A Practical Approach to Data Modeling using CCO

Rod Moten
Datanova Scientific
Baltimore, Maryland

Bill Barnhill
EOIR Technologies
APG, MD

***Abstract*—In this paper, we present work in progress on using the Information Domain ontologies of CCO (Common Core Ontologies) as a domain model for land combat. Our goal is to use the domain model as a common semantics for multiple land combat logical models. In the paper, we show how our domain model can be mapped to different logical models in a manner that is less labor intensive than the approach commonly used by users of CCO. We demonstrate our approach by describing how our domain model, which is a domain ontology of CCO, is mapped to logical models created in Ecore and NIEM (National Information Exchange Model).**

I. Introduction

There are three primary forms of a data model, domain model, logical model, and a physical model [1]. A domain model specifies the concepts that data represents, the properties of the concepts and the relationships between concepts. A logical model species the logical structure of data. A physical model species how data is represented in machine readable format. Ideally, a logical model is derived directly from a domain model or a formal relationship is defined between the domain model and the logical model. In these cases, the domain model serves as the semantics of the logical model. Semantics is assigned to the logical model via a mapping between the domain model and the logical model.

There are multiple approaches of performing this mapping. One approach is to develop a mapping between objects in the domain model and and objects in the logical model. For example, the domain model could be defined using an ontology. The mapping specifies how to convert objects in the logical models to individuals in the ontology.

We used this approach for several projects where the domain models were domain ontologies of CCO (Common Core Ontologies) [2]. CCO is a collection of upper, middle, and domain ontologies in OWL that extend BFO (Basic Formal Ontologies) [3]. Figure 1 contains a diagram of the ontologies in CCO.

One of the authors of this paper has used CCO for creating domain ontologies for a motion imagery analysis application [4] and other projects. In all of these projects, we sought to use ontologies conformant to the CCO as domain models. In addition, we sought to create mappings from the logic models of existing tactical military software systems to the domain models. We required the assistance of an ontologist with in-depth knowledge of CCO to create the mappings. As a result, using CCO may have a higher cost than an approach that allows programmers or data architects to develop the mapping independently. As a result, the government sponsor of the projects considered the use of CCO impractical for tactical military systems.

We believe that CCO is practical for tactical military systems. The problems we encountered were due to how CCO was used. The problems we encountered occurred because of differences in the modeling objectives of a logical model and a domain model defined as a formal ontology. A logical model defines the symbolic structure of entities for automated processing and analysis. The structure is chosen in order to simplify processing and analysis. For example, the essential properties of a person, such as name and birth date, are modeled as attributes of the same object in a logical model. However, the domain ontologies of CCO are specifications of the metaphysical make up of entities. Therefore, essential properties of the same entity may have different structural representations as individuals in the CCO. In other words, the graph patterns of the triples representing the essential attributes of the same entity may be different. For example, a birth date for a person is a temporal interval for a birth event that occurs on a person agent. A name of a person is an information bearer that inheres on a person agent. This means to map a person entity in a logical model requires determining how each attribute is represented metaphysically and then create the triples accordingly.

An approach that requires examining each attribute equates to defining a separate function for converting each attribute to individuals in the domain model. If we measure the cost of creating a mapping based on the number of functions that have to be created, then an approach that used a single function for mapping sets of entities to concepts may be less expensive than an approach that required a function for each attribute.

To develop an approach based on converting sets of entities to concepts, we propose modeling a domain model as information about the metaphysical properties of entities. In other words, consider the domain to be the terms that designate the entities and relationships between the entities. For example, Aircraft and F-14 would be concepts where F-14 is subsumed by Aircraft. In this case, there are multiple Aircraft individuals and multiple F-14 individuals which are also Aircraft individuals. However, in an information model, there is only one designator term for all aircrafts and one designator term for all F-14s. The subsumption relationship between Aircraft and F-14 could be modeled using a descriptive term, such as derives-from. More specifically, the relationship could be modeled as the triple ‘F-14 derives-from Aircraft’. This means the domain ontology has to extend the Information Domain ontologies of

