CUBRC

An Overview of the Common Core Ontologies

9 August 2017

Prepared by:
CUBRC, Inc.
4455 Genesee St., Buffalo, NY 14225
POC: Ron Rudnicki, Senior Research Scientist
716-204-5208
rudnicki@cubrc.org

1 Introduction

The Common Core Ontologies (CCO) comprise eleven ontologies that aim to represent and integrate taxonomies of generic classes and relations across all domains of interest. Accompanying these ontologies is a rule-based method for representing the content of any data source whatsoever through constructing domain ontologies as extensions of CCO. (See "Best Practices of Ontology Development".) In this paper, we describe the content and structure of CCO in order to assist developers of domain ontologies in locating mid-level ontology content, as well as those needing to map data sources to the CCO for ingest and querying purposes.

This document is structured as follows. Section 2 discusses CCO's design principles. Section 3 presents the semantic structure inherited from the upper-level ontology Basic Formal Ontology (BFO). Section 4 presents an overview of the content of each of the eleven mid-level ontologies comprising CCO together with the import structure that links them together. Section 5 provides a very broad overview of CCO's domain-level extension ontologies. Section 6 draws the document to a conclusion.

This document adopts the following typographical convention: ontology classes will be expressed in small caps (e.g., FUNCTION, OBJECT AGGREGATE) and ontology relationships will be expressed in bolded, italicized lowercase with individual words joined by underscores (***realizes***, ***has_part***).

OBJECT AGGREGATE ***has_part*** OBJECT
PROCESS ***realizes*** FUNCTION

2 Design

2.1 Ontology Language and Editing Software

CCO is implemented using Web Ontology Language (OWL) 2, which adds expressivity to the associated Resource Description Framework (RDF) and RDF Schema (RDFS). OWL, RDF and RDFS allows one to define hierarchies of classes and relationships (called object properties), create instances of classes (called individuals), link individuals to data values (via data properties), assign values to XML Schema Definition datatypes, assert relationships between classes (class expressions and class expression axioms) and between object properties (e.g., object or data subproperties, reflexivity, symmetry, and transitivity), or between classes and object properties (e.g., domain and range restrictions), as well as create useful annotations for classes, individuals, and relationships.

The full OWL 2 syntax and structural specification can be found here. See the following links for more information about RDF and RDFS. The CCO was built using the free, open-source OWL ontology editor Protégé (Stanford Center for Biomedical Informatics Research), which can downloaded here.

2.2 Realism

The goal of traditional data models is to represent data elements and relationships relevant to the design needs of their respective databases. By contrast, the CCO adopts a "realism-based" approach (Smith, 2008; Smith and Ceusters, 2010), according to which an ontology should be designed to model not only data, but also the entities data is about. This approach stems from the conviction that disparate ways of capturing data are best rendered interoperable by rendering them conformant to the ways things actually are. Realism implies, then, that any given assertion in an ontology can be evaluated on the basis of an objective criterion: *Is the assertion true?* Accordingly, this approach shifts ontology development away from the parochial concerns of particular implementations and toward expanded interoperability.

As will be discussed more fully in a subsequent section, adopting realism also invites a distinction between representations of *real* entities and representations of *information* entities. For example, it allows one to distinguish explicitly between a representation of a patient John Doe and a representation of the electronic medical records that are *about* John Doe. The former represents a real person, whereas the latter represents (fallible) data about the real person.

2.3 Modularity

The CCO adopts a modular approach to ontology development, according to which different ontologies are responsible for representing reality at different levels of granularity or in different domains. For example, an ontology for representing watercraft wouldn't define classes for engine parts or radios, even though most watercraft have engines and radios. Specifically, the CCO follows the categorization of ontologies into upper-level, mid-level, and domain-level ontologies:

- An upp*er-level ontology* is one that identifies those generic types of entities which belong to the formal structure of the world (e.g., OBJECT, PROCESS, SPATIAL REGION), together with formal specifications of how those types of entities are related to others (e.g., OBJECT ***participates_in*** PROCESS).
- A *mid-level ontology* is one that adds general content to the structure outlined in the upper-level ontology by identifying types of entities which directly specialize the upper-level types, but which are also common to many domains of interest. Classes that appear in mid-level ontologies are still fairly basic with respect to particular knowledge domains and often require further specialization to be useful for data modeling (e.g., PERSON, ACT OF COMMUNICATION, or GEOPOLITICAL ENTITY).
- A *domain-level ontology* is one that identifies types that further specialize the basic types from one or more mid-level ontologies. Domain ontologies describe objects, events, and relationships that are of interest to a more limited number of knowledge domains (e.g., INTELLIGENCE ANALYST ROLE, PORTION OF AMMONIUM NITRATE, or ACT OF WATERCRAFT REGISTRATION).

2.4 Namespaces, URIs, and Term Curation

Following the Semantic Web rules for linked data (Berners-Lee, 2006; Bizer, et al., 2009), the CCO adopts HTTP-based uniform resource identifiers (URIs) as names for entities. With one exception (see Section 4.11), CCO utilizes a single HTTP namespace for the URIs of all classes, properties, and individuals:

> http://www.ontologyrepository.com/CommonCoreOntologies/

Utilizing a common namespace for ontology terms yields two principal benefits. First, a common namespace facilitates RDF querying by allowing users to define a single prefix for CCO terms. Second, a common namespace makes it easier to refactor terms from one ontology to another as the need arises.

The name of a class, property, or individual is introduced following the forward slash (/) at the end of the namespace. For example, the CCO class person has the URI:

> http://www.ontologyrepository.com/CommonCoreOntologies/Person

The CCO adopts the following conventions for naming classes, properties, and individuals which contain multiple words. For classes and individuals, camel case is used (e.g., ArtifactModel, JohnDoe). For properties, each word is lowercase and joined by underscores (e.g., described_by). Thus, the full URIs would look as follows:

http://www.ontologyrepository.com/CommonCoreOntologies/ArtifactModel	(Class)
http://www.ontologyrepository.com/CommonCoreOntologies/JohnDoe	(Individual)
http://www.ontologyrepository.com/CommonCoreOntologies/described_by	(Property)

Because CCO ontologies are developed modularly (see Section 2.3), different classes, properties, and individuals are curated in different ontologies, rather than in a single ontology. Accordingly, each CCO class, property, and individual is annotated with information about which ontology contains it and is responsible for its continued curation. Specifically, the annotation property ***is_curated_in_ontology*** contains the URI of the ontology itself. For example, the class Person is defined and curated in the Agent Ontology, which has the URI:

> http://www.ontologyrepository.com/CommonCoreOntologies/Mid/AgentOntology

The use of 'Mid' and 'Domain' in the ontology URI indicates whether an ontology is considered to be mid- or domain-level in its scope. Note that because CCO extends from, but does not curate, BFO or RO, no BFO classes or RO properties share this common namespace.

2.5 Minimal Asserted Class Axiom Expressions

CCO is designed such that explicit connections between classes, other than subclass relationships, are not typically asserted. For example, there are no class axiom expressions linking the CCO class PERSON to classes such as WEIGHT or OCCUPATION, as might analogously be done in the model for a relational database. The reason for this is that a guiding principle in the development and application of the CCO is to produce a vocabulary that can integrate all information from any data source about every type of entity, and not to prescribe the types of information that should be collected or queried about a particular type of entity.

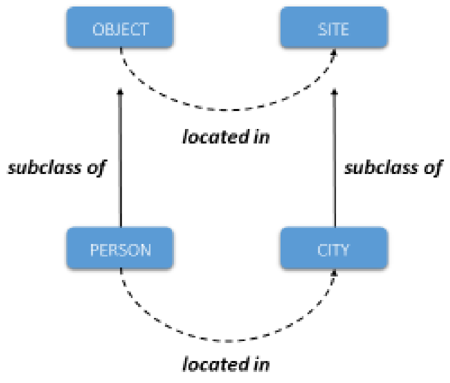
2.6 Minimal Object Properties

One of the design principles of the CCO is to keep the number of relationships (in OWL, object properties) between classes or individuals to a minimum. The reasoning behind this principle is that the purpose of the CCO is to enable the cross-linking of as many disparate data sets as required. But the syntax of OWL prohibits object properties from being linked to other information. Thus, the less information that is stored in object properties the more information that can be linked to other sources.

