Time Ontology Extended for Non-Gregorian Calendar Applications

Editor: Mark Gahegan, University of Auckland, New Zealand
Solicited review(s): Karl Grossner, Stanford University, USA; Brandon Whitehead, University of Auckland, New Zealand; one anonymous reviewer

Simon J D Cox
CSIRO Land and Water, PO Box 56, Highett, Vic. 3190, Australia
simon.cox@csiro.au

Abstract.

We have extended OWL-Time to support the encoding of temporal position in a range of reference systems, in addition to the Gregorian calendar and conventional clock. Two alternative implementations are provided: as a pure extension or OWL-Time, or as a replacement, both of which preserve the same representation for the cases originally supported by OWL-Time. The combination of the generalized temporal position encoding and the temporal interval topology from OWL-Time support a range of applications in a variety of cultural and technical settings. These are illustrated with examples involving non-Gregorian calendars, Unix-time, and geologic time using both chronometric and stratigraphic timescales.

Keywords: temporal reference system, calendar, temporal topology, ontology re-use

1. Introduction

OWL-Time [6] provides a lightweight model for the formalization of temporal objects, based on Allen's temporal interval calculus [1]. Developed primarily to support web applications, date-time positions are expressed using the familiar Gregorian calendar and conventional clock.

Many other calendars and temporal reference systems are used in particular cultural and scholarly contexts. For example, the Julian calendar was used throughout Europe until the 16th century, and is still used for computing key dates in some orthodox Christian communities. Lunisolar (e.g. Hebrew) and lunar (e.g. Islamic) calendars are still used, and many similar have been used historically. Dynastic calendars (counting years within eras defined by the reign of a monarch or dynasty) were used earlier in many cultures. In more contemporary applications, Loran-C, Unix and GPS time are based on seconds counted from a specified origin (in 1958, 1970 and 1980, respectively). Archaeological and geological applications use chronometric scales based on years counted backwards from 'the present' (defined as 1950 for radiocarbon dating [15]), or named periods associated with specified correlation markers [4,5,9].

Since OWL-Time only allows for Gregorian dates and times, applications that require other reference systems must look elsewhere. However, the basic structures provided by OWL-Time are not specific to a particular reference system, so it would be preferable to use them in the context of a more generic solution. Some previous extensions to OWL-Time have been proposed, in particular to deal with aggregates and recurrent intervals [12]. Perrin et al. [13] use the OWL-Time temporal relations in their geologic timescale ontology, but introduce a new class and property for geochronologic instants.

In this paper we describe extensions to OWL-Time, which support

- the use of an explicit temporal reference system when specifying temporal position,

- general encodings for dates and times not based on the Gregorian calendar and clocks.

We provide two implementations: “Time-plus” is a pure extension to OWL-Time, and “Time-new” is a generalized replacement[1], which preserves the core of OWL-Time within a modified class hierarchy.

We illustrate the application of the ontology with a number of examples. While we do not provide a model for describing temporal reference systems, we show how the extended ontology supports the description of ordinal temporal reference systems also extending into deep time.

2. Time-plus - extension to OWL-Time

2.1. Temporal reference system

A useful classification of temporal reference systems is provided in ISO 19108 [8]. Four kinds of system are distinguished:

- Coordinate systems, in which a (temporal) position is expressed as a signed quantity, offset from a specified origin
- Ordinal reference systems, based on ordered named intervals
- Calendars, in which position is expressed using a year-month-day structure
- Clocks, in which position within a day is expressed in hours-minutes-seconds.

The temporal reference system used in OWL-Time is not explicitly specified [6], but can be inferred to be the Gregorian calendar and conventional clock as this is specified for the `xsd:dateTime`, `xsd:gYear`, `xsd:gMonth` and `xsd:gDay` datatypes used in encoding values [3,7].

In Time-plus we introduce an additional property, denoted `tplus:hasTRS`, to support explicit indication of a temporal reference system. Its range is the class of temporal reference systems, denoted `tplus:TRS`. As defined in this ontology, this class is a stub that is only a placeholder for the top of a hierarchy of temporal reference system types, so no properties are defined with the domain `tplus:TRS`, and there are no local restrictions.