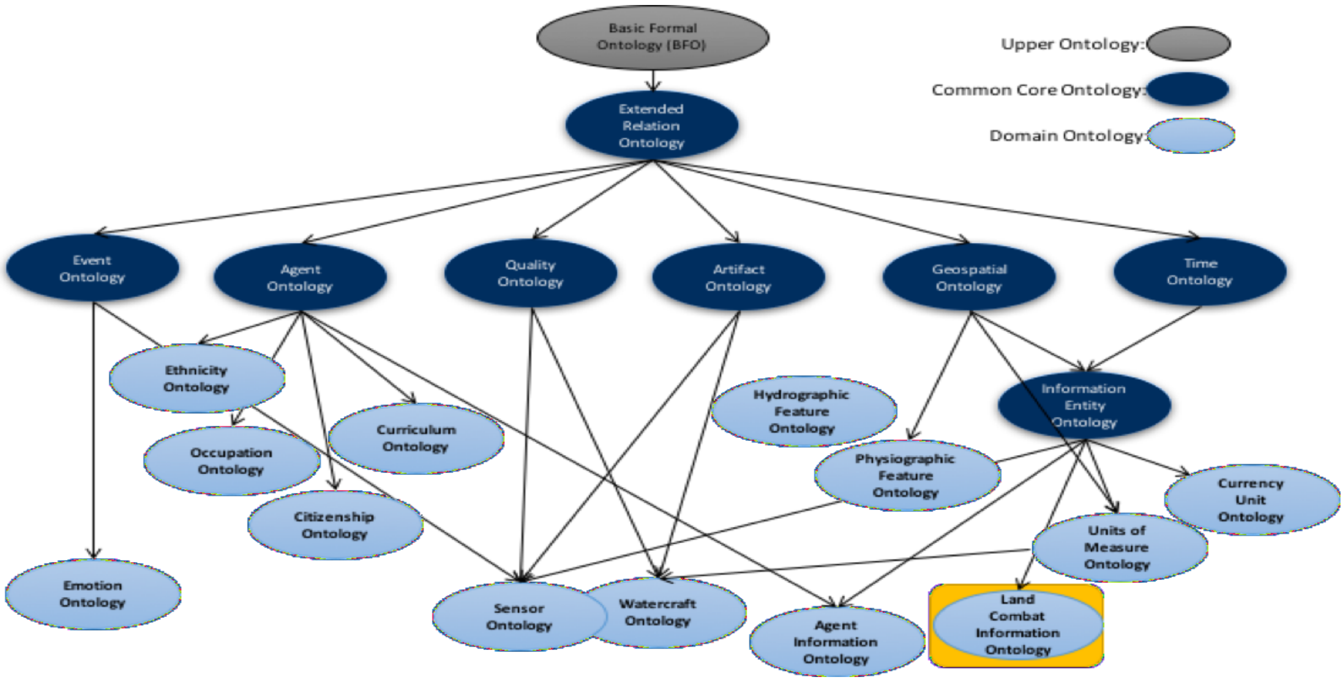

Fig. 1. The ontologies of CCO and the Land Combat Information Ontology.

CCO. However, we have to ensure that the domain ontology isn't just an OWL encoding of a logical model. This approach is used by some techniques for automatically creating schema from ontologies [5].

Using this approach, we do not map objects in the logical model to individuals in the ontology. Instead, we create a mapping where the domain model represents concepts that have direct mapping to syntactic classes in the logical model. This mapping should be more intuitive to data architects since it requires little knowledge of CCO and ontology development.

In this paper, we demonstrate a method for creating domain ontologies in CCO that can be systematically mapped to logical models. In Section II, we provide an overview of the Information Domain ontologies of CCO. Then in Section III we describe how a domain ontology should extend the Information Domain ontologies by creating a proof–of–concept domain ontology for land combat. Then in Section IV we describe how the domain ontology maps to logical models in ECore [6] and NIEM (National Information Exchange Model)[1]. We conclude the paper in Section V with a discussion on why we think our approach faithfully encodes the semantics of the domain and isn't merely a logical model in OWL.

II. Information Ontologies in CCO

The information entity ontology is partitioned into two class hierarchies, *information bearing entities* and i*nformation content entities*. We call information bearing entities *information bearers* for short.

An information bearers is and independent continuant that carries information. For example, a track of an aircraft is an information bearer because it contains information about the flight pattern of an aircraft.

