Modeling Information with 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

Table of Contents

Table of Contents ii
1 Introduction 1
2 Content, Bearers, and Values 2
2.1 Resolving Information Content Entities 4
2.2 Annotating without Provenance 4
3 Types of Information 5
3.1 Descriptions, Measurements, and Measurement Units 6
3.2 Names and Identifiers 7
3.3 Plans and Specifications 7
3.4 Using Sets 8
4 Modeling Examples 9
4.1 Address 9
4.2 Place and Date of Birth 9
4.3 Person's Organizational Role 10
4.4 Gain and Loss of Organizational Role 10
4.5 Employment Status 11
4.6 Weather Event with Lat-Long Coordinates 12
4.7 Max and Min Temperature 12
4.8 Event Start and End Times 13
4.9 Vehicle Speed 14
4.10 Car Model Specification 14
4.11 Designed vs. Actual Artifact Performance 15
4.12 Effect of Aircraft Maintenance 16
4.13 Financial Value of Vehicle 17
4.14 Sortie 17
4.15 Planned vs. Actual Operation 18
4.16 Planned Range of Durations 18
5 Conclusion 19

1 Introduction

Utilizing ontologies to achieve data interoperability hinges upon conformance to a common *semantics*, or way of expressing the content of that data. Specifically, utilizing the Common Core Ontologies (CCO) to integrate data sources requires knowledge of how the CCO structures different types of information.

The goal of this document is to help explain how to model various types of information in ways conformant to the semantics of the CCO. The intended audience of this document includes ontology users who need to map existing data sources to the CCO, query writers who want to extract information expressed using the CCO's semantics, and ontology developers needing to create domain ontologies that extend from the CCO.

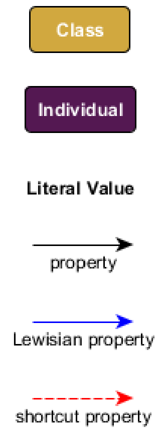
All readers of this document should also read the document "An Overview of the Common Core Ontologies" (v1.0), which explains the design of the CCO, its upper-level semantic framework, and the content of the individual ontologies that compose the CCO. Ontology developers should also read the document, "Best Practices of Ontology Development," as it describes best practices for creating ontologies which are conformant to CCO.

This document is structured as follows. Section 2 describes the CCO approach to representing the content and provenance of a data source and their relationship to the values that appear in that data source. Section 3 explains the main categories of information that can be represented in the CCO: descriptions, designators, and directives. This section also describes generally how to represent nominal groupings of entities (or sets) within CCO. Section 4 surveys over a dozen examples of common modeling patterns, which illustrate more concretely how CCO's semantics would be implemented to structure and integrate types of information.

Within this document, ontology classes will be expressed typographically in small caps (e.g., FUNCTION, OBJECT AGGREGATE) and ontology properties or relationships in bolded, italicized lowercase with individual words joined by underscores (***realizes***, ***has_part***). Thus:

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

Furthermore, the following symbols are used in the graph diagrams to represent classes, individuals, properties, and literal values:



In these diagrams the dashed red arrow links an individual to a literal value by skipping intermediary nodes of the following types: INFORMATION CONTENT ENTITY and INFORMATION BEARING ENTITY. Thus, the following two graphs are two different graphical representations of the same RDF triples:

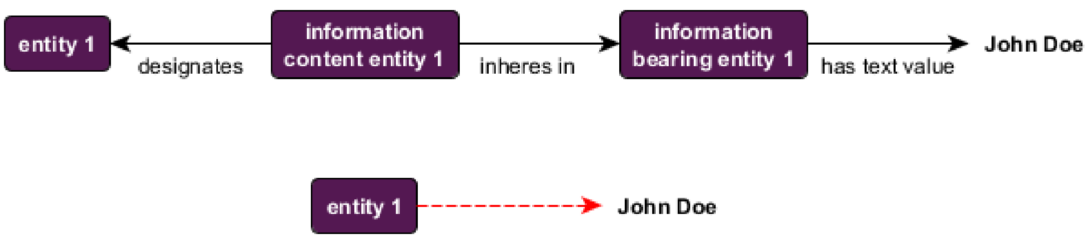


Figure 1: A graphical shortcut.

Note that this convention is used <u>only</u> to render some diagrams in this document more perspicuous and does <u>not</u> reflect the actual semantics of the CCO. The reader should note that this purely graphical convention is different from the use of annotation properties to link instances of INFORMATION CONTENT ENTITY to literal values (see the discussion in Section 2.2).

2 Content, Bearers, and Values

Because the CCO are designed as realism-based ontologies (see "An Overview of the Common Core Ontologies," Section 2.2), they draw a distinction between representations of *real* entities (e.g., John Doe, John Doe's weight), representations of *information* entities (e.g., John Doe's name, some measurement of John Doe's weight), and representations of entities that *bear* that information (e.g., a patient medical record, an employee database). These latter two kinds of representation are captured by the CCO classes: INFORMATION CONTENT ENTITY and INFORMATION BEARING ENTITY. It is the bearing entities which are attached to particular literal values (e.g., strings, integers) by means of data properties (e.g., *has_text_value*, *has_integer_value*). (For more information on the distinction between these two classes, see "An Overview of the Common Core Ontologies," Section 4.1.) Thus, all mappings from data sources to CCO-conformant ontologies should follow the same general pattern:

> ENTITY ***is_subject_of*** INFORMATION CONTENT ENTITY
> INFORMATION CONTENT ENTITY ***inheres_in*** INFORMATION BEARING ENTITY
> INFORMATION BEARING ENTITY ***has_text_value*** *Literal*

Note that the specific data property above (***has_text_value***) is merely an example of a data property linking an information bearer to a literal value (in this case, a text string). In fact, CCO defines several other data properties for linking information bearers to different kinds of literal values, including:

> ***has_integer_value***
> ***has_decimal_value***
> ***has_boolean_value***
> ***has_date_value***
> ***has_datetime_value***
> ***has_latitude_value***
> ***has_longitude_value***
> ***has_URI_value***

The trifold distinction between information content, information bearers, and the subject of that information allows CCO to accomplish two things. The first is to represent the fact that a single piece of information can be found in multiple locations (books, portable hard drives, databases, etc.). The second is to track the *provenance* of information: its origin, history, and quality. Thus, CCO provides the capability to represent the fact that a piece of information is attached to a particular database or has a particular confidence rating.

The CCO's semantics does **not** permit linking data values directly to the subjects of information content entities. Rather, all CCO data properties are [restricted by axioms] such that that they can link **only** instances of INFORMATION BEARING ENTITY to literal values. For example, the following graph would be **semantically incorrect**:

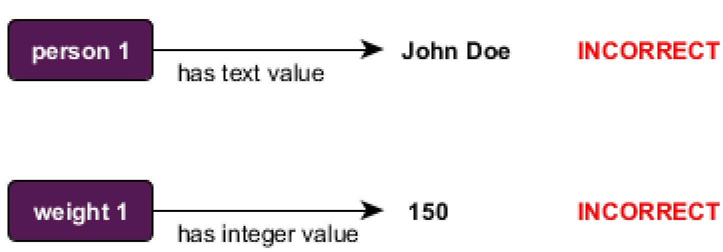


Figure 2: Incorrect way to link entities to data values.