2.2. Temporal position

Two encodings for temporal position are provided in OWL-Time. Using the datatype `xsd:dateTime`, position is encoded as a structured literal, with a precision of seconds or finer indicated by the precision of the seconds field in the value. Using the class `time:DateTimeDescription` temporal position is given as a value structured using separate properties, with precision in the range from seconds to years, indicated by the value of the `time:unitType` property[2]. The properties `time:year`, `time:month` and `time:day` have range `xsd:gYear`, `xsd:gMonth` and `xsd:gDay` respectively, which are tied to the Gregorian calendar [3,14]. Thus, OWL-Time is only capable of expressing temporal position in terms of the conventional 12-month calendar with January being the first month of the year, and using the 24-hour clock within each day. Furthermore, strictly speaking, the Gregorian calendar is valid only for time positions in the ‘Common Era’ (i.e. commencing with the year 1 C.E.).

In order to support a wider range of temporal position descriptions, in Time-plus we introduce two classes - `tplus:GenDateTimeDescription` and `tplus:TimePosition`. The former is a generalization of `time:DateTimeDescription`, with the range of `tplus:year`, `tplus:month` and `tplus:day` properties adapted for use with non-Gregorian calendars. In `tplus:TimePosition`, two properties provide alternative encodings for the value: `tplus:numericValue` for position described using a number, and `tplus:nominalValue` for position indicated as a name. The property `tplus:hasTRS` (described above) is mandatory on individuals from both classes, to indicate the reference system. We also introduce properties `tplus:inDateTime` and `tplus:inTimePosition` as alternatives to `time:inDateTime` and `time:inXSDDateTime` provided in OWL-Time. The new classes and properties are summarized in Table 1, with relationships to OWL-Time shown in Figure 1.

These extensions allow temporal position to be specified using all the temporal reference system types mentioned in 2.1:

- A temporal coordinate using a `tplus:TimePosition` with a `tplus:numericValue`
- An ordinal position using a `tplus:TimePosition` with a `tplus:nominalValue`

---

[1] The Time-plus ontology is published at http://def.seegrid.csiro.au/ontology/time/plus and Time-new at http://def.seegrid.csiro.au/ontology/time/new. Elements of the ontologies are denoted in this paper with the namespace prefixes `tplus`: and `tnew`:, respectively.

[2] Resources from OWL-Time are denoted in this paper with the prefix “time”.

- A position in the Gregorian calendar using `time:DateTimeDescription` with `time:unitType` set to `time:unitDay`
- A position in an arbitrary calendar using `tplus:GenDateTimeDescription` with `time:unitType` set to `time:unitDay`
- A position in a calendar/clock system using either
  - `time:DateTimeDescription` or `tplus:GenDateTimeDescription` with `time:unitType` set to `time:unitHour`, `time:unitMinute` or `time:unitSecond`,
  - `xsd:dateTime` with the default (Gregorian) reference system.

The range of `tplus:nominalValue` in the context of the generalized `tplus:TimePosition` is a string, since the requirement is to support designation of position in any ordinal reference system, the full set of which is not known (e.g. 'beginning of Late Cambrian', 'end of the bronze age', 'end of Ming dynasty', 'start of the reign of Pope Leo IX').

The range of each generalized date property is defined using the datatype restriction capability of OWL2 [11] so that it has the same *lexical* form as the original property used in OWL-Time, extended as necessary. For example, the definition of `tplus:genMonth` is shown in Listing 1, which allows month values up to 20 in order to accommodate calendars such as the Hebrew lunisolar calendar (which has a leap-month in some years), the Baha'i calendar (19 months in a year) and the Mayan calendars (18 and 20 periods per year). A note in section 3.3.14 of the XML Schema – Datatypes specification [14] draws attention to the difficulty of converting month values between calendars, and warns against use of gMonth for non-Gregorian calendars.

Note, however, that a generic OWL processor will not support value-based reasoning over purely lexical representations.