3 Upper-Level Semantic Framework

CCO is designed as a mid-level extension of Basic Formal Ontology (BFO) and the Relation Ontology (RO), an upper-level ontology framework widely used to structure and integrate ontologies in the biomedical domain (Arp, et al., 2015). BFO aims to represent the most generic categories of entity, and RO the most generic types of relations that hold between them, by defining a small number of classes and relations. CCO then extends from BFO-RO in the sense that every class in CCO is asserted to be a subclass of some class in BFO, and that CCO adopts the generic relations defined in RO (e.g., ***has_part***) (Smith and Grenon, 2004). Accordingly, CCO classes and relations are heavily constrained by the BFO-RO framework, from which it inherits much of its basic semantic relationships.

For example, the CCO class PERSON is asserted to be a subclass of the BFO class OBJECT and CITY as a subclass of BFO SITE. Now BFO specifies that instances the class OBJECT are related to instances of the class QUALITY by means of the RO ***located_in*** relation. Therefore, CCO likewise specifies that instances of the class PERSON are related to instances of the class CITY by means of the ***located_in*** relation.



Because CCO takes a top-down approach to ontology development, understanding this basic upper-level semantic framework is crucial for constructing domain-level CCO extension ontologies. The methodology behind the CCO does not require reinventing the semantics of an ontology anew for each domain. Rather, it categorizes entities and relations in a way conformant to the generic semantic structure inherited from BFO. The purpose of this section, then, is to present an overview of BFO's classes and relations and how they work together to construct a basic model of reality.

The five areas this section covers are: (1) objects and processes, (2) attributes, (3) time and place, (4) parthood and aggregation, and (5) fiat entities. The discussion of these areas in the first five subsections will ignore some of the finer details of BFO's class hierarchy, but the sixth subsection presents a snapshot of the complete BFO class hierarchy (see Arp, et al., 2015).

3.1 Objects and Processes

The principal ontological distinction in Basic Formal Ontology is between the class OBJECT and PROCESS (Smith, 1998; 2012; Bittner, et al., 2004). Examples of objects include persons, artifacts (vehicles, buildings, machines, tools), and natural objects (proteins, meteorites). Examples of processes in include actions and events (planning, artifact processing, criminal acts), states (the state of being employed), and changes (loss of employment). The relational pattern is:

OBJECT ***participates_in*** PROCESS
PROCESS ***has_participant*** OBJECT

Note that although most processes involve an object actively changing something or passively undergoing change, PROCESS also includes object states, in which an object does not change with respect to one of its attributes over some period of time. Thus, we can describe a person (an object) having the role of surgeon (an attribute) over some specific period of time. This notion of an object state is captured by the CCO class STASIS, which will be discussed more fully in Section 4.4.

3.2 Attributes

BFO draws a distinction between several types of object attributes (Smith, 1998). It distinguishes first between attributes which can migrate between objects and those which are tied to just one specific object. In BFO terminology, the former are called GENERICALLY DEPENDENT CONTINUANT and the latter SPECIFICALLY DEPENDENT CONTINUANT (Ceusters and Smith, 2015). The most perspicuous example of the former is information, which can belong to different objects simultaneously (copies of a book, or of a file on multiple hard drives). Information is discussed in greater depth in Section 4.1. In BFO, the latter class (SPECIFICALLY DEPENDENT CONTINUANT) is divided into several subclasses, which will be discussed below. The ***bearer_of*** relation and its inverse, ***inheres_in***, is used between objects and their attributes:

OBJECT ***bearer_of*** GENERICALLY DEPENDENT CONTINUANT
GENERICALLY DEPENDENT CONTINUANT ***inheres_in*** OBJECT

OBJECT ***bearer_of*** SPECIFICALLY DEPENDENT CONTINUANT
SPECIFICALLY DEPENDENT CONTINUANT ***inheres_in*** OBJECT

One important subclass of SPECIFICALLY DEPENDENT CONTINUANT is the class QUALITY, which comprises the overt or manifest attributes of an object (mass, length, coloring). QUALITY is distinguished from DISPOSITION, which represents physical attributes of objects that are realized in processes (the genetic disposition of a patient to develop colon cancer), and ROLE, which represents socially grounded attributes realized in the exercise of that role (a person's role as surgeon, a computer resource's role as network server) (Arp and Smith, 2008). Furthermore, DISPOSITION contains the subclass FUNCTION, which comprises dispositions designed by human invention (the function of a grenade to explode) or by natural evolutionary forces (the function of the heart to pump blood throughout an organism) (Spear, et al., 2016).

Instances of DISPOSITION, ROLE, and FUNCTION have a special relationship to the processes in which they are realized, namely, the relation of realization:

> DISPOSITION ***realized_by*** PROCESS
> PROCESS ***realizes*** DISPOSITION
>
> ROLE ***realized_by*** PROCESS
> PROCESS ***realizes*** ROLE
>
> FUNCTION ***realized_by*** PROCESS
> PROCESS ***realizes*** FUNCTION

Processes can also have attributes, which are represented by the BFO class PROCESS PROFILE. Specifically, a process profile is an abstraction of some relevant facet of a process (typically, a change or rate of change of some object attribute). For example, the speed of some vessel (the rate of its distance travelled divided by the time elapsed) can be represented as a process profile of the movement in which that vessel participates. The relationship between processes and process profiles is one of processual parthood:

> PROCESS ***has_process_part*** PROCESS PROFILE
> PROCESS PROFILE ***is_part_of_process*** PROCESS

3.3 Time and Place

Objects and processes can be related to times and places (Bittner, et al., 2004). Places are represented in two distinct, complementary ways by the BFO classes SITE (an immaterial region bound by an object) and SPATIAL REGION (an immaterial region of space-time relative to a frame of reference). Examples of SITE include: a person's chest cavity, the site which contains a building, and the geospatial region of a country. An example of SPATIAL REGION would be the heliosphere (the spatial region in which solar wind has significant influence).

The notions of site and spatial region differ in subtle ways, but a simple example can illustrate the difference between them: the hold of a ship which travels from point A to point B is a site which moves from one spatial region to another. In other words, spatial regions are inert whereas sites can travel with the physical objects which bound them. That being said, at any particular time a site will coincide with some spatial region.

Objects and processes have different relationships to sites and spatial regions. For objects, the relationships are:

> OBJECT ***located_in*** SITE
> SITE ***location_of*** OBJECT
>
> OBJECT ***located_in*** SPATIAL REGION
> SPATIAL REGION ***location_of*** OBJECT

For processes, the relationships are:

PROCESS ***occurs_at*** SITE
SITE ***is_site_of*** PROCESS

PROCESS ***occurs_at*** SPATIAL REGION
SPATIAL REGION ***is_site_of*** PROCESS

As for time, BFO defines the class TEMPORAL REGION, on which processes occur:

PROCESS ***occurs_on*** TEMPORAL REGION
TEMPORAL REGION ***is_temporal_region_of*** PROCESS

Keep in mind the distinction between the two relations occurs at (place) and occurs on (time). Also note that objects (and their attributes) have no direct asserted relationship to temporal regions. Relations between objects and temporal regions are always mediated via processes. The CCO Time Ontology (Section 4.7) extends from the BFO class temporal region to define various types of temporal intervals (e.g., day, month, and year).

3.4 Parthood and Aggregation

BFO defines a relationship between objects and their parts:

OBJECT ***has_part*** OBJECT
OBJECT ***part_of*** OBJECT

It also defines a relationship between processes and their parts, which was previewed in the previous discussion of process profiles:

PROCESS ***has_process_part*** PROCESS
PROCESS ***is_part_of_process*** PROCESS

BFO also introduces the class OBJECT AGGREGATE as a way of representing groups of objects. The relationship here is also one of parthood:

OBJECT AGGREGATE ***has_part*** OBJECT
OBJECT ***part_of*** OBJECT AGGREGATE

3.5 Fiat Entities

Lastly, BFO defines a few classes for “fiat entities,” i.e., entities which are not demarcated by bona fide, physically continuous boundaries (Smith, 2001). One class of these fiat entities is FIAT OBJECT, which includes examples such as the northern hemisphere of a planet or the bow of a ship. Another is CONTINUANT FIAT BOUNDARY, which includes examples such as the boundary between two nations or the boundary between geological layers of the earth (Smith, 1995). At the heart of BFO’s notion of fiat entity is that certain entities can be demarcated and identified despite there being no break in physical continuity. CCO then defines a relationship between boundaries and sites derived from the Region Connection Calculus 8 (Randell, et al., 1992):

