, but few tools exist to assist in the creation of such rules. Users are faced with the choice of general-purpose rule editors or of lengthy hand-coded XML files. To ease the translation process, we designed and developed Snoggle, a graphical ontology mapping environment.

### *Visual Vocabulary*

Central to Snoggle is the idea of a visual language created specifically for mapping. This language is designed for the general problem of mapping OWL-based data and not toward rule creation or any one particular rule language. Snoggle currently exports its results as SWRL, but can support exporters for other rule languages. Snoggle's visual language is depicted in Figure 2.



**Figure 2: Visual Mapping Language Elements**

Snoggle defines a mapping rule as consisting of two graphs: the source graph and the target graph. Using the visual language depicted in Figure 2, users draw structures using the source ontology in the source graph and a corresponding structure using the

target ontology in the target graph. The user then “maps” the source graph to the target graph by drawing arrows between corresponding nodes. To accommodate unit conversions and other calculations, the source graph can also include nodes representing standard or user-defined SWRL builtin functions.

#### *Representing Variables, Literals, and Comments*

Variables in Snoggle are represented by rectangles (referred to as nodes) on the canvas. When creating a variable, the user must explicitly choose either a SWRL data variable or a SWRL individual variable. Data variables are displayed with rounded corners and individual variables are shown with hard corners, and the background shading on a variable node is dependent on the ontology (source or target) from which it came. Literal values are displayed within ovals and comments are displayed in pastel-colored rectangles.

#### *Mapping Rules*

Within the canvas, every node in the source (antecedent) region that corresponds to a node in the target (consequent) region must be connected via a map. A map is simply a boldface arrow which is drawn between a node in the source region and a corresponding node in the target region.

As a user builds rules, graphical depictions of the source and target ontologies incrementally update themselves to show which aspects of the two ontologies remain unmapped. This provides a quick means of assessing coverage and tracking workflow. At any time, Snoggle allows the user to export the current mapping as a set of SWRL rules, or rules of some other language supported by an exporter.

## **4.1 Snoggle Design**

Snoggle is conceptually divided into four principal user-facing components: the workspace, the canvas, the ontology browser, and the editors. Each of these presents a particular view of the rule development process and together they coordinate all of Snoggle’s user-directed functionality. In addition to these components, a SWRL Model and Canvas Model play important roles in transforming graphs on the screen into mapping rules.

#### *Workspace*

The Snoggle workspace is the container for all other Snoggle operations. It handles the loading and unloading of Snoggle project files, which contain all the information about a particular mapping task. Mapping from one ontology to another generally requires a large set of rules, depending on the size of the concerned ontologies, so projects are intended to contain multiple rules at the same time.

Each project file contains the following information:

- References to the source and target ontologies

- A set of prefixes for defined namespaces
- Any custom builtins loaded into the project
- The mapping rules created for the project

Each component of the Snoggle workspace shown in Figure 3 is described separately.