2.1 Resolving Information Content Entities

A single instance of INFORMATION CONTENT ENTITY can inhere in multiple instances of INFORMATION BEARING ENTITY. For example, the content of a document can reside simultaneously in multiple copies of that document, and in various mediums. 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.). Typically, a person only has *one* name, but that name (content) can be found in *many* particular physical tokens.

This means—providing sufficient entity resolution—that many instances of INFORMATION CONTENT ENTITY would be resolved to a single resource with a single URI. For example, if John Doe's name appears in multiple databases, then a correctly resolved RDF graph that integrates those databases would contain exactly one node representing the information content of John Doe's name. That node would link to multiple nodes representing each unique physical bearer of John Doe's name.

This raises the question of under what conditions information content entities should be resolved to a single OWL/RDF individual. The answer is: *Whenever the content is identical, that content is a <u>single</u> instance of INFORMATION CONTENT ENTITY.* Thus, the origin of the content is irrelevant to whether the same content is being captured. For example, if two doctors separately determine that John Doe weighs 150lbs, then the content they record is identical, even though these events occur at two different times and places. By contrast, if one doctor measures the weight incorrectly, then different information content is generated.

2.2 Annotating without Provenance

In some situations, users may wish to link instances of INFORMATION CONTENT ENTITY to values without tracking provenance. To meet this need, the CCO introduces an annotation property, ***is_tokenized_by***, which links literal values directly to instances of INFORMATION CONTENT ENTITY.

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

The following figure represents these two ways of modeling the fact that a person has the name "John Doe". In the first, the provenance of the data value "John Doe" is tracked by means of an information bearing entity and the ***has_text_value*** relation. In the second, the provenance is ignored, and the information content entity is simply annotated with the value by means of the ***is_tokenized_by*** relation.

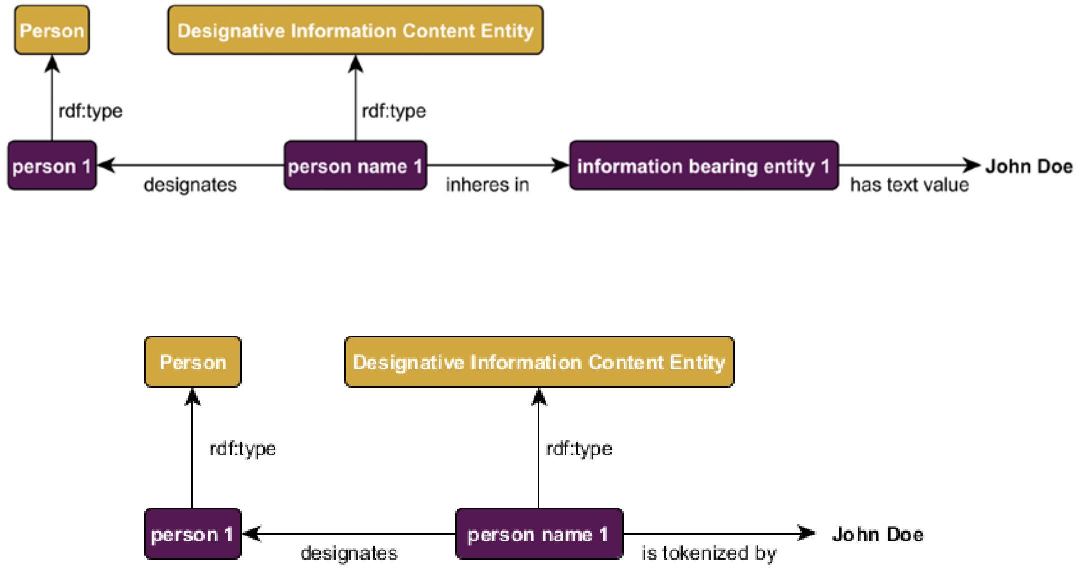


Figure 3: Tracking provenance vs. merely annotating.

3 Types of Information

An information content entity is linked to the entity that information is about by means of the *is_about* property (see "An Overview of the Common Core Ontologies," Section 4.1). This property has three subproperties, each of which represents a different way information can be about the subject of that information: ***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, DIRECTIVE INFORMATION CONTENT ENTITY and DESIGNATIVE INFORMATION CONTENT ENTITY.

3.1 Descriptions, Measurements, and Measurement Units

The class DESCRIPTIVE INFORMATION CONTENT ENTITY is used to represent information that is a measurement or representation of some portion of reality. More specific kinds of descriptive information are defined as subclasses, the most important of which are those that represent measurement information.

The class MEASUREMENT INFORMATION CONTENT ENTITY divides into four subclasses, each of which represents a different kind of measurement:

Class Name	Description	Example
RATIO MEASUREMENT INFORMATION CONTENT ENTITY	Quantitative measurements with a true-zero value.	Temperature in Kelvin.
INTERVAL MEASUREMENT INFORMATION CONTENT ENTITY	Quantitative measurements with no true-zero value.	Temperature in Fahrenheit, Celsius.
ORDINAL MEASUREMENT INFORMATION CONTENT ENTITY	Measurements of entities into rank orders.	Max and min temperature for a given day.
NOMINAL MEASUREMENT INFORMATION CONTENT ENTITY	Groupings of entities according to some shared (possibly arbitrary) characteristic.	Temperature categorized as "hot" and "cold".

Strictly speaking, every instance of MEASUREMENT INFORMATION CONTENT ENTITY is neutral with respect to measurement units. The specific subclasses of MEASUREMENT INFORMATION CONTENT ENTITY mentioned above capture only the *scale* that is used (e.g., ratio, interval), but not the unit of measurement (e.g., Kelvin, Fahrenheit). For example, measurements in terms of Fahrenheit and Celsius use the same scale (interval), but different measurement units. By contrast, measurements in terms of Kelvin use not only a different measurement unit, but also a different measurement scale, and so a different type of MEASUREMENT INFORMATION CONTENT ENTITY.

The CCO class MEASUREMENT UNIT and its subclasses are used to represent various general types of measurement units. However, note that specific measurement units (Fahrenheit, inch, kilometers per hour) are represented in CCO as *individuals*, not *classes*. Thus, examples of subclasses of MEASUREMENT UNIT include MEASUREMENT UNIT OF TEMPERATURE, MEASUREMENT UNIT OF LENGTH, and MEASUREMENT UNIT OF SPEED, whereas instances of these classes would be Fahrenheit, inch, and kilometers per hour, respectively.