Listing 1 – definition of tplus:genMonth as a string pattern (Turtle syntax [2])

```
tplus:genMonth a      rdfs:Datatype ;
  rdfs:label          "Generalized month"^^xsd:string ;
  owl:onDataType      xsd:string ;
  owl:withRestrictions  ( [ xsd:pattern
"--(0[1-9]|1[0-9]|20)(Z|(\\+|-)((0[0-9]|1[0-3]):[0-
5][0-9]|14:00))?"^^xsd:string ] ) .
```

Table 1 – Elements of the ontology

<table>
<tr><td>Classes</td><td colspan="2"><code>tplus:TRS</code>
<code>tplus:GenDateTimeDescription</code>
<code>tplus:TimePosition</code></td></tr>
<tr><td rowspan="5">Properties</td><td><code>tplus:hasTRS</code></td><td>domain <code>tplus:TimePosition|tplus:GenDateTimeDescription</code>
range <code>tplus:TRS</code></td></tr>
<tr><td><code>tplus:inDateTime</code></td><td>domain <code>time:Instant</code>
range <code>tplus:GenDateTimeDescription</code></td></tr>
<tr><td><code>tplus:day</code>
<code>tplus:month</code>
<code>tplus:year</code></td><td>domain <code>tplus:GenDateTimeDescription</code>
range <code>tplus:genDay</code>
range <code>tplus:genMonth</code>
range <code>tplus:genYear</code></td></tr>
<tr><td><code>tplus:inTimePosition</code></td><td>range <code>tplus:TimePosition</code></td></tr>
<tr><td><code>tplus:numericValue</code>
<code>tplus:nominalValue</code></td><td>domain <code>tplus:TimePosition</code>
range <code>tplus:Number</code>
range <code>xsd:string</code></td></tr>
<tr><td rowspan="2">Datatypes</td><td><code>tplus:genDay</code>
<code>tplus:genMonth</code>
<code>tplus:genYear</code></td><td>String with pattern constraint matching lexical representation of <code>xsd:gDay|xsd:gMonth|xsd:gYear</code></td></tr>
<tr><td><code>tplus:Number</code></td><td>set of numbers</td></tr>
</table>