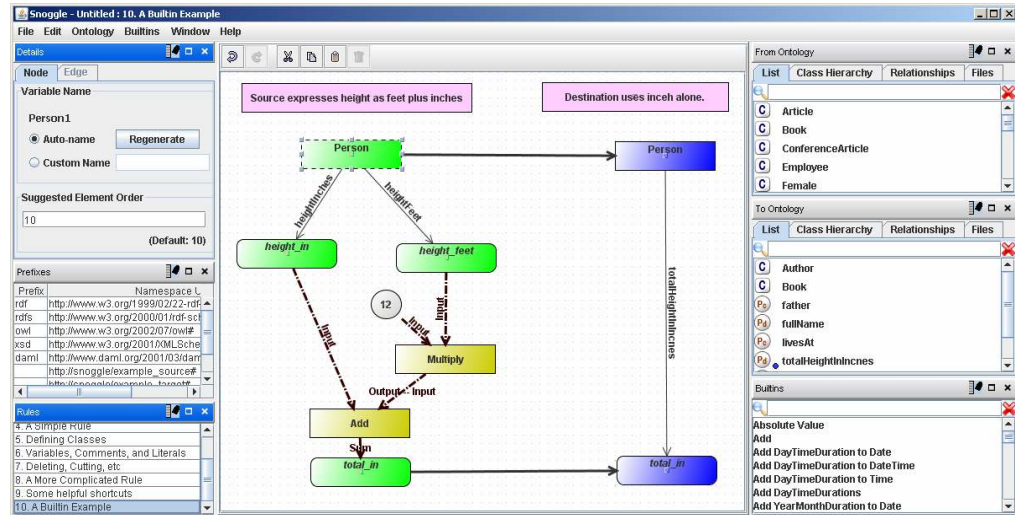


Figure 3: Snoggle Workspace

### Ontology Browser

The Ontology Browser is a container for viewing any ontology loaded into a project. It provides three ways to visualize these ontologies: a flat list of classes and properties, a tree hierarchy of classes, and a bi-directional tree depicting the relationships between properties and classes. The most common mode of ontology mapping uses two separate ontology browsers: a “From” browser and a “To” browser. As objects in the “From” and “To” ontology are used to create rules for a project, users have the option of having them automatically hidden from the Ontology Browser’s view, thus providing a running list of everything that remains to be mapped in a particular Snoggle project.

### Canvas

Snoggle’s Canvas provides a graphical space on which to draw mapping rules using the objects loaded into the Ontology Browsers. The Canvas is divided into two special areas – the left side contains all entities and relationships that form the antecedent of the mapping rule, and the right side contains those that appear in the consequent. The left side represents the source ontology’s structure and the right side contains the equivalent structure in the target ontology. Nodes in the antecedent are displayed in green, and nodes in the consequent are displayed in blue.

Using the visual language, users drag objects from the Ontology Browser onto the Canvas and connect them with various types of arcs. Class objects dragged from the Ontology Browser become typed variable nodes; property objects can be dragged onto edges between nodes. Builtin functions are represented with a special color node, with incoming and outgoing edges mapped to the arguments of the builtin.

#### *Builtins Viewer*

The Builtins Viewer contains a list of builtin functions that can be used in the project. All of the builtin functions defined within the SWRL specification are included. Custom Builtins can also be added by loading a file containing function definitions. Builtins are represented by red rectangles with hard corners.

#### *Rule Editor*

The Rule Editor keeps track of the various rules within a project, and manages the loading and unloading of those rules onto the Canvas.

#### *Namespace Editor*

The Namespace Editor manages all namespaces loaded into the project and handles the automatic prefixing of displayed information.

#### *Object Editor*

The Object Editor provides information and settings for the object on the Canvas that currently has focus. The available settings depend on the particular type of object selected, such as variable nodes to contain class and variable information. Arcs denote input to a builtin that allows the user to specify which of the builtin's inputs are represented by the source of the arc.

An illustration of the Canvas defining a rule between two definitions of the class Person is shown in Figure 3. Note the use of the Add and Multiply builtin functions to convert different representations of height.

## **5 Application to a Maritime Query**

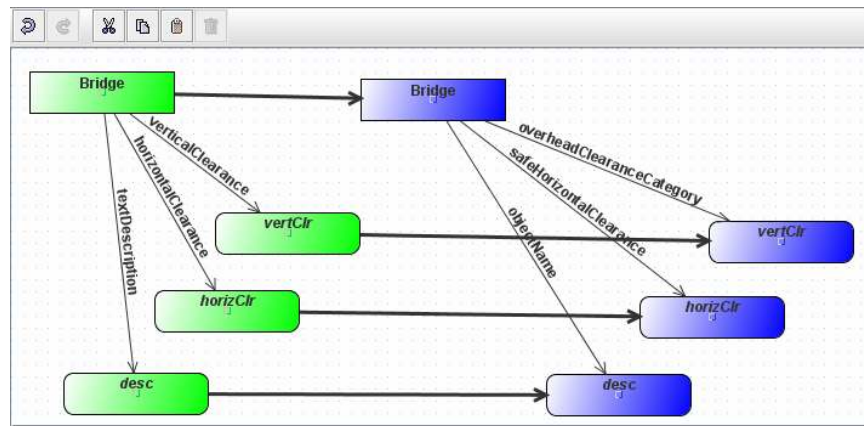
We used Snoggle to integrate the heterogeneous maritime ontologies introduced in Section 2 to answer a maritime query that neither data source could answer alone: “Which bridges in Boston Harbor can ship  $x$  navigate through?” In other words, given ship  $x$  with an aircraft of  $y$ , find all bridges where the height of the bridge  $h > y$ . Digital Nautical Chart® (DNC®) and Electronic Nautical Chart (ENC) are the maritime infrastructure data sources used. A data source containing information associated with ship  $x$  was also used.