SITE ***externally_connects_with*** CONTINUANT FIAT BOUNDARY
CONTINUANT FIAT BOUNDARY ***externally_connects_with*** SITE

3.6 BFO Class Hierarchy Overview

The following page presents an at-a-glance view of the class hierarchy of Basic Formal Ontology, with all classes represented:

- ENTITY
  - CONTINUANT
    - INDEPENDENT CONTINUANT
      - IMMATERIAL ENTITY
        - SITE
        - SPATIAL REGION
          - ZERO-DIMENSIONAL SPATIAL REGION
          - ONE-DIMENSIONAL SPATIAL REGION
          - TWO-DIMENSIONAL SPATIAL REGION
          - THREE-DIMENSIONAL SPATIAL REGION
        - FIAT CONTINUANT BOUNDARY
          - ZERO-DIMENSIONAL CONTINUANT FIAT BOUNDARY
          - ONE-DIMENSIONAL CONTINUANT FIAT BOUNDARY
          - TWO-DIMENSIONAL CONTINUANT FIAT BOUNDARY
      - MATERIAL ENTITY
        - OBJECT
        - OBJECT AGGREGATE
        - FIAT OBJECT
    - SPECIFICALLY DEPENDENT CONTINUANT
      - QUALITY
      - REALIZABLE ENTITY
        - DISPOSITION
          - FUNCTION
        - ROLE
    - GENERICALLY DEPENDENT CONTINUANT
  - OCCURRENT
    - PROCESS
    - PROCESS BOUNDARY
    - TEMPORAL REGION
      - ZERO-DIMENSIONAL TEMPORAL REGION
      - ONE-DIMENSIONAL TEMPORAL REGION
    - SPATIOTEMPORAL REGION

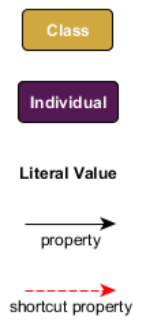
4 Mid-Level Content

The eleven mid-level ontologies that comprise the Common Core are:

1. Information Entity Ontology
2. Agent Ontology
3. Quality Ontology
4. Event Ontology
5. Artifact Ontology
6. Time Ontology
7. Geospatial Ontology
8. Units of Measure Ontology
9. Currency Unit Ontology
10. Extended Relation Ontology
11. Lewisian Relation Ontology

The content of each of these ontologies is built within the upper-level semantic framework defined by BFO-RO. This means that the basic class hierarchy, as well as many relationships, are defined by BFO and merely inherited by CCO. Consequently, compliance with the semantics of the CCO requires compliance with the semantic framework described in Section 3.

In the diagrams that follow, these symbols are used to represent classes, individuals, properties, and literal values:



The dashed red arrow that links an individual to a literal value is for presentation purposes <u>only</u>. It indicates that the intermediary nodes for instances of the classes INFORMATION CONTENT ENTITY and INFORMATION BEARING ENTITY (see Section 4.1) are not being shown in the diagram. This convention is used to render some diagrams in this document more perspicuous and does <u>not</u> reflect the actual semantics of the CCO. The reader should especially note that this purely graphical convention is different from the use of the ***is_tokenized_by*** annotation property to link instances of INFORMATION CONTENT ENTITY directly to literal values (see the discussion in Section 4.1).

4.1 Information Entity Ontology

The Information Entity Ontology represents types and provenance of information. Significantly, the ontology draws a distinction in representing (1) the *content* of some piece of information, and (2) the *expressions* of that content in some *medium* (Smith, et al., 2013). The former are represented with the class INFORMATION CONTENT ENTITY, whereas the latter are represented with the class INFORMATION BEARING ENTITY. This distinction is necessary for connecting pieces of data (e.g., the values in a data table) to entities in the real world (e.g., books, documents, severs, databases) and for tracking the provenance of data.

An information bearing entity is a concrete, material object which bears some information content, and which is linked to particular data values:

INFORMATION BEARING ENTITY ***bearer_of*** INFORMATION CONTENT ENTITY
INFORMATION CONTENT ENTITY ***inheres_in*** INFORMATION BEARING ENTITY

INFORMATION BEARING ENTITY ***has_text_value*** *Literal*
INFORMATION BEARING ENTITY ***has_integer_value*** *Literal*

A single information content entity can inhere in multiple information bearers. For example, the content of a document can reside simultaneously in multiple copies of that document. Likewise, a single name or identifier can be reproduced in many physical objects (written on a driver's license, a social security card, a nametag, etc.). What is essential to keep in mind is: *Whenever identical content is found in multiple sources, that content is a single instance of INFORMATION CONTENT ENTITY.* A person only has *one* name, but that name (content) can be found in *many* particular physical tokens. (See the discussion of the BFO class GENERICALLY DEPENDENT CONTINUANT in Section 3.2.)

An information content entity stands in a relation of "aboutness", or reference, to the entity the content is about (Ceusters and Smith, 2015):

INFORMATION CONTENT ENTITY ***is_about*** ENTITY
ENTITY ***is_subject_of*** INFORMATION CONTENT ENTITY

Note that the BFO class ENTITY is the most generic parent class in the ontology: every BFO class is a subclass of ENTITY. Note also, then, that there are no restrictions on what type of entity an information content entity can be about. An instance of INFORMATION CONTENT ENTITY can be about an instance of OBJECT or PROCESS or SPATIAL REGION or any other class.

The ***is_about*** relation divides into three subproperties, each of which represents a different relation between information content and what that information is about, namely: ***describes***, ***prescribes***, and ***designates***. The ***describes*** relation is used for information such as reports and representations (images), the ***prescribes*** relation is used for information such as plans and artifact specifications, and the ***designates*** relation is used for information such as names and other identifiers.

INFORMATION CONTENT ENTITY ***describes*** ENTITY

ENTITY ***described_by*** INFORMATION CONTENT ENTITY

INFORMATION CONTENT ENTITY ***prescribes*** ENTITY
ENTITY ***prescribed_by*** INFORMATION CONTENT ENTITY

INFORMATION CONTENT ENTITY ***designates*** ENTITY
ENTITY ***designated_by*** INFORMATION CONTENT ENTITY

The Information Entity Ontology also defines subclasses of INFORMATION CONTENT ENTITY which correspond to each of these three relations: DESCRIPTIVE INFORMATION CONTENT ENTITY (for the ***describes*** relation), DIRECTIVE INFORMATION CONTENT ENTITY (for the ***prescribes*** relation) and DESIGNATIVE INFORMATION CONTENT ENTITY (for the ***designates*** relation).

An example of the first would be the content of a newspaper describing a weather event:

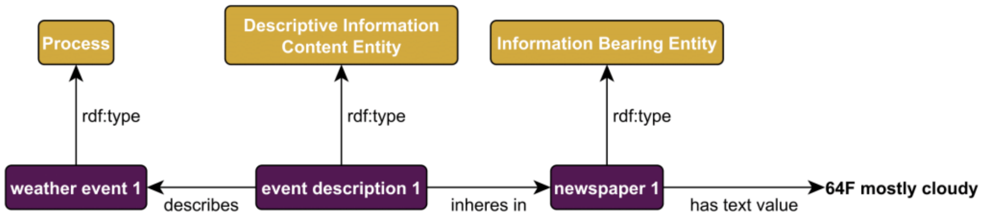


Figure 1: A *description* of an event

An example of the second would be the content of a plan for some military operation:

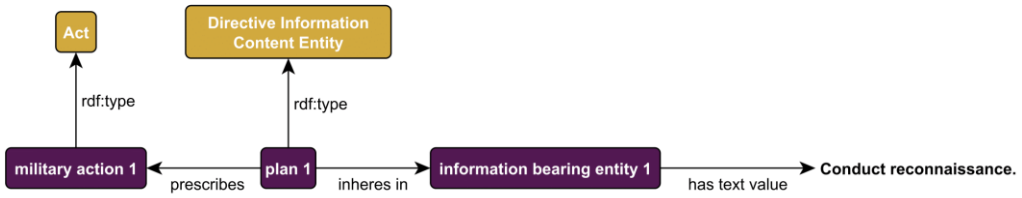


Figure 2: A *prescription* of an action

An example of the third would be the content of a proper name or an ID number:

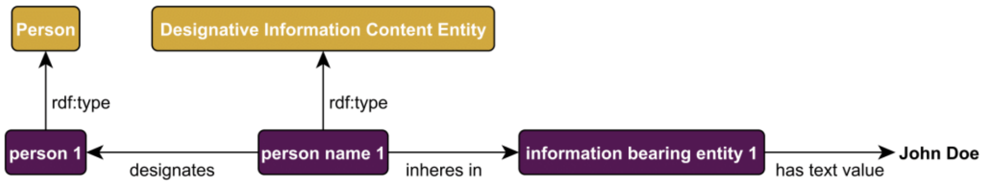


Figure 3: The *designation* of a person

The trifold distinction—between information content, information bearers, and the subject of that information—allows a CCO-aligned dataset to track the *provenance* of information: its origin, history, and quality. In some situations, however, users may wish to represent information about individuals without tracking provenance. To meet this need, the CCO introduces an OWL annotation property, ***is_tokenized_by***, which link literal values directly to instances of INFORMATION CONTENT ENTITY.