Information content entities are things used to represent information for an information bearer. For example, a 2D graph could be the information content entity of an air track. In this case, the 2D graph is the information that represents the flight pattern of an aircraft. In addition, a 3D graph could be the information content of the air track. The information content entity does not have to be unique to its bearer. For example, -20 degrees Celsius is an information content entity that inheres in many information bearers, such as the current temperature or the lowest operating temperature.

Information content entities are organized into three hierarchies, *directive information*, *designative information*, and *descriptive information*. In this paper, we only use designative and descriptive information entities. Therefore, we omit describing directive information. Designative content entities consist of a set of symbols that denote some entity. Type codes are an example of designative content entities. Descriptive content entities consist of a set of propositions that describe some entity. Numeric scales are examples of descriptive content entities.

There is only one class for Information Bearers, Information Entity Bearers. Our domain ontology for land combat will define a hierarchy for land combat terms with Information Entity Bearer as the root.

III. Land Combat Domain Model

In this section, we give an overview how we created the land combat domain model as an extension of the Information Entity Ontology.

[1]https://www.niem.gov/

Descriptive Name	Acronym/Standard Name
Common Warfighting Symbology	MIL-STD-2525C
Variable Message Format	MIL-STD-6017C
US Message Text Format	MIL-STD-6040 Rev. B
Modernized Intelligence Database	MIDB
Ground-Warfighter Geospatial Data Model	GGDM

TABLE I
LAND COMBAT DOMAIN SOURCES

A. Identify Sources

The first step in creating the domain model is identifying the sources of the information entities. For the land combat proof–of–concept, we use the standards in Table I.

B. Define Class Hierarchy

For the second step, we defined a class hierarchies that extend Information Bearing Entity and Information Content Entity.

Our approach is based on the assumption that the domain model is a conceptualization of information about entities. More specifically, the domain model consists of concepts that can be classified as an *entity report*, an *entity artifact*, or an *entity representation*. An entity report is a concept which captures in a structured machine-readable form one or more observations about an entity's state at a given time, as observed by an agent with a given location (where the agent can be human or software). An entity artifact is a concept which describes assertions about an entity. Entity artifacts are derived either from entity records or from other entity artifacts. For example, a detailed entity artifact about a person can be created from multiple entity records obtained from HUMINT sources. There can be more than one entity artifact asserting information about a given entity or there may be no entity artifacts asserting information about a particular entity. An entity representation is a concept describing human understandable signs and symbols which can be presented to a human actor via some sensory medium (e.g., an audible alert, a PowerPoint deck, a printed document). Figure 2 shows an example of the entity informational categories.

We partition the terms into two groups. We define OWL classes for each of these groups. The first group of terms are terms representing entity artifacts and entity reports. We call these terms *LC (Land Combat) Information Entities*. The second group of terms contain qualities, traits, roles, and characteristics of the entity referenced by an entity artifact or an entity report. The class for this group of terms will be Information Content Entity classes. Figure 3 shows a snapshot of the object properties, LC Information Bearing Entities, and the Information Content Entity classes.

C. Convert Terms to Individuals

In this step, we present the guidelines we used to determine the terms from the source documents we used as individuals in the ontology. We use the noun and adjective phrases in the source documents to create the individuals in the ontology. For example, the terms 'aircraft carrier', 'light', 'guided missile',

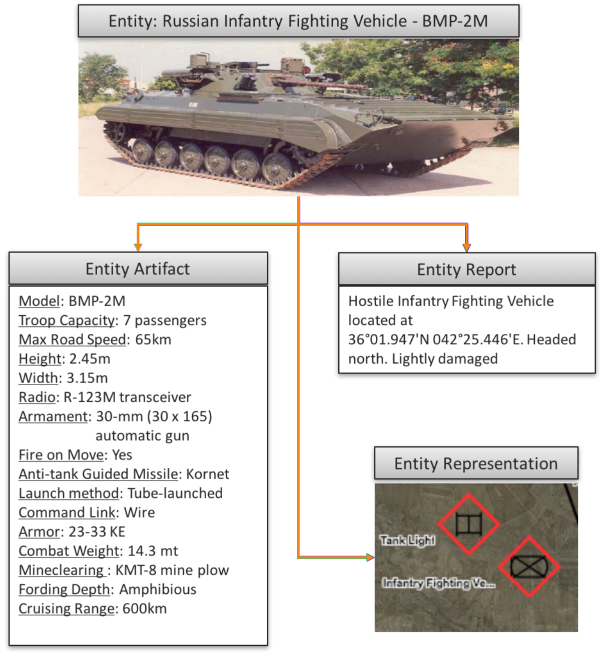

Fig. 2. Example depicting informational entity categories.