DNC® and ENC contain similar maritime infrastructure objects such as bridges, piers, and buoys, and an ontology was generated for each data source based on their respective specifications. However, since both data sources are based on different specifications, their associated properties are also equivalent but named differently.



For instance, both ontologies have a Bridge class and corresponding properties associated with the class. However, the property corresponding to the distance from the bridge span to the surface of the water is different in the two ontologies. The DNC® property is named `overheadClearanceCategory` while the ENC property is named `verticalClearance`. In addition to semantic differences, the instance data itself varies between the two data sources. For example, the DNC® data source contains bridges that are not present in the ENC data source and vice versa.

The mapping between corresponding Bridge classes was constructed using Snoggle as illustrated in Figure 4.



**Figure 4: Mapping of Bridge classes from ENC to DNC®**

The resulting SWRL rule created by Snoggle for the bridge mapping is as follows (using a more accessible presentation syntax – the actual SWRL/XML syntax appears in the Appendix) :

```

enc:Bridge(?Bridge1) &
enc:textDescription(?Bridge1, ?desc) &
enc:horizontalClearance(?Bridge1, ?horizClr) &
enc:verticalClearance(?Bridge1, ?vertClr)
=>
dnc:Bridge(?Bridge1) &
dnc:objectName(?Bridge1, ?desc) &
dnc:safeHorizontalClearance(?Bridge1, ?horizClr) &
dnc:overheadClearanceCategory(?Bridge1, ?vertClr)

```

This SWRL rule maps the `verticalClearance` attribute of the ENC ontology to the `overheadClearanceCategory` property of the DNC® ontology. The mapping will integrate the two datasets and allow the merged result to be queried using the `overheadClearanceCategory` property.

The SWRL mapping rules were translated into Jena rules using the SweetRules<sup>4</sup> package within a maritime application interface. The application then executed the Jena rules using the Jena rule engine. This resulted in the addition of overheadClearanceCategory statements to the ENC RDF instance data. The SPARQL query shown below was executed and returned all DNC® and ENC bridge instances that ship x (in this case, a Tennessee class battleship) could navigate under. The returned geometries were then converted to KML and the results were displayed in Google Earth.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
ns#>
PREFIX dnc:
<http://ww4.geoenterpriselab.com/ont/dnc/dncsubset.owl#
>
PREFIX class:
<http://ww4.geoenterpriselab.com/ont/watercraft/watercr
aft_1.0.owl?object=Class#>