Instances of the class MEASUREMENT UNIT are linked only to instances of INFORMATION BEARING ENTITY, <u>not</u> INFORMATION CONTENT ENTITY. Thus, a measurement of some temperature in Fahrenheit would be represented in a graph like the following:

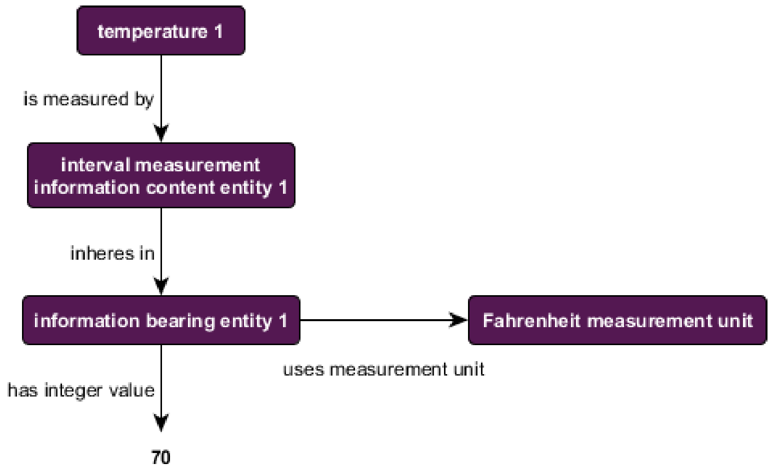


Figure 4: Relationship between measurement information and measurement unit.

Note also that all literal values for temperatures measured according to the Fahrenheit scale should use the same instance of Fahrenheit measurement unit.

3.2 Names and Identifiers

The class DESIGNATIVE INFORMATION CONTENT ENTITY is used to represent information that names or identifies some entity. Its two most prominent subclasses are DESIGNATIVE NAME, which is used for natural language names (John Doe, the United States), and CODE IDENTIFIER, which is used for designators that are encoded for special purposes (social security numbers, country codes). Other subclasses include BARCODE, ARBITRARY IDENTIFIER, and SITE IDENTIFIER.

3.3 Plans and Specifications

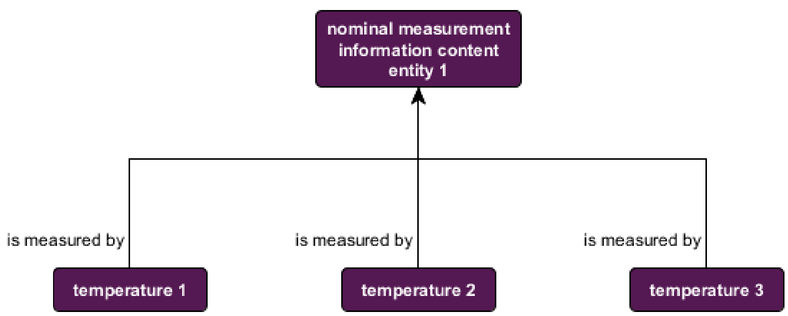
The class DIRECTIVE INFORMATION CONTENT ENTITY is used to represent information that prescribes or specifies how some entity or portion of reality *should* be. Its two most important subclasses are PLAN and ARTIFACT MODEL. The former is used for information which prescribes how some process should unfold, both the objective and the course of action to be followed to achieve that objective. The class OBJECTIVE is closely related to PLAN, in that every instance of PLAN *has_part* some instance of OBJECTIVE. The difference is that whereas the objective prescribes the end goal, the plan prescribes some means of achieving that goal. ARTIFACT MODEL is a class for representing information about how an artifact should be constructed or should function. Other subclasses DIRECTIVE INFORMATION CONTENT ENTITY include LANGUAGE, a set of information which prescribes accepted ways of communicating, and ALGORITHM, which prescribes inputs and outputs of mathematical functions (as implemented, for example, in computer programming).

Note that an instance of DIRECTIVE INFORMATION CONTENT ENTITY might be related to other entities by two different *prescribes* relations: the ordinary CCO *prescribes* and the Lewisian *prescribes* relation. In the former case, the relation would link a plan to, for example, some actual military action that was following the plan (or trying to). In the latter case, the relation would like that same plan to the military action as described exactly in the plan. (See "An Overview of the Common Core Ontologies," Section 4.11.)

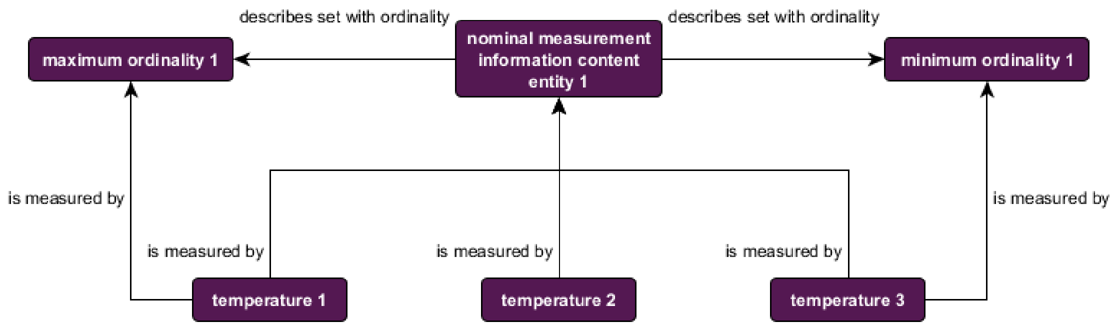
3.4 Using Sets

The class NOMINAL MEASUREMENT INFORMATION CONTENT ENTITY can be used to represent what mathematicians traditionally call a set (or more precisely, the *intension* of a set). The basic idea is that it is sometimes advantageous to be able to represent arbitrary groupings of entities within one's ontology-aligned data. For example, someone might wish to group all temperatures recorded in a given year and identify one temperature as the highest temperature recorded relative to that set. The temperatures do not constitute an ontology class (the grouping here is not a bona fide type of entity). Rather, the temperatures belong to a purely convenient, possibly arbitrary, way of grouping individuals into a set.

To represent a scenario like this, the CCO uses the class NOMINAL INFORMATION CONTENT ENTITY. An instance of that class can be treated as the naming of a set (its intension), which then stands in *is_about* relationships (specifically, by means of the *is_measured_by* subproperty) to the individuals in the set (its extension). For example, one could represent three temperatures belonging to a set as follows:



It then becomes possible, for example, to identify the maximum and minimum members of a particular set. Specifically, CCO introduces subclasses of ORDINAL MEASUREMENT INFORMATION CONTENT ENTITY, namely, MAXIMUM ORDINAL MEASUREMENT INFORMATION and MINIMUM ORDINAL MEASUREMENT INFORMATION, along with a property to link sets to ordinalities, namely, *measures_set_with_ordinality*. Thus, one could represent the maximum and minimum temperatures in the same set as follows:

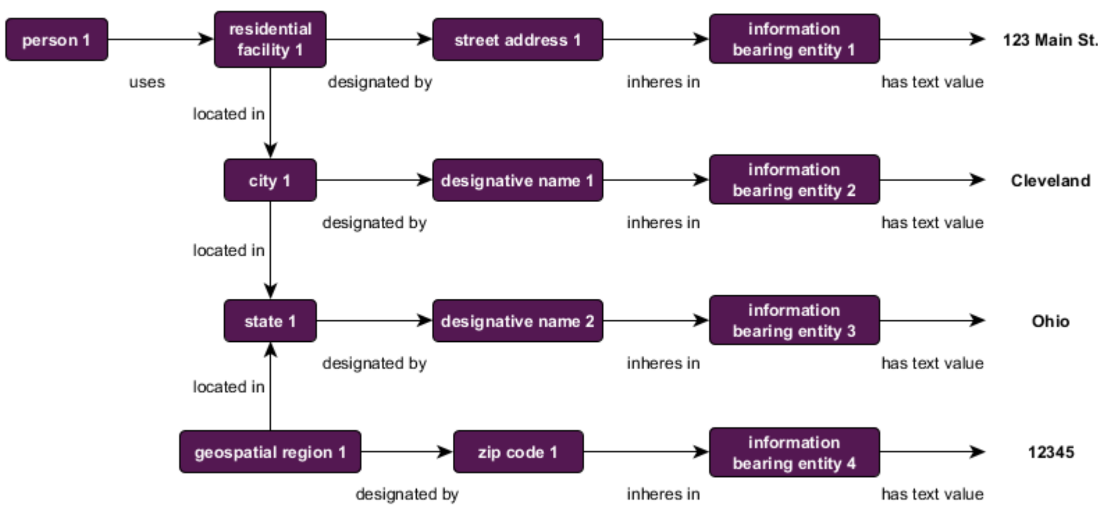


4 Modeling Examples

This section contains a number of examples which are meant to illustrate how the various pieces describes in “An Overview of the Common Core Ontologies” can be fitted together into a robust semantic representation. The examples were selected are intended to illustrate common mapping patterns

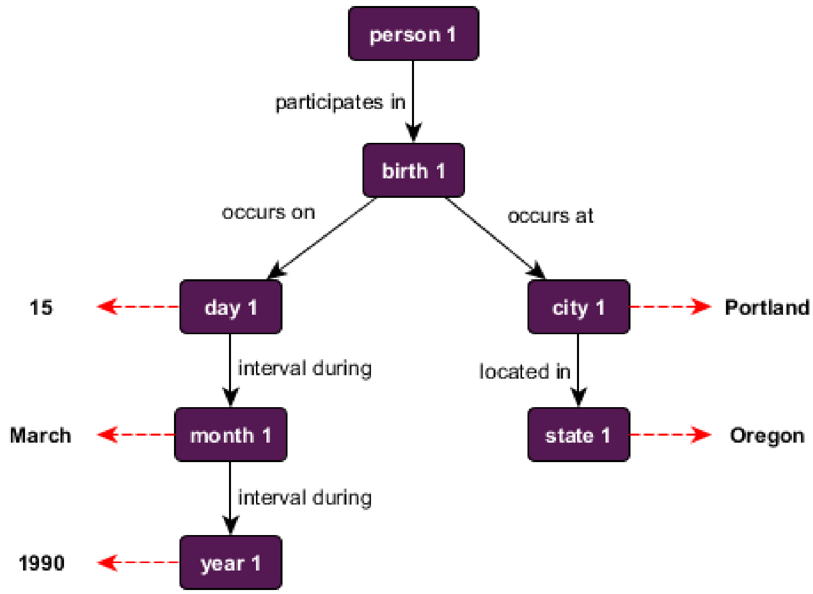
4.1 Address

A person’s address. The basic pattern is that separate data table entries are mapped to different identifiers, which stand in different geospatial relations to one another (*located_in*).



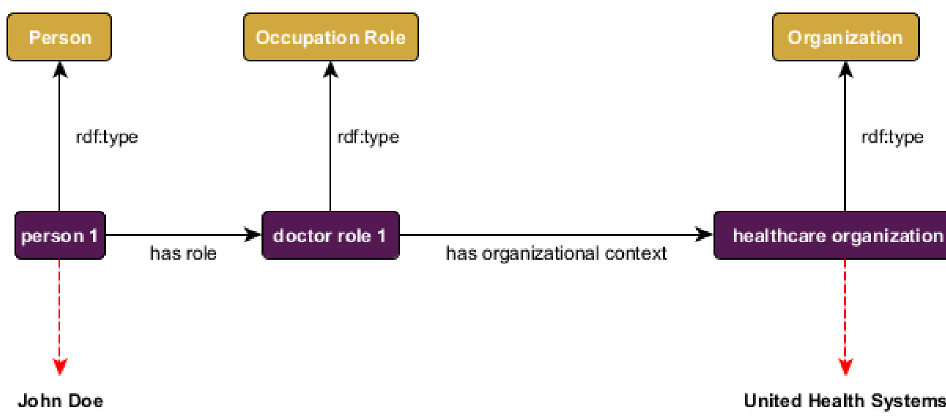
4.2 Place and Date of Birth

The date and place of a person’s birth. The birth is an event, which is associated with a particular time and location, and in which the person participates. Other data could be linked to that event node, e.g., any healthcare professionals involved in the birth process.



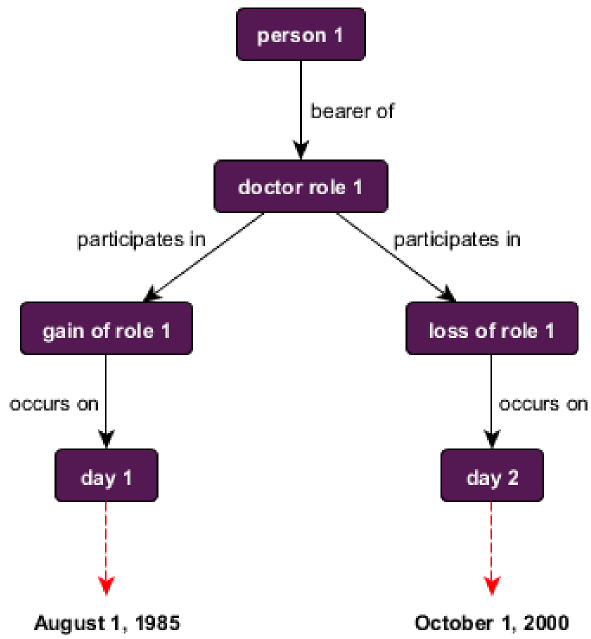
4.3 Person's Organizational Role

A person named John Doe has a role of doctor within a healthcare organization. The person is distinct from his role, and role exists within the context of a particular organization.



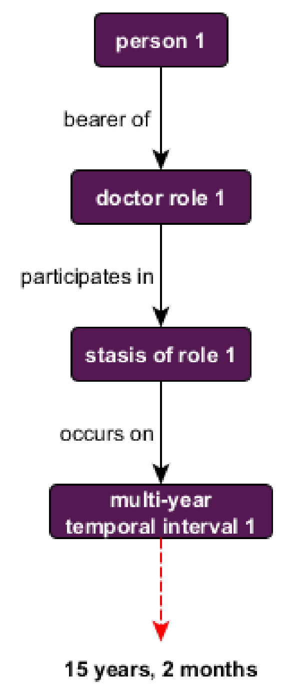
4.4 Gain and Loss of Organizational Role

John Doe becomes employed as a doctor on August 1, 1985 and leaves that position on October 1, 2000. The gain and loss are change processes, which are associated with different temporal intervals.