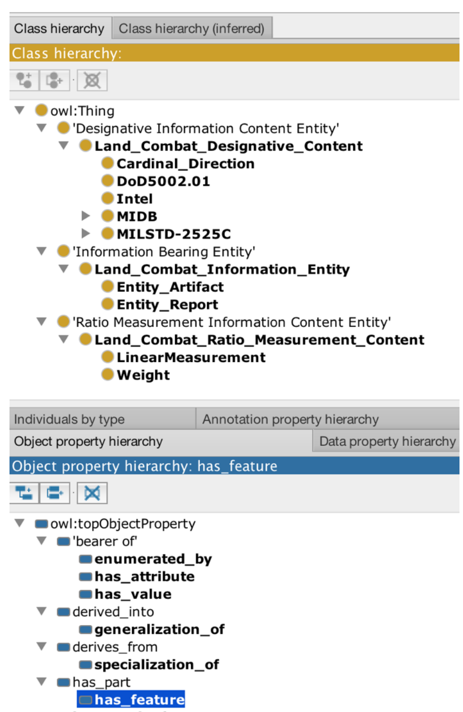

Fig. 3. Screen shot of the Land Combat Domain Model T-Box in Protégé.

and ‘nuclear powered’ are noun and adjective phrases in USMTF. Each of these terms will be an individual. The adjective phrases will become land combat designative content individuals. The noun phrases will become information content entity individuals and information bearing entity individuals.

The usage of the noun phrase determines which class the term belongs to. If the noun phrase is an entity, such as aircraft carrier, then it will become an LC Information Entity. If the noun phrase is the value of a type code, then it will become an Information Content Entity. More specifically, it will be an individual of a subclass of LC Designative Content. If it is a multi-valued numeric attribute, then it will be an individual of an LC Ratio Measurement Content subclass.

The individuals of the LC Relation class are verb phrases that describe a relationship between terms in the standard. For example, 2525C contains a taxonomy of air tracks about different kinds of aircraft. Therefore, ‘is about’ is a relation between the LC Information Entities. Notice that the relation individuals may not be verb phrases in the standard. Instead, they are conceptualization of the relationships between terms in the standard.

D. Define Ontological Relationships of the Domain

By defining relationships between terms using an individual, we can support defining an arbitrary number of relations. We can use OWL properties as *meta–relationships* between individuals. More specifically, we define a fixed set of OWL properties for defining subsumption and composition relationships between individuals. These relationships hold for all domains.

Each of the meta–relation properties is a CCO property or a sub-property of a CCO property. Figure 4 depicts pictorially a sample of triples using all of the meta–relation properties. The CCO properties are in blue and the derived properties are in black. The ‘derives from’ indicates the subject has all of the same properties as the object. Therefore, ‘Stragetic Bomber’ and ‘Tactical Bomber’ each have a ‘Fixed Wing’ as a quality. The ‘derives from’ property is the only subsumption property in our model. The properties ‘has object’ and ‘has subject’ are used to indicate the subject and object of an LC relation. The properties ‘has feature’, ‘has part’, ‘has value’, and ‘has code’ all indicate a part–whole relationship between the subject and object. The difference between the three is the range of the properties. The range of ‘has feature’ is Information Content Entities, but the range of ‘has part’ is an LC Info Entity class. The range of ‘has code’ is LC Info Type Code. And the range of ‘has value’ is subclass of LC Ration Measurement Info Term. The property ‘enumerated by’ indicates the enumerations of a type code. The property ‘has quality’ indicates the object is a quality of the subject.

IV. LAND COMBAT LOGICAL MODELS

In this section, we describe how classes and individuals from the domain model created in Section III map to logical models in ECore and NIEM.

A. Mapping to ECore

ECore is a metal model for defining models in EMF (Eclipse Modeling Framework) [6]. Using Ecore, developers can create models similar to UML Class diagrams and automatically generate code from the models. Ecore contains constructs and features common in object-oriented design, such as classes, enumerations, and inheritance.