SELECT ?bridge ?ohc ?geom
WHERE {
    ?bridge rdf:type dnc:Bridge .
    ?bridge dnc:overheadClearanceCategory ?ohc .
    ?bridge dnc:coordinates ?geom .
    ?bridge dnc:objectName ?nam .
    <http://ww4.geoenterpriselab.com/rdf/watercraft.rdf?obj
ect=Class/Military/Battleship/Tennessee#Object>
class:airDraft ?airdraft .
    FILTER (?airdraft < ?ohc)
}
```

In conclusion, the mapping between the DNC® and ENC ontologies integrates both a military and a commercial maritime data source. Since the data contained within these sources is not identical, the mapping allows both of the sources to be used to paint a more complete picture of the maritime infrastructure.

## 6 Related Work

We use the term ontology translation to refer to a three phase process: 1) identifying related ontologies, 2) aligning related classes within them, and 3) creating executable mappings between those classes. Various subsets of this process have been addressed by other researchers in the ontology and database communities under a variety of names including ontology/schema alignment, matching, merging, mapping, and articulation. Kalfoglou [5] and Noy [12] surveyed such work; links to more recent

<sup>4</sup> <http://sweetrules.projects.semwebcentral.org/>

work are available at [ontologymatching.org](http://ontologymatching.org). Kotis [7] defines the ontology merging problem to be:

Given two ontologies find an alignment between these two ontologies, and then, get the minimal union of their translated vocabularies and axioms with respect to their alignment.

The alignment and merging of ontologies has been addressed by tools such as the INRIA alignment tool [1], the HCONE approach to ontology merging [7] and PROMPT [12], a merging plug-in to Protégé. Many alignment tools place some measure of confidence or rank order on the suggested mappings between concepts. All of these tools can align with some degree of accuracy, and are usually between 30% and 40% effective at precise matching [2]. It can be assumed that human reasoning will be required to clarify the relations between concepts. Snoggle is intended to capture human-defined mappings in a graphical representation.

While previous work has focused on the first two phases, Snoggle currently covers the third phase of the ontology translation process to create executable mappings. We would like to extend it to address earlier phases, through the use of INRIA's Alignment API<sup>5</sup> [1] which is already supported by many matching tools.

Snoggle differs from other SWRL rule editors such as Protégé [13], the Object Rule Editor (ORE) [10], and RuleVISor [9] in supporting a graph-based visual paradigm and in focusing on ontology translation through the use of multiple ontology browsers, a divided Canvas, and coverage indicators.

## 7 Conclusions

This paper has motivated the need for ontology alignment and shown the utility of creating mapping rules between data source and domain ontologies. We highlighted this need through a case study involving multiple overlapping but incomplete maritime ontologies and data sets.

Snoggle is a new tool that provides an easy method to visually define SWRL mappings between OWL ontologies using a simple workflow. It defines an intuitive visual language to describe the mapping metaphor and permits use of that language through a drag-and-drop interface using visualizations of source and target ontology and the mappings between them.

We demonstrated Snoggle in practical use by aligning elements of the maritime ontologies from the case study so that a query could be performed across them that would not be possible with either of the ontologies alone. This straightforward mapping could easily be expanded to more complicated operations using Snoggle's

---

<sup>5</sup> INRIA's alignment tool is also capable of exporting OWL and SWRL.

builtin support, encompassing operations ranging from unit conversions to spatiotemporal processing.

#### **Acknowledgments.**

The authors express their appreciation to the National Geospatial-Intelligence Agency (NGA) for support in developing Snoggle and preparing this paper.

#### **References**

1. Euzenat, Jerome, "An API for ontology alignment (version 2.1)," INRIA Rhone-Alpes, 2006.
2. Euzenat, Jerome, et. al. "Results of the Ontology Alignment Evaluation Initiative 2006," Ontology Alignment Evaluation Initiative (OAIE), Proc. 1<sup>st</sup> International Workshop on Ontology Matching at the 5<sup>th</sup> International Semantic Web Conference, Athens, GA, Nov. 2006.
3. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., Dean, M. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," W3C Member Submission, 21 May 2004.
4. International Hydrographic Organization, "IHO Transfer Standard for Digital Hydrographic Data," S-57 Edition 3.1, Nov. 2000.
5. Kalfoglou, Y., Schorlemmer, M., "Ontology mapping: the state of the art", The Knowledge Engineering Review 18(1), 2003.
6. Kolas, Dave, Dean, Mike and Hebel, John, "Geospatial Semantic Web: Architecture of Ontologies," Proc. IEEE Aerospace Conference, Big Sky, MT, March 2006.
7. Kotis, Konstantinos, and Vouros, George. "The HCONE Approach to Ontology Merging," Proc. First European Semantic Web Symposium, Heraklion, Greece, May, 2004.
8. Malyankar, Raphael. "Elements of Semantic Web Infrastructure for Maritime Information," National Technical Meeting for the Institute of Navigation, San Diego, CA, 2002.
9. Matheus, Christopher. "SAWA: An Assistant for Higher-Level Fusion and Situational Awareness," Proc. SPIE Conference on Multisensor, Multisource Information Fusion, Orlando, FL, March 2005.
10. Munoz, A., Vera, A., Botia J., and Gomez Skarmeta, A. "Defining Basic Behaviors in Ambient Intelligence Environments by Means of Rule-based Programming with Visual Tools". Proc. 1st Workshop on Artificial Intelligence Techniques for Ambient Intelligence. ECAI 2006, Trento, Italy.
11. National Imagery and Mapping Agency (NIMA), "Performance Specification Digital Nautical Chart," MIL-PRF-89023, 19 Dec. 1997.
12. Noy, Natalya, and Musen, Mark. "Evaluating Ontology-Mapping Tools: Requirements and Experience," Proc. OntoWeb-SIG3 Workshop at the 13<sup>th</sup> International Conference on Knowledge Engineering and Knowledge Management, Siguenza, Spain, 2002.
13. O'Connor, M., Knublauch, H., Tu, S., Grosz, B., Dean, M., Grosso, W., Musen, M., "Supporting Rule Systems Interoperability on the Semantic Web with SWRL", Proc. 4<sup>th</sup> International Semantic Web Conference, Galway, Ireland, November 2005.
14. Ressler, J., Dean, M. and Kolas, D., "Geospatial Ontology Trade Study", [http://projects.semwebcentral.org/docman/?group\\_id=84](http://projects.semwebcentral.org/docman/?group_id=84), June 2007.

**Appendix: ENC to DNC Mapping Rule in SWRL XML Format**

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<swrlx:Ontology
xmlns:swrlx="http://www.w3.org/2003/11/swrlx"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns:j.0="http://protege.stanford.edu/plugins/owl/protege#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:owlx="http://www.w3.org/2003/05/owl-xml"
xmlns:pl="http://www.owl-ontologies.com/assert.owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:ruleml="http://www.w3.org/2003/11/ruleml"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ruleml:var>vertClr</ruleml:var>
  <ruleml:var>Bridge1</ruleml:var>
  <ruleml:var>Bridge1</ruleml:var>
  <ruleml:var>horizClr</ruleml:var>
  <ruleml:var>desc</ruleml:var>
  <ruleml:var>Bridge1</ruleml:var>
  <ruleml:var>Bridge1</ruleml:var>
  <ruleml:imp>
    <ruleml:_rlabel ruleml:href="#BridgeMap"/>
    <ruleml:_body>
      <swrlx:classAtom>
        <owlx:Class
owlx:name="http://ww4.geoenterpriselab.com/ont/s57/S57V31C016.owl#Bridge"/>
          <ruleml:var>Bridge1</ruleml:var>
        </swrlx:classAtom>
        <swrlx:datavaluedPropertyAtom
swrlx:property="http://ww4.geoenterpriselab.com/ont/s57/S57V31C016.owl#horizontalClearance">
          <ruleml:var>Bridge1</ruleml:var>
          <ruleml:var>horizClr</ruleml:var>
        </swrlx:datavaluedPropertyAtom>
        <swrlx:datavaluedPropertyAtom
swrlx:property="http://ww4.geoenterpriselab.com/ont/s57/S57V31C016.owl#verticalClearance">
          <ruleml:var>Bridge1</ruleml:var>
          <ruleml:var>vertClr</ruleml:var>
        </swrlx:datavaluedPropertyAtom>
        <swrlx:datavaluedPropertyAtom
swrlx:property="http://ww4.geoenterpriselab.com/ont/s57/S57V31C016.owl#textDescription">