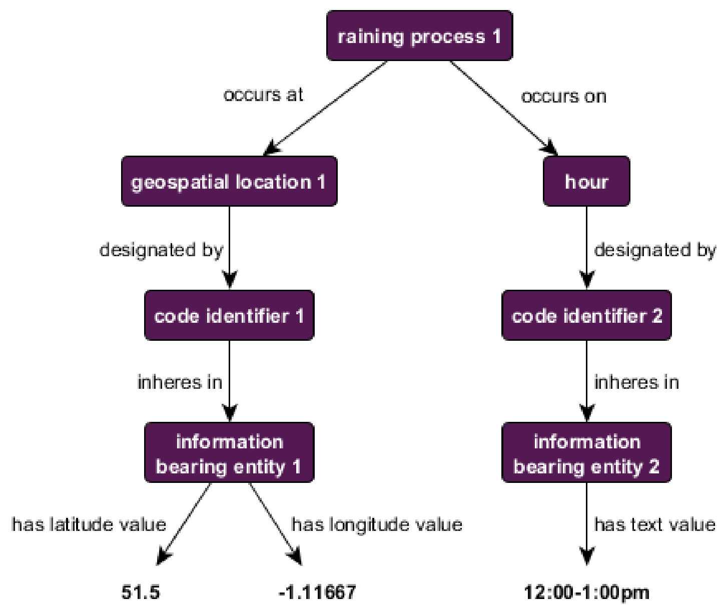
4.5 Employment Status

John Doe was employed as a doctor for 15 years. This employment status is captured by the CCO class STASIS, this instance of which has a duration of just over 15 years.



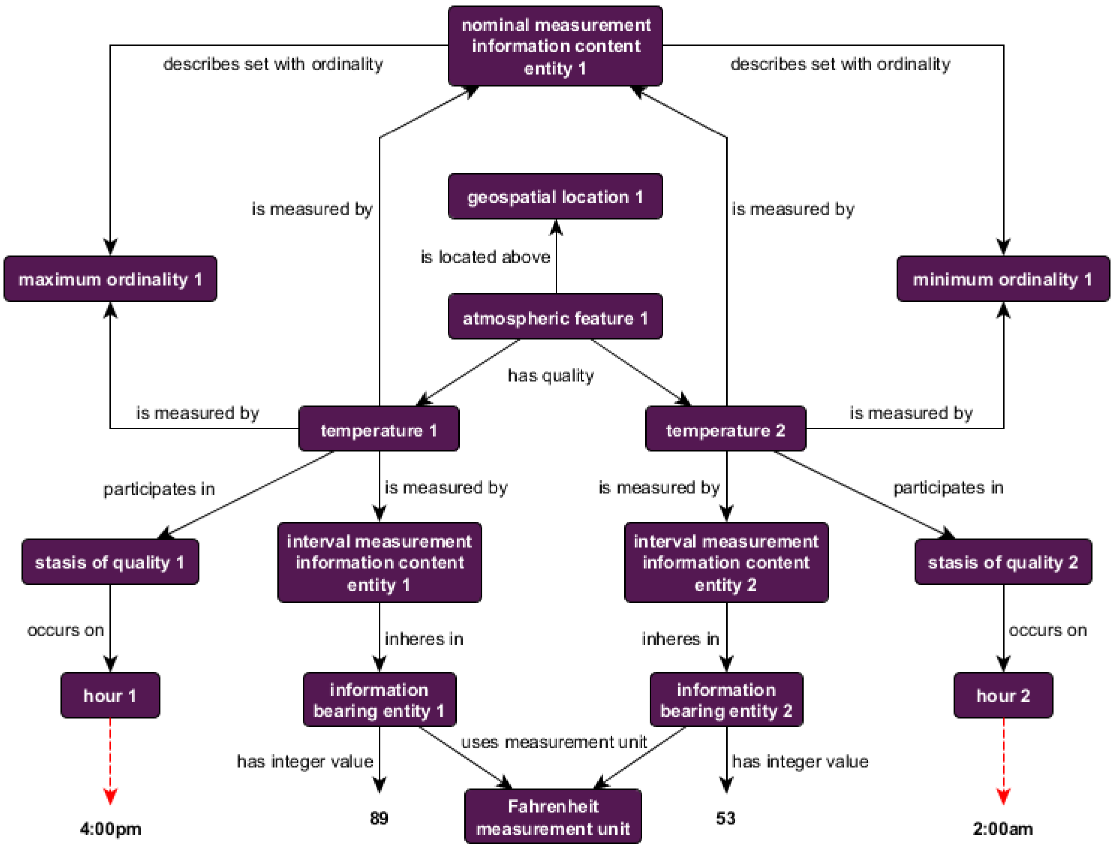
4.6 Weather Event with Lat-Long Coordinates

Rain occurs during an hour-long period at a location identified by certain latitude-longitude coordinates. This example illustrates again the relationship of an event to places and times, and specifically of identifying locations via latitude and longitude coordinates.



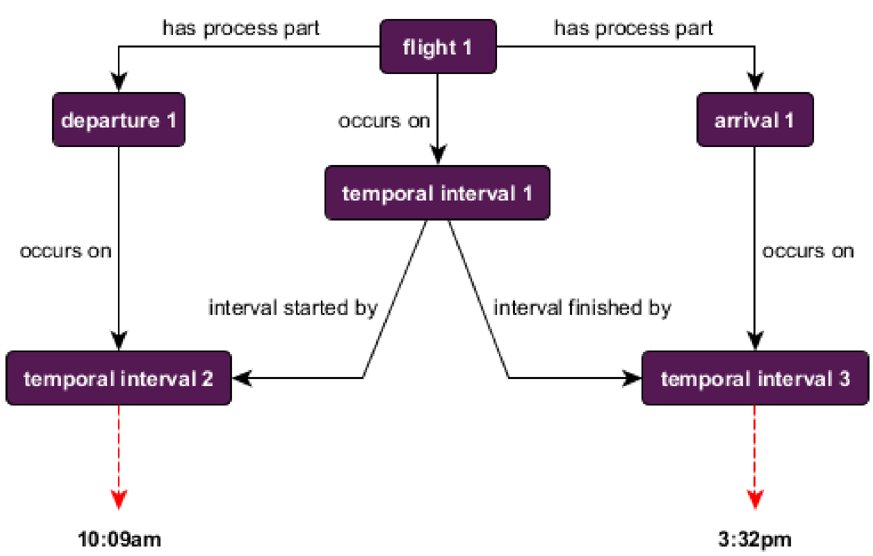
4.7 Max and Min Temperature

The maximum and minimum temperatures for a given day. This example illustrates the usage of sets along with maximum and minimum ordinality (see Section 3.4). It also illustrates how measurement units get attached to measurement information.