Mapping to an object model in ECore is straightforward. Each of the individuals of Type Code becomes an Enumeration class in ECore. The enumerations are determined by the ‘enumerated-by’ property. More specifically, if A ‘enumerated by’ X and A ‘enumerated by’ Y are triples, then X and Y are the enumeration literals of enumeration class corresponding to A.

Each LC Info Entity individual will be a class in ECore that extends the root class `InfoEntity`. The derived from property determines its subclasses and parent class. More specifically, if A ‘specialization of’ B or B ‘generalization of’ A is a triple, then the ECore class corresponding to A, will be a subclass of the ECore class corresponding to B. The attributes of the classes will be defined as follows. For each triple $S\ p\ O$, where S is a LC Entity Info Individual and p is one of the properties, ‘has feature’, ‘has value’, ‘has attribute’, or ‘has part’, there will be an attribute in the class corresponding to O whose type is the type corresponding to O. Each of these types will be created as classes using the same approach.

If the ECore class created from the Entity individual A does not have any attributes, then it can be made into an enumerated class. This will require the individual B in a triple A ‘specialization of’ B or B ‘generalization of’ A be converted into an enumeration literal.

Each A ‘is record of’ B triple will be converted into an association class. More specifically, it will be converted into a class that contains two attributes, `subject` and `object`. The type of `subject` will be the type corresponding to A. The type of `object` will be the type corresponding to B.

B. Mapping to NIEM

NIEM is a logical model developed by the U.S. Government to enable state and federal agencies to share data. The purpose of NIEM is to establish a common structured vocabulary for a set of terms used in all domains relevant to government activities, such as person and location, and a set of common terms used in specialized domains relevant to some government activities, such as hospital and unmanned vehicle. NIEM uses XSD and UML to define the terms so that it can be readily used in software.

In NIEM, terms are partitioned into *elements* and *types*. An element represent properties or attributes of objects. A type represents a set of objects that have the same properties and semantics.

Each Entity individual will be a NIEM type. Elements of the NIEM types are determined by the objects in triples. Objects of ‘has feature’, ‘has attribute’, and ‘has part’ will be come composite elements. Objects of ‘has value’ will be come scalar