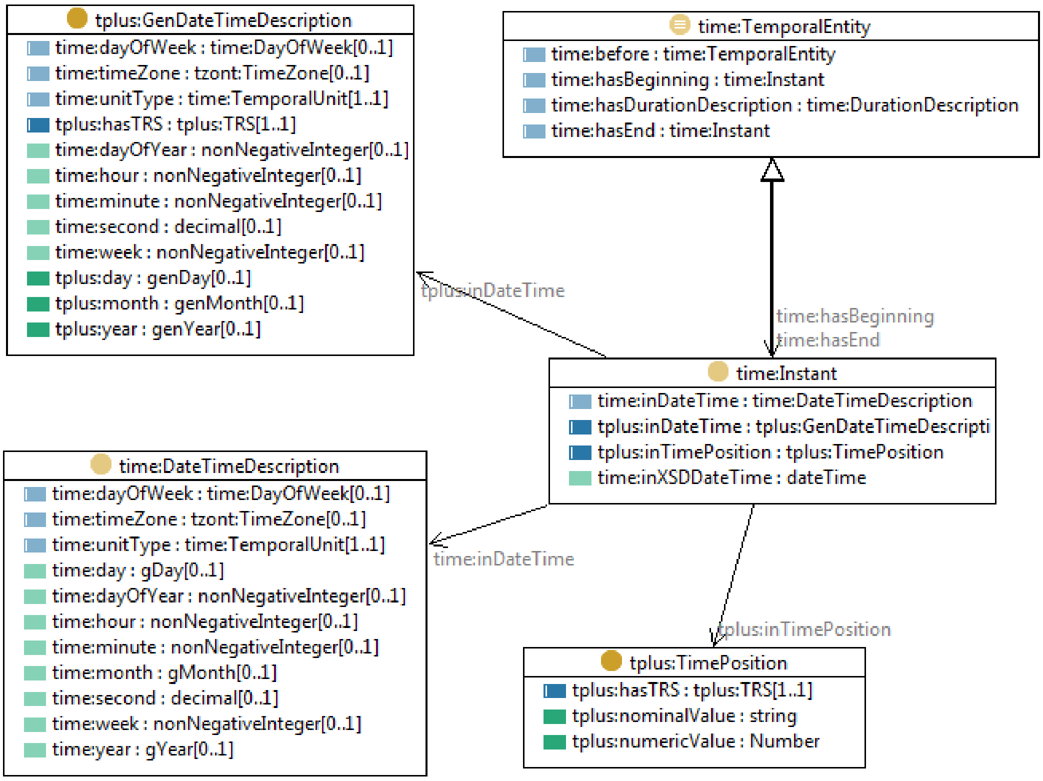

Figure 1 – Extended time ontology for supporting additional temporal position encodings.
Resources from OWL-Time use the namespace prefix "time:". (UML-style representation of classes and properties from TopBraid.)

3. Time-new - generalization of OWL-Time

The Time-plus ontology described above extends OWL-Time non-intrusively. However, while the class `tplus:TimePosition` adds distinct new functionality, the class `tplus:GenDateTimeDescription` merely clones `time:DateTimeDescription`, with the ranges of the three date properties generalized, plus a mandatory `tplus:hasTRS` property. A more natural implementation would recognize that the membership of `time:DateTimeDescription` is a subclass of `tplus:GenDateTimeDescription`.

We have used the latter approach in Time-new, which is summarized in Figure 2. The class `tnew:DateTimeDescription` is sub-classed from the generalized version `tnew:GenDateTimeDescription`, using local OWL constraints to reproduce the original semantics as shown in Listing 2.

Listing 2 – Local constraints on values of day, month and year properties in the context of a Gregorian date-time description

```
tnew:DateTimeDescription
  a    owl:Class ;
  rdfs:subClassOf  tnew:GenDateTimeDescription ;
  rdfs:subClassOf  [ a      owl:Restriction ;
                     owl:allValuesFrom  xsd:gDay ;
                     owl:onProperty     tnew:day
                   ] ;
  rdfs:subClassOf  [ a      owl:Restriction ;
                     owl:allValuesFrom  xsd:gMonth ;
                     owl:onProperty     tnew:month
                   ] ;
  rdfs:subClassOf  [ a      owl:Restriction ;
                     owl:allValuesFrom  xsd:gYear ;
                     owl:onProperty     tnew:year
                   ] ;
  rdfs:subClassOf  [ a         owl:Restriction ;
                     owl:hasValue
<http://dbpedia.org/resource/Gregorian_calendar> ;
                     owl:onProperty  tnew:hasTRS
                   ] ;
.
```

While this appears to reflect the class semantics better, the mechanics of OWL datatype definition and derivation introduce complications.

The datatypes `xsd:gYear`, `xsd:gMonth` and `xsd:gDay`, used in the original OWL-Time and in the constraints on the corresponding class in Time-new, are specified in the Gregorian calendar [14] (else there would be no need for the generalized class). However, the date-fragment datatypes are 'atomic types' [14], which cannot be derived from a more generalized datatype by restriction on facets. Thus, while `tnew:year`, `tnew:month` and `tnew:day` are defined as datatype properties, their ranges are left unspecified, and so not even the lexical form is constrained when used in the context of individuals from the class `tnew:GenDateTimeDescription`. This points towards a pattern where a subclass, sibling to `tnew:DateTimeDescription`, should be defined for *each* calendar of interest, with suitable constraints on at least the lexical space of the day, month and year properties, following the pattern of Listings 1 and 2. Note that `tnew:DateTimeDescription` is the only specialization built-in to Time-new, because it is required for the most common (Gregorian) applications, and it only leverages the standard XML Schema datatypes.

Overshadowing all this, however, is the fact that even the XML Schema date-fragment datatypes are not built in to OWL [11], so are not supported by generic OWL reasoning applications. The "temporal reasoner" referred to in the OWL-Time specification [6] might be expected to deal with this, and "Time-new reasoners" could in turn map each additional lexical form to a rigorous value-space in order to support value-based processing. Note that the definition of the date-fragment datatypes is a particularly complex area in XML Schema, with them strictly being understood as elements in the "seven-property model" of a calendar/clock system. It is notable that the values are *not* integers: the lexical representation of `xsd:gDay` has a prefix "---", `xsd:gMonth` a leading "--", and values in `xsd:gYear` must include at least four digits and there are complications around the interpretation of values below "0001" since there is no year zero in the conventional calendar [14].