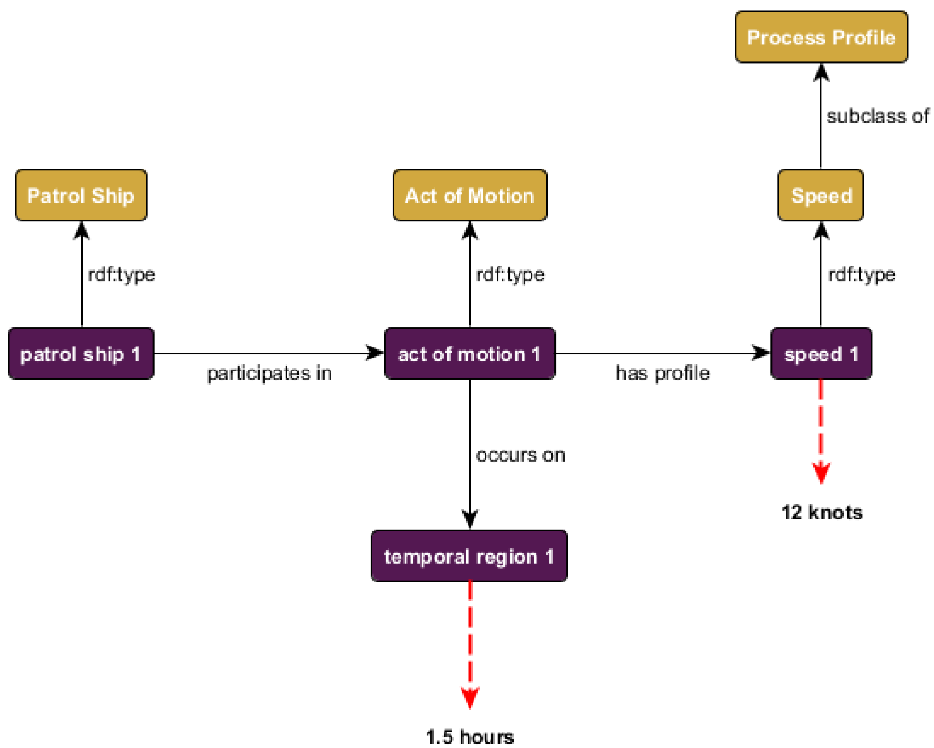
4.8 Event Start and End Times

A flight departs at 10:09am and arrives at 3:32pm. This example illustrates how the full duration of the flight (temporal interval 1) is related to its starting (temporal interval 2) and end times (temporal interval 3).



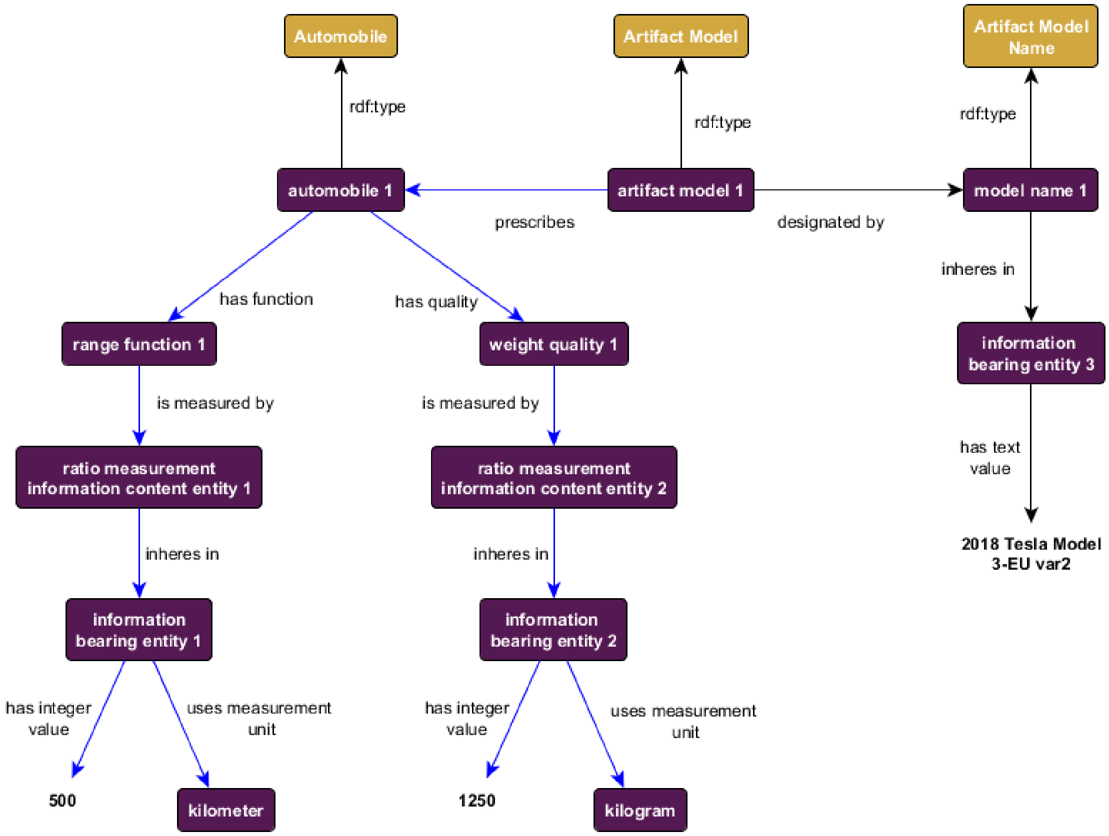
4.9 Vehicle Speed

A patrol ship moves at a speed of 12 knots over 1.5 hours. This illustrates how process profiles are linked to processes. (See "An Overview of the Common Core Ontologies," Section 3.2).