INFORMATION CONTENT ENTITY ***is_tokenized_by*** *Literal*

Thus, Figure 3's example of a person named John Doe would be modeled as follows:

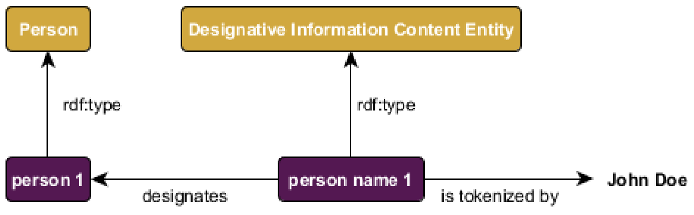


Figure 4: Annotating information with literal values.

4.2 Agent Ontology

The Agent Ontology represents agents, their qualities, and the roles they have in various contexts. The class AGENT comprises both individual agents (PERSON) and coordinated groups of individuals (ORGANIZATION). Examples of agents' qualities include HEIGHT, WEIGHT, and EYE COLOR. Examples of agents' roles include CITIZEN ROLE, OCCUPATION ROLE, and ALLY ROLE. Thus, the Agent Ontology enables the representation of a description of the roles an agent has in a particular context:

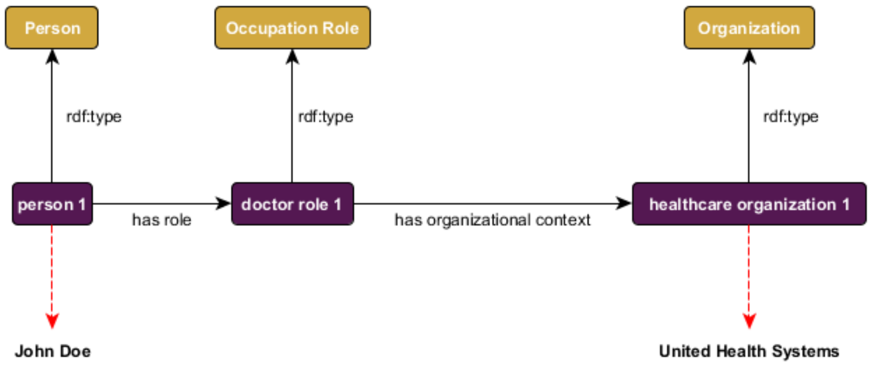


Figure 5: Persons bear roles in organizational contexts

Or the events in which agents are participants, e.g., a college graduation:

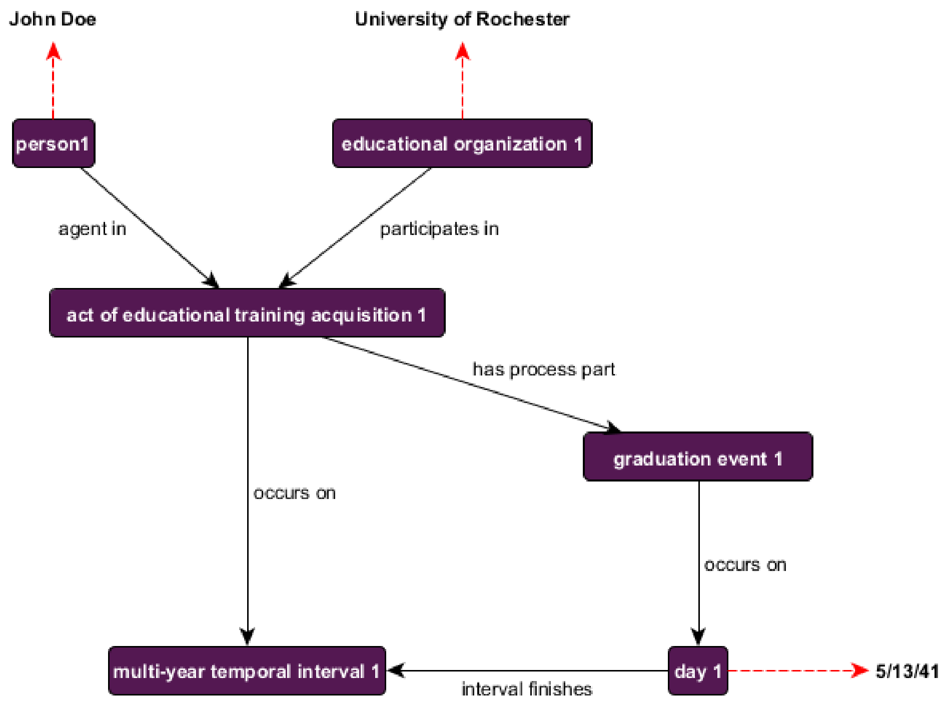


Figure 6: Persons and things participate in events, which occur at times

4.3 Quality Ontology

The Quality Ontology represents the attributes of agents, artifacts, and events. These attributes change over time and are often used to differentiate objects from others of the same or similar type. Since attributes are always dependent on other entities (their bearers), the classes contained in the Quality Ontology are only of value when used in combination with classes from other ontologies. Much of the content is adapted from the Phenotypic Trait Ontology (PATO) and extends the BFO classes: QUALITY, REALIZABLE ENTITY (parent class of DISPOSITION and ROLE), and PROCESS PROFILE. Subclasses of qualities in Quality Ontology include: SHAPE QUALITY, WEIGHT, and TEMPERATURE. Subclasses of REALIZABLE ENTITY include: MAGNETISM, COLOR, and VULNERABILITY. Subclasses of PROCESS PROFILE include: SPEED and FREQUENCY.

Thus, the Quality Ontology enables one to represent the qualities of a person:

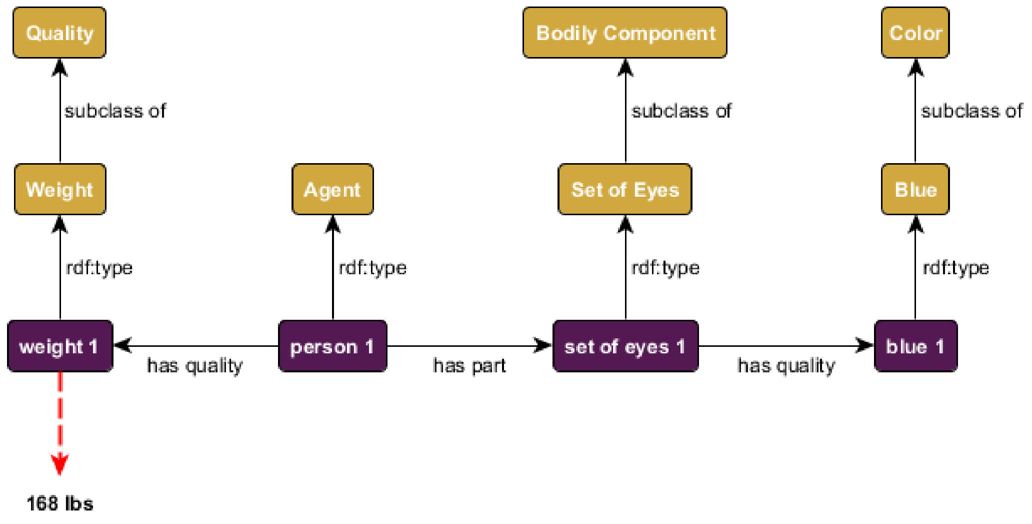


Figure 7: Attributes of a person

Or the speed of a patrol ship:

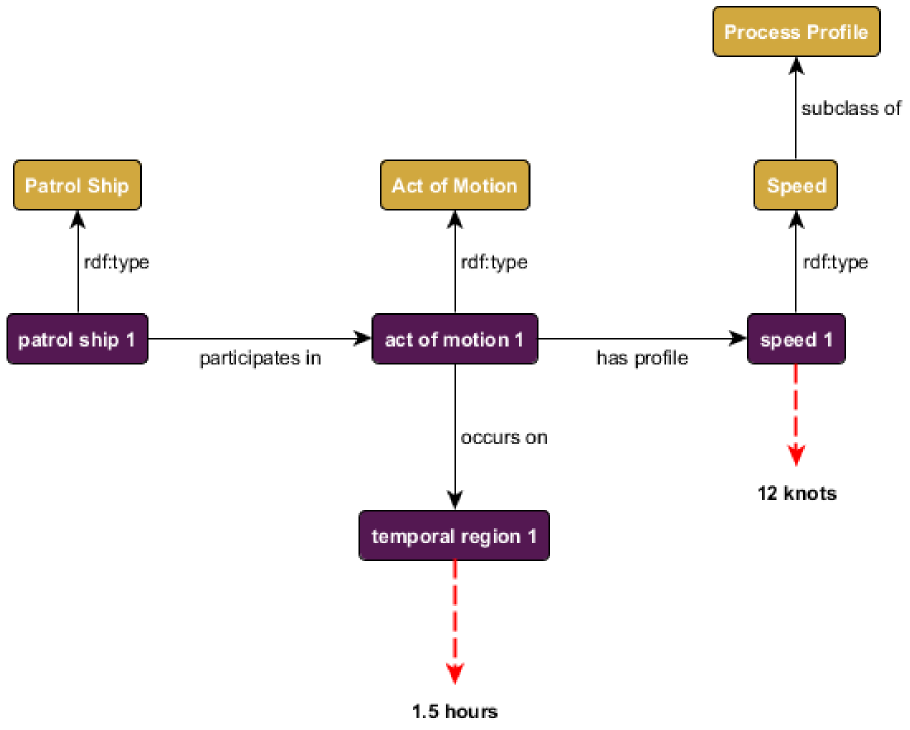


Figure 8: Attributes of a moving ship

4.4 Event Ontology

The Event Ontology represents events, actions, processes, and states of the world in which agents, artifacts, etc., are participants. The general relationships between objects and processes, and between processes and places and time, have been described already (Sections 3.1, 3.3). The Event Ontology builds off of these basic relationships to define a richer vocabulary for describing a variety of actions and events. For example, if we think of a person's birth as an event, we can represent the person as a participant in the birth, and we can express when and where the event occurred:

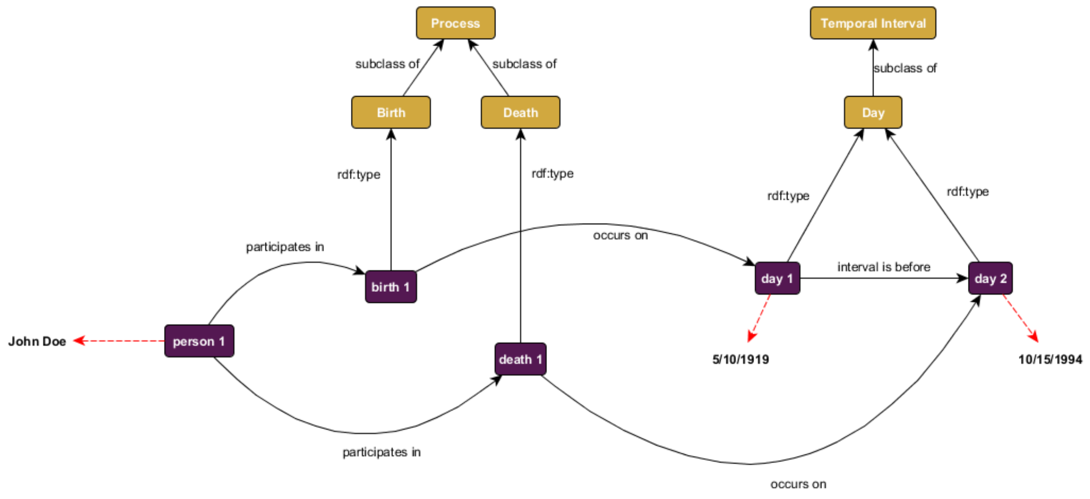


Figure 9: A person's date of birth and date of death

Moreover, if the process in question is an intentional act, then we can represent the agent which has some causal role in that act via use of a relation that is more specific than mere participation.

AGENT ***agent_in*** INTENTIONAL ACT

INTENTIONAL ACT ***has_agent*** AGENT

The Event Ontology also allows one to represent the fact that objects change their attributes over time in virtue of their participation in processes. Such change is manifested in the gain and loss of qualities, functions, and roles, which also participate in those processes. The restriction to binary relationships in OWL makes changes in attributes difficult to represent. The solution to this problem in CCO is to treat the gain and loss of attributes as events to which the object, attribute, and time are related via binary relations. To represent the specific time when an attribute was gained, CCO introduces the class CHANGE (a subclass of PROCESS). For example, this allows one to represent Barack Obama's gaining the role of president:

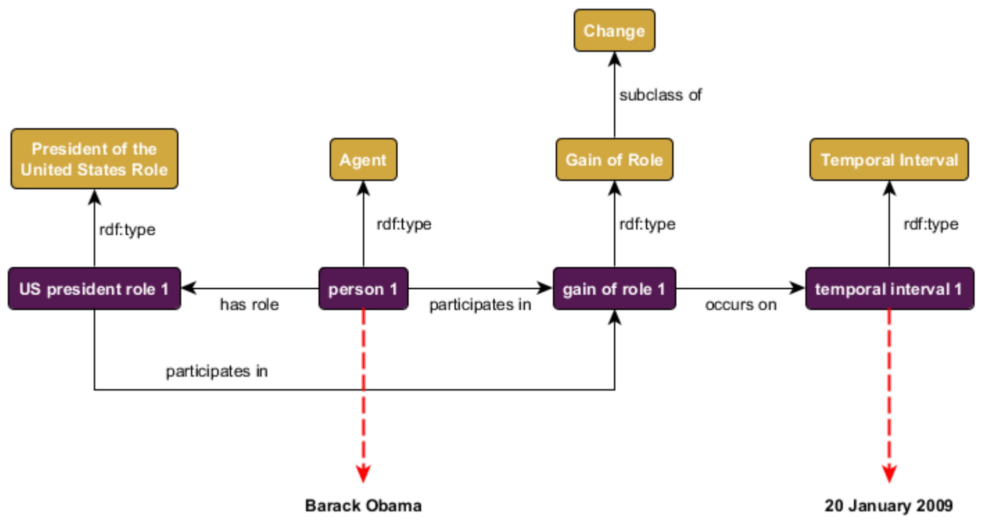


Figure 10: Obama gains the role of president

In other cases it is important to merely represent the duration of time for which some attribute remains fixed. CCO represents this by means of the class STASIS. This class allows one, for instance, to represent how long Barack Obama had the role of president:

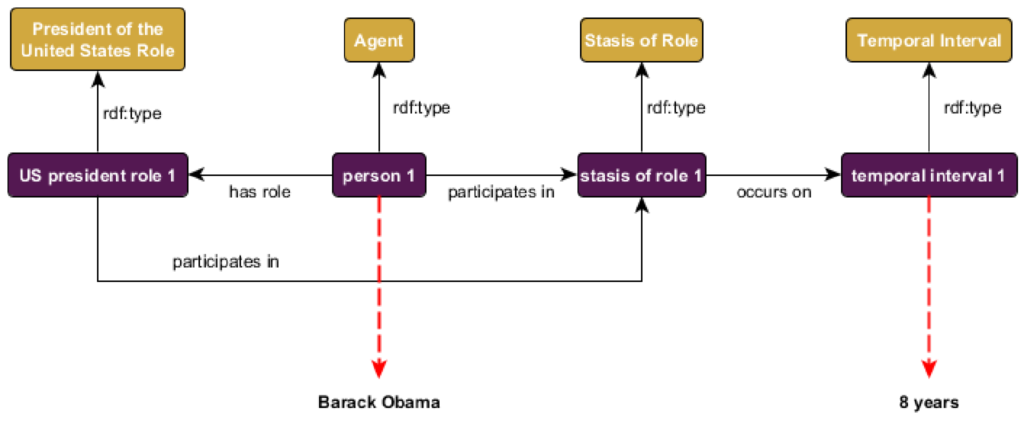


Figure 11: Obama bears the president role for 8 years

4.5 Artifact Ontology

The Artifact Ontology represents artifacts, their designed qualities and functions, and the specifications that provide models for artifact production or modification. It contains terms representing general types of artifacts including: COMMUNICATION INSTRUMENT, FACILITY, VEHICLE, and WEAPON, and subclasses of each. Moreover, the ontology allows a user to make assertions about which qualities or functions an artifact is designed to have. For example, that a grenade has a designed function of exploding:

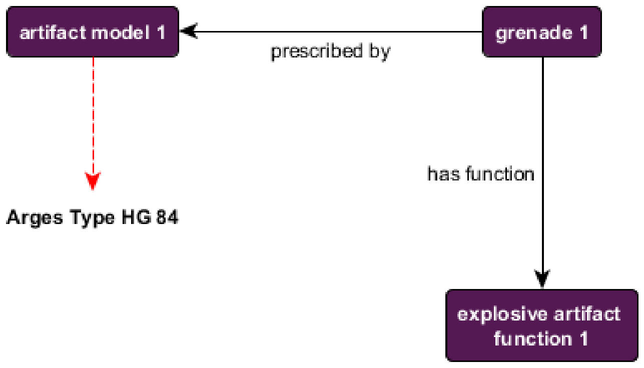


Figure 12: An artifact designed to have a function

4.6 Geospatial Ontology

The Geospatial Ontology represents geospatial regions and sites, including classes such as: CITY, STATE, COUNTRY, ENVIRONMENTAL FEATURE, and BOUNDING BOX. Thus, this ontology provides the basic vocabulary for describing the locations of agents and occurrences of events. For example, it allows us to assert that the Empire State Building is located in New York City, which is located in turn in New York State:

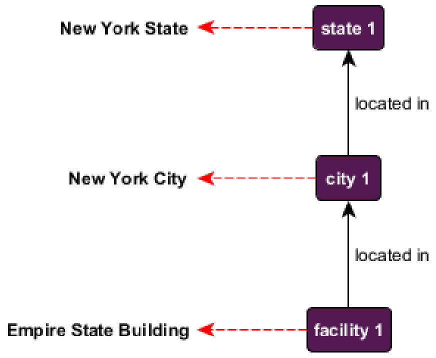


Figure 13: The location of the Empire State Building

Alternatively, it allows us to assert that the event of a person's birth occurs at a particular city:

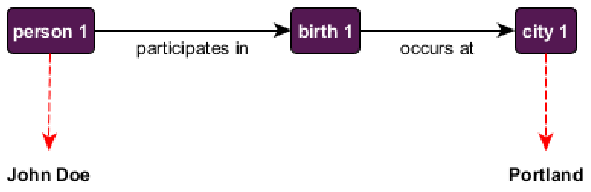


Figure 14: Where a birth event occurs

In addition, the ontology contains a host of formal geospatial relations derived from the Regional Connection Calculus 8 (e.g., ***disconnected_with***, ***externally_connects_with***, and ***overlaps_with***), which together provide a basis for basic geospatial reasoning.

4.7 Time Ontology

The Time Ontology represents temporal intervals such as: YEAR, MONTH, DAY, MORNING, and MULTI-HOUR TEMPORAL INTERVAL. In addition, the ontology contains temporal relations (e.g. ***interval disjoint***, ***interval meets***, ***interval overlaps***, and ***interval starts***) that provide a basis for basic temporal reasoning. This ontology provides the basic vocabulary for describing when events occur. For example, it allows us to assert that a person's birth happened on a particular day, which is part of a particular month, which is part of a particular year:

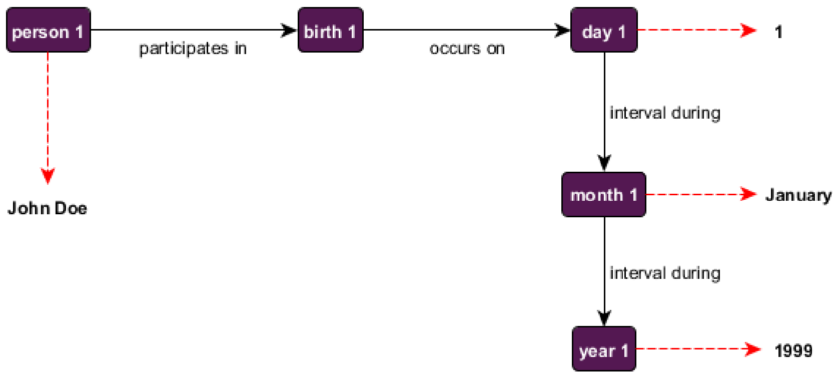


Figure 15: When a birth event occurs

Note that information about the times and locations of events is often imprecise. One source might describe an event that occurred in Iraq in the Spring of 2004, another an event that occurred in Baghdad in May of 2004, and another an event that occurred in the al-Kadhimya neighborhood on May 17th at 1:38pm EDT. If an entity resolution algorithm should determine that all three events are in fact one and the same, it becomes difficult to express this finding in a data model that has only tokens to represent times and places.

In CCO, one can facilitate the consolidation of data about the time and place of some event by taking advantage of the fact that every event occurs in some unique time and place. Since time and place are treated as entities rather than tokens, the "three" events, times, and places from the original sources become resolved to a single event occurring at a single time and place, where the time and place are each described (more or less accurately) in three distinct ways:

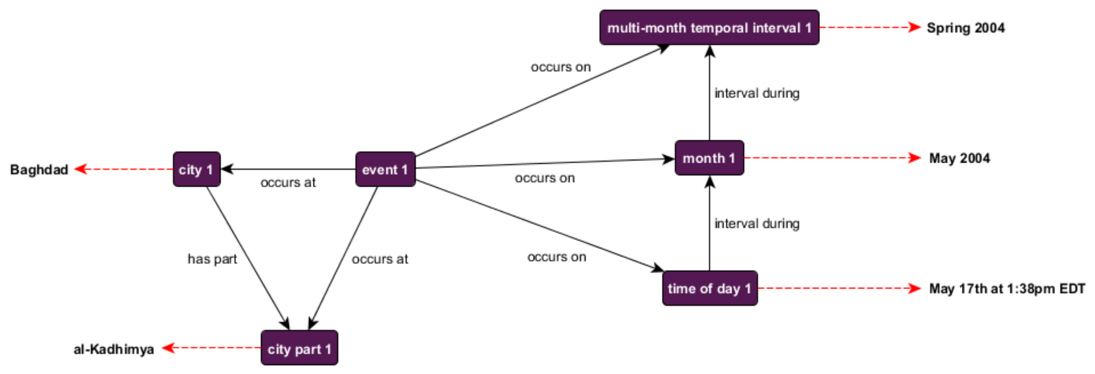


Figure 16: Imprecise times and dates for an event

4.8 Units of Measure Ontology

The Units of Measure Ontology represents the standardly employed units of measurement, and as such is complementary to the Information Entity Ontology. This ontology extends from the Information Entity Ontology class MEASUREMENT UNIT, and provides several subclasses of measurement unit, such as: MEASUREMENT UNIT OF AREA, MEASUREMENT UNIT OF ENERGY, and MEASUREMENT UNIT OF LENGTH. Specific measurement units are instances of these classes, examples of which include: Acre, Horsepower, and Kilometer, respectively.