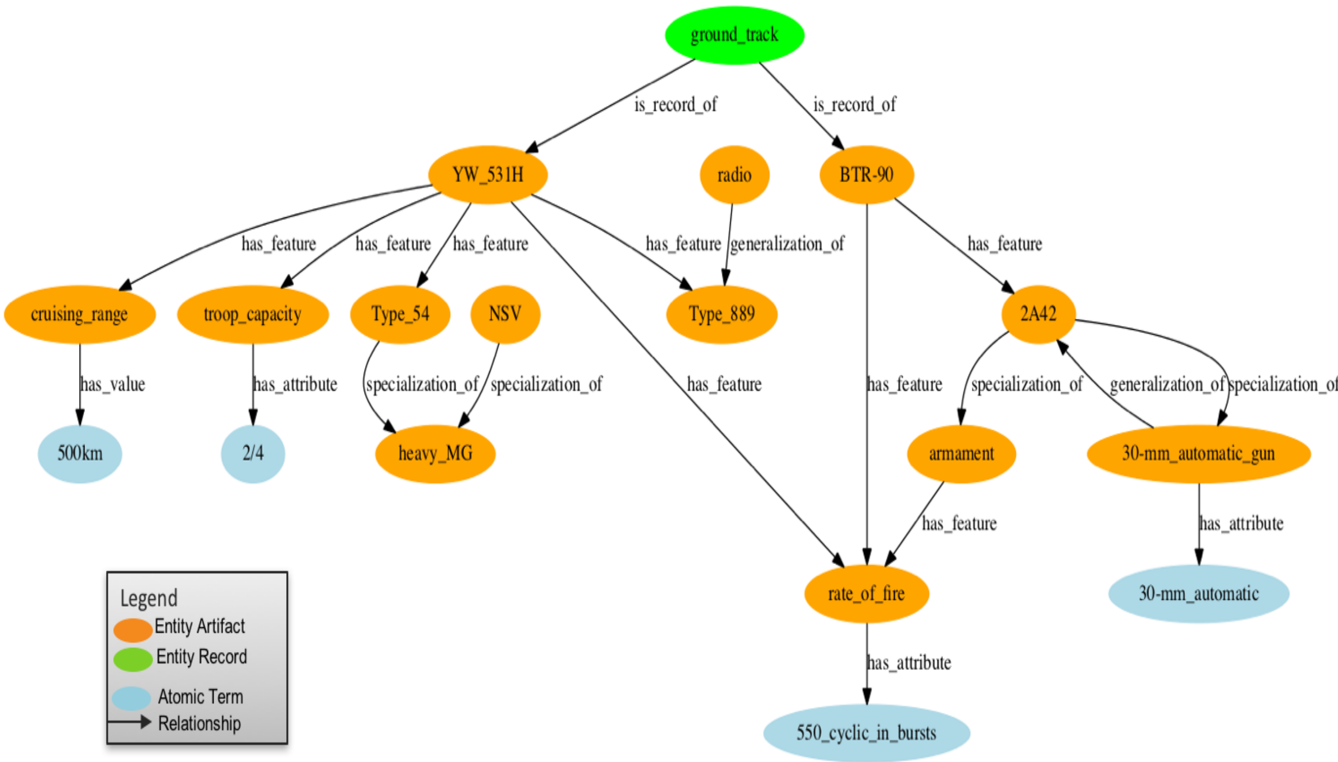

Fig. 4. Example illustrating use of meta–properties

elements. The 'generalization of' and 'specialization of' will determine inheritance.

Code Lists can be created in a similar fashion to how enumerated classes are created in ECore. Association Types can be created from 'is record of' triples.

A logical modeler determines whether an object of 'has attribute' should be considered Metadata. NIEM Augmentation and Extension Augmentation point and extensions are determined from 'derives from'. The logical modeler determines whether to create an augmentation point or an extension.

V. Conclusion

We described an approach to create a domain model in OWL for which logical models can be derived in a systematic way. Our approach is truly a domain model because it uses terminology from domain documents to create the ontology entities. In addition, the domain model contains the ontological relationships from the domain. For instance, it is able to specify that two concepts are related because one concept is a quality of another concept. In addition, it is able to capture role relationships.

We provided an overview of how we intend to use domain models created with our approach to generate logical models in Ecore and NIEM. We believe project managers will consider our approach suitable for their projects because it does not require expertise in ontologies and in-depth knowledge of CCO.

In the future, we plan to build a complete land combat domain model using the sources mentioned in Table I. We hope this domain model will be used as a common semantics for U.S. Army's initiative to use a single computing platform for multiple army battle command systems [7].

References

[1] *Interim report: ANSI/X3/SPARC Study Group on Data Base Management Systems*. Washington, D.C.: ACM, 1975.

[2] J. R. Schoening, D. K. Duff, D. A. Hines, K. M. Riser, T. Pham, G. H. Stolovy, J. Houser, R. Rudnicki, R. Ganger, and A. James, "PED fusion via enterprise ontology," in *Proceedings SPIE 9464,Ground/Air Multisensor Interoperability, Intergration, and Networking for Persistent ISR VI*. International Society for Optics and Photonics, May 2015.

[3] R. Arp, B. Smith, and A. D. Spear, *Building ontologies with basic formal ontology*. Mit Press, 2015.

[4] W. R. Thissell, R. Czajkowski, F. Schrenk, T. Selway, A. J. Ries, S. Patel, P. L. McDermott, R. Moten, R. Rudnicki, G. Seetharaman, I. Ersoy, and K. Palaniappan, "A Scalable Architecture for Operational FMV Exploitation," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 10–18.

[5] M. J. O'Connor and A. Das, "Acquiring OWL Ontologies from XML Documents," in *Proceedings of the Sixth International Conference on Knowledge Capture*, ser. K-CAP '11. New York, NY, USA: ACM, 2011, pp. 17–24.

[6] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro, *EMF: eclipse modeling framework*. Pearson Education, 2008.

[7] S. Lyngaas, "Four years on, Army common operating environment takes shape" *FCW*, Sep. 2015. [Online]. Available: https://fcw.com/articles/2015/09/22/army-mobile-computing.aspx