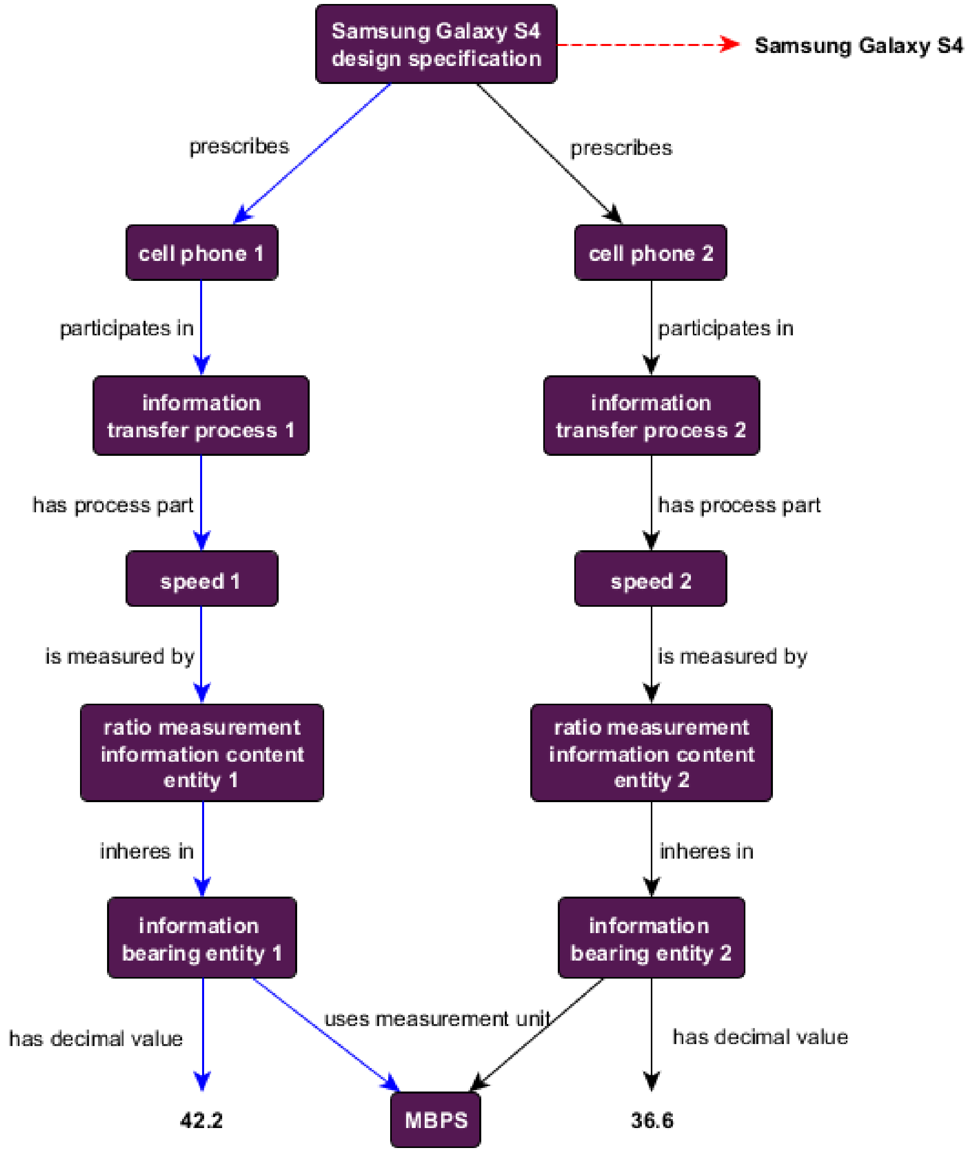
4.10 Car Model Specification

The range and weight specification for a 2018 Tesla Model 3-EU var2. This graph illustrates how to represent the content of an artifact specification. The node "automobile 1" does not represent an actual Tesla Model 3, but the "ideal" or "model" Tesla Model 3 as detailed in the specification. For this reason, the links on the lefthand side of the graph are all from the Lewisian Relation Ontology. (See "An Overview of the Common Core Ontologies," Section 4.11.)



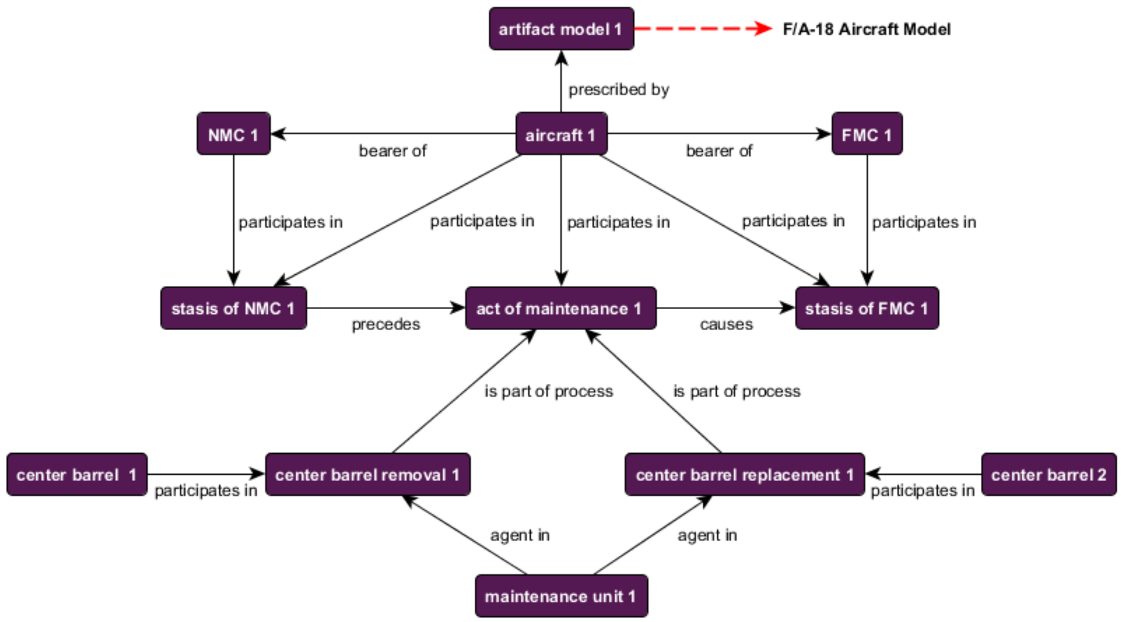
4.11 Designed vs. Actual Artifact Performance

A Samsung Galaxy S4 fails to meet its designed data transfer rate of 42.2MBPS. This example illustrates how an artifact's actual performance can diverge from the performance it *should* have, as detailed in the artifact model.



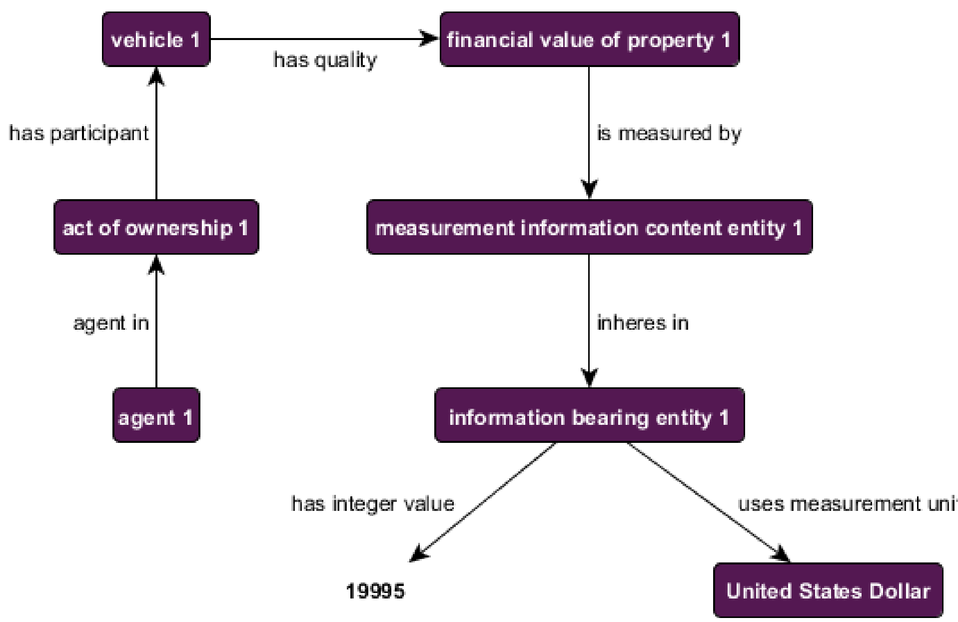
4.12 Effect of Aircraft Maintenance

A Non-Mission Capable F/A-18 aircraft undergoes maintenance (replacement of center barrel) and is thus restored to Fully Mission Capable status. This graph illustrates how various processes can be related causally, such that they change the status (STASIS) of some entity.