```

```

        <ruleml:var>Bridge1</ruleml:var>
        <ruleml:var>desc</ruleml:var>
    </swrlx:datavaluedPropertyAtom>
</ruleml:_body>
<ruleml:_head>
    <swrlx:classAtom>
        <owlx:Class
owlx:name="http://ww4.geoenterpriselab.com/ont/dnc/dncsubset.owl#Bridge"/>
        <ruleml:var>Bridge1</ruleml:var>
    </swrlx:classAtom>
    <swrlx:datavaluedPropertyAtom
swrlx:property="http://ww4.geoenterpriselab.com/ont/dnc/dncsubset.owl#safeHorizontalClearance">
        <ruleml:var>Bridge1</ruleml:var>
        <ruleml:var>horizClr</ruleml:var>
    </swrlx:datavaluedPropertyAtom>
    <swrlx:datavaluedPropertyAtom
swrlx:property="http://ww4.geoenterpriselab.com/ont/dnc/dncsubset.owl#overheadClearanceCategory">
        <ruleml:var>Bridge1</ruleml:var>
        <ruleml:var>vertClr</ruleml:var>
    </swrlx:datavaluedPropertyAtom>
    <swrlx:datavaluedPropertyAtom
swrlx:property="http://ww4.geoenterpriselab.com/ont/dnc/dncsubset.owl#objectName">
        <ruleml:var>Bridge1</ruleml:var>
        <ruleml:var>desc</ruleml:var>
    </swrlx:datavaluedPropertyAtom>
</ruleml:_head>
</ruleml:imp>
</swrlx:Ontology>

```