Keep in mind that individual measurement units (Acre, Horsepower, Kilometer) are linked to instances of INFORMATION BEARING ENTITY, **not** INFORMATION CONTENT ENTITY. The reason is that a measurement of some phenomenon can be expressed multiple ways, differing only in how they express that content—i.e., which unit of measurement they employ. For example, two measurements of length can convey the exact same content in two different units:

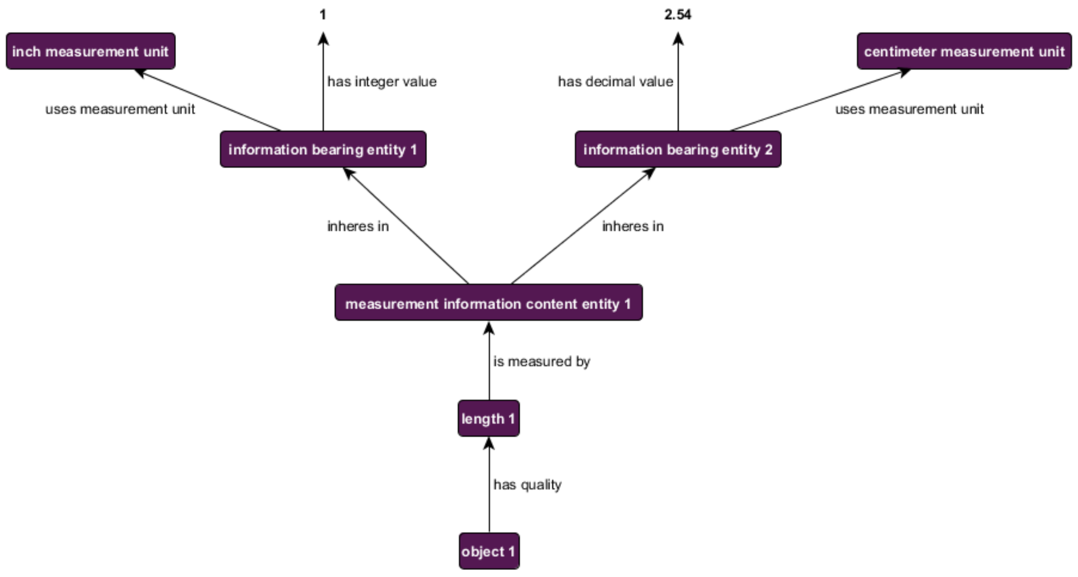


Figure 17: Using two different measuring units

4.9 Currency Unit Ontology

The Currency Unit Ontology represents monetary currency, and is likewise complementary to the Measurement Unit and Information Entity Ontologies. This ontology represents currency as another subclass of MEASUREMENT UNIT, namely, a unit for measuring financial value. Its sole class MEASUREMENT UNIT OF CURRENCY has numerous instances, e.g., United States Dollar, United Kingdom Pound, and Swiss Franc. Graphically, currency units are treated exactly as other kinds of measurement units:

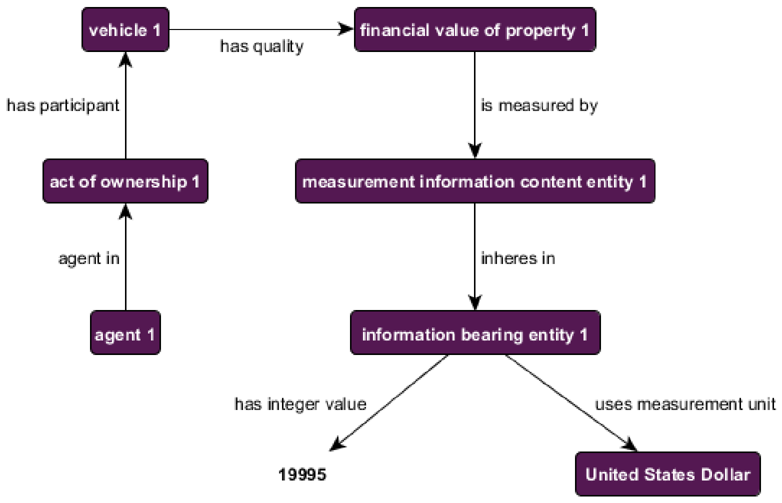


Figure 18: Expressing financial value in numbers and currency units

4.10 Extended Relation Ontology

One of the design principles of CCO is to keep the number of relationships to a minimum (Section 2.6). With this principle in mind, it remains the case that the object properties defined by RO are not sufficient to relate mid-level CCO classes. The Extended Relation Ontology fills this gap by defining approximately 75 object properties that link together the content of the rest of CCO. For example, whereas BFO relates objects to processes via the ***has_participant*** relation, the Extended Relation Ontology adds the ***has_input*** and ***has_output*** relations in order to differentiate, e.g., the roles of water and water vapor in the process of evaporation. Many of the object properties defined in the Extended Relation Ontology are utilized in the diagrams above.

The Extended Relation Ontology also includes several annotation properties for capturing metadata for classes, relationships, and individuals. Specifically, Extended Relation Ontology defines the following annotation properties:

acronym	An Alternative Label that consists of a shortened or abbreviated form of the rdfs:label and is used to denote the entity.
alternative_label	A term or phrase that may be used in place of the stated rdfs:label to denote the entity in question.
definition	A natural language explication of the meaning of the term.

definition_source	A citation of where all or some of the information used to create the term's definition was acquired from.
doctrinal_source	A Definition Source that consists of a formalized doctrine in which the term is authoritatively defined.
elucidation	A clarification or further explanation of a term beyond what is included in the definition or which is used when the term is primitive such that no non-circular definition can be given for it.
example_of_usage	A phrase, sentence or set of terms intended to convey the conventional usage of the term.
http_query_string	The text of an HTTP request that can be sent to a SPARQL Protocol service.
query_text	The text of a query that is associated with a class.

The addition of the ***http_query_string*** and ***query_text*** annotation properties are intended to provide a means of defining a user specific result set of information associated with a class or individual. The ***http_query_string*** is perhaps best suited to implementations that utilize a specific SPARQL endpoint where the ***query_text*** property enables the programmatic addition of other parameters such as the endpoint to which the query will be sent.

4.11 Lewisian Relation Ontology

The Lewisian Relation Ontology, named for the American philosopher David Lewis (d. 2001), provides a way of representing states of affairs that are prescribed (e.g., an action prescribed by a plan or some functionality prescribed by an artifact's design specification), but which do not exist yet, or may never exist.

The principal impetus behind this ontology can be illustrated with a simple example. Not all plans unfold exactly according to the plan. Therefore, one must draw a distinction between two events, both of which are related to the original plan: (*A*) how the event actually unfolded and (*B*) how that event *should* have unfolded. Likewise, not all artifacts perform as they are designed to. Accordingly, one could relate an artifact design specification both (*A*) to some actual artifact that was designed according to that specification, but which may not be functioning to specification, and (*B*) to the artifact as it *should* be functioning. In short, it is often necessary to distinguish the ideal plan (or artifact) from the actual plan (or artifact).

To differentiate between these two relationships, CCO utilizes an alternative set of object and data properties for relating instances of DIRECTIVE INFORMATION CONTENT ENTITY (or one of its subclasses, such as PLAN, ARTIFACT MODEL, or PERFORMANCE SPECIFICATION) to such non-real entities. These alternative properties defined in the Lewisian Relation Ontology are duplicates or counterparts of the properties defined in the Relation Ontology (see Section 3) and Extended Relation Ontology (see Section 4.10).

The Lewisian properties have the same name as the Extended Relation Ontology properties, but have an amended namespace, namely:

http://www.ontologyrepository.com/CommonCoreOntologies/LewisianRelationOntology/

Thus, the Extended Relation Ontology term “prescribes” and the Lewisian term “prescribes” differ in their URIs:

http://www.ontologyrepository.com/CommonCoreOntologies/prescribes
http://www.ontologyrepository.com/CommonCoreOntologies/LewisianRelationOntology/prescribes

Example: A plan prescribes that the USS Bremerton participate in a mission on April 25, 2017. However, when that plan is set in motion, for some reason, the USS Dallas is used instead. The mission is carried out on the same day as prescribed. The following diagram illustrates the way in which Lewisian relations, shown in blue, and standard CCO relations, shown in black, distinguish the planned entities from the existing, thus ensuring reliable querying over RDF triples.

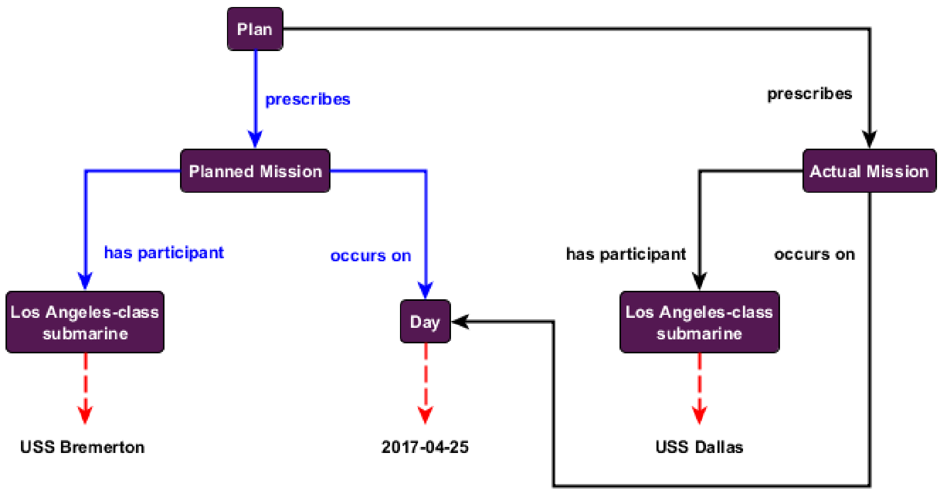


Figure 19: A planned vs. actual mission

A query that filters out the Lewisian properties will return only the triples associated with actual mission. Conversely, if only data associated with the plan is desired, then filtering triples associated with standard CCO relations will return only the planned graph structure, even in the case where the planned graph overlaps with the actual one. This provides a clear, easily implemented method for maintaining graph similarity between planned vs. actual entities, thus ensuring consistent semantics throughout.