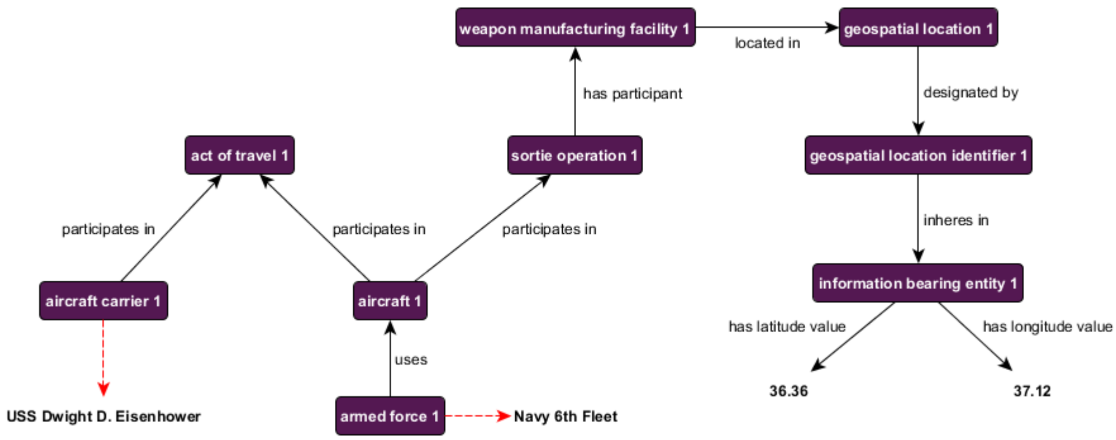
4.13 Financial Value of Vehicle

A vehicle is valued at $19,995. This graph illustrates how financial value is attached to artifacts.



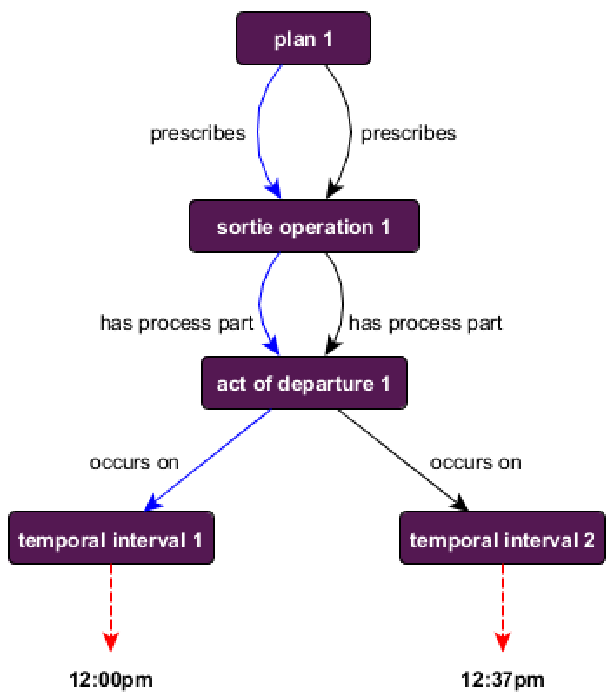
4.14 Sortie

A Navy 6th Fleet aircraft takes off from the U.S.S. Eisenhower to take place in an operation over a weapon manufacturing facility located at certain latitude-longitude coordinates. This example illustrates how to model a planned operation.



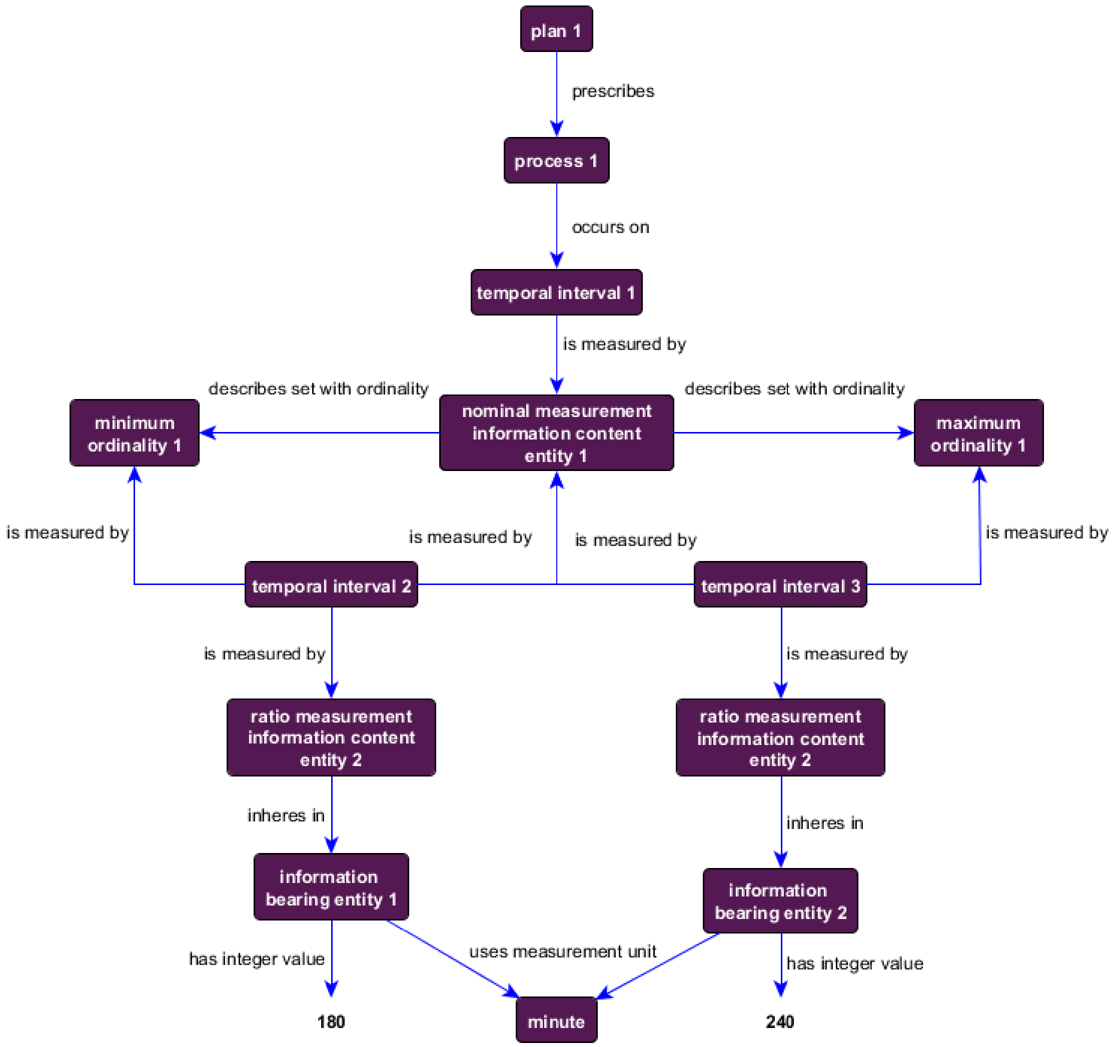
4.15 Planned vs. Actual Operation

A planned sortie was supposed to begin at 12:00pm, but instead began at 12:37pm. This example illustrates how the initial planned operation can diverge in some respect from the actual execution of the plan. Note that, in some respects, the actual execution conforms to the initial plan, but in other respects diverges.



4.16 Planned Range of Durations

A plan prescribes that a process should have a duration of between 180 and 240 minutes. This graph presents a more complex example, one which involves a range of outcomes prescribed by a plan.



5 Conclusion

Conformity to a common semantic model promotes greater data interoperability. This document provided some background information regarding how to represent different types of information within the context of the Common Core Ontologies. It also presented a number of examples, which aim to illustrate how concrete cases would be modeled within CCO.