4.12 Import Structure

The component ontologies of the CCO are connected to one another by the import relation (described elsewhere in this document as the extension relation). When one ontology imports another, the vocabulary of the imported ontology becomes available for use in the importing ontology. A roadmap of this import structure is illustrated in the figure below. Users can choose to import some or all of the ontologies of the suite as their needs require.

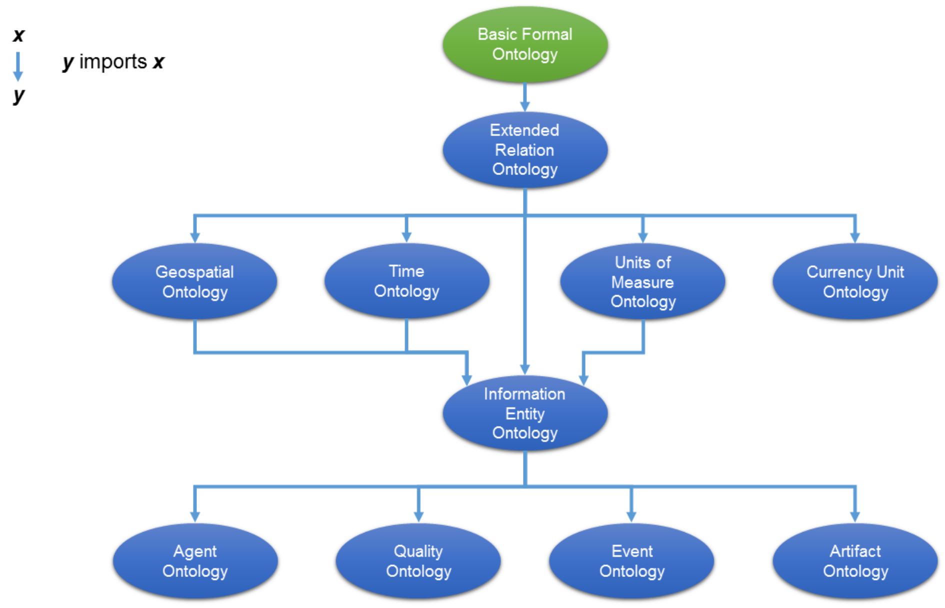


Figure 20: Common Core Ontologies architecture and importation relations

5 Domain-Level Content

The goal of the mid-level Common Core Ontologies is to represent entities of interest for a wide array of domains. However, they are not designed to capture the level of detail or specificity needed for users to annotate all the data within their respective domains. Users can readily develop domain-level ontologies extended from the mid-level content of Common Core.

To date, numerous domain-level extensions of the CCO have been developed:

1. Affective State Ontology
2. Agent History Ontology
3. Agent Information Ontology
4. Aircraft Ontology
5. Air Force Action Taken Codes Ontology
6. Air Force Aircraft Maintenance Ontology
7. Air Force How-Malfunction Codes Ontology
8. Air Force Maintenance Status Codes Ontology
9. Air Force Type Maintenance Designators Ontology
10. Air Force When-Discovered Codes Ontology
11. Army Universal Task List Ontology
12. Citizenship Ontology
13. Curriculum Ontology
14. Cyber Ontology
15. Ethnicity Ontology
16. Food and Allergy Ontology

17. Food Ontology
18. Hydrographic Feature Ontology
19. Joint Doctrine Ontology
20. Legal and Criminal Act Ontology
21. Maintenance Activity Ontology
22. Medical Information Ontology
23. Military Command and Control Ontology
24. Military Intelligence Ontology
25. Military Occupations Ontology
26. Military Operation Ontology
27. Military Planning Ontology
28. Occupation Ontology
29. Outer Space Ontology
30. Physiographic Feature Ontology
31. Planning Ontology
32. Sensor Ontology
33. Skills Ontology
34. Spacecraft Mission Ontology
35. Spacecraft Ontology
36. Space Event Ontology
37. Space Object Ontology
38. Transportation Infrastructure Ontology
39. Undersea Warfare Ontology
40. Watercraft Ontology

6 Conclusion

The Common Core Ontologies are a set of mid-level ontologies that provide terminology that describes human-activity. They provide the means to express complex relationships that other OWL-based vocabularies cannot. They were developed under the adherence of principles designed to maximize their ability to provide interoperability and reduce the costs associated with organizing enterprise information. They are grounded in doctrine, vetted against data, and subjected to quality tests. But most importantly the ontologies provide a starting point on which enterprise data interoperability can be built.

References

Arp, R. and Smith, B., 2008, 'Function, role, and disposition in Basic Formal Ontology,' *Proceedings of Bio-Ontologies Workshop*, Intelligent Systems for Molecular Biology (ISMB 2008), Toronto, 45-48.

Arp, R., Smith, B., and Spear, A.D. 2015, *Building Ontologies with Basic Formal Ontology*, Cambridge, MA: MIT Press.

Berners-Lee, T., 2006, 'Linked data – design issues,'

<https://www.w3.org/DesignIssues/LinkedData.html>. Retrieved August 2, 2017.

Bittner, T., Donnelly, M., and Smith, B., 2004, 'Endurants and perdurants in directly depicting ontologies,' *AI Communications* 13(4): 247-258.

Bizer, C., Health, T., and Berners-Lee, T., 2009, 'Linked data: the story so far,' *International Journal of Semantic Web Information Systems* 5(3): 1-22.

Ceusters, W. and Smith, B., 2015, 'Aboutness: Towards foundations for the Information Artifact Ontology,' *Proceedings of the Sixth International Conference on Biomedical Ontology* (ICBO), Lisbon (CEUR 1515), 1-5.

Randell, D.A., Cui, Z., and Cohn, A.G., 1992, 'A spatial logic based on regions and connections,' *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, pp. 165-176.

Smith, B., 1995, 'On drawing lines on a map,' in A.U. Frank and W. Kuhn (eds.), *Spatial Information Theory: A Theoretical Basis for GIS* (Lecture Notes in Computer Science 988), Berlin/Heidelberg/New York: Springer, 475-484.

Smith, B., 1998, 'Basic concepts of formal ontology,' in N. Guarino (ed.), *Formal Ontology in Information Systems*, Amsterdam: IOS Press, 19-28.

Smith, B., 2001, 'Fiat objects,' *Topoi* 20(2): 131-148.

Smith, B., 2008, 'New desiderata for biomedical terminologies,' in K. Munn and B. Smith (eds.), *Applied Ontology: An Introduction*, Frankfurt/Lancaster: Ontos, 83-109.

Smith, B., 2012, 'On classifying material entities in Basic Formal Ontology,' *Interdisciplinary Ontology: Proceedings of the Third Interdisciplinary Ontology Meeting*, Tokyo: Keio University Press, 1-13.

Smith, B. and Ceusters, W., 2010, 'Ontological realism: A methodology for coordinated evolution of scientific ontologies,' *Applied Ontology* 5(3): 139-188.

Smith, B. and Grenon, P., 2004, 'The cornucopia of formal-ontological relations,' *Dialectica* 58(3): 279-296.

Smith, B., Malyuta, T., Rudnicki, R., Mandrick, W., Salmen, D., Morosoff, P., Duff, D.K., Schoening, J., and Parent, K., 2013, 'IAO-Intel: An Ontology of Information Artifacts in the Intelligence Domain,' *Proceedings of the Eighth International Conference on Semantic Technologies for Intelligence, Defense, and Security*, Fairfax, CEUR, 1097, 33-40.

Spear, A.D., Ceusters, W., and Smith, B., 2016, 'Functions in Basic Formal Ontology,' *Applied Ontology* 11(2): 103-128.