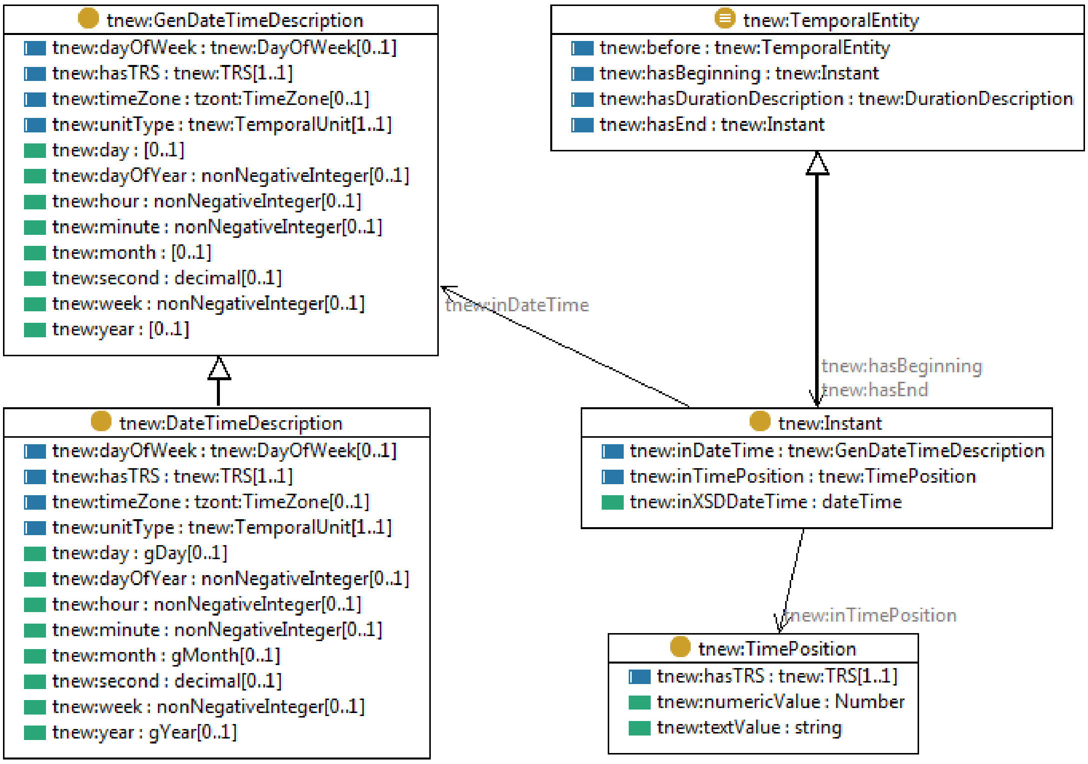

Figure 2 – Generalized time ontology for supporting additional temporal position encodings. In the context of tnew:DateTimeDescription, tnew:hasTRS is constrained to take the value <http://dbpedia.org/resource/Gregorian_calendar>

4. OWL profile and conformance

Time-plus and Time-new inherit the limitations of OWL-Time [6] concerning OWL 2 conformance:

- restrictions on cardinalities of properties in the time:DateTimeDescription class are not expressible using the OWL 2 EL, RL or QL profiles [10];
- the date-fragment datatypes used in time:DateTimeDescription are not built-in to OWL, so may not be recognized by a generic OWL processor.

Time-plus and Time-new meet the requirements for the OWL 2 DL profile, except that some vocabularies used only for metadata annotations on the ontology are not formal OWL ontologies (e.g. Dublin Core and its imports).

5. Examples

Listing 3 provides an example of a time instant with position formalized in four different ways, using Time-plus:

- `time:inXSDDateTime` shows the date and time encoded in compact form using the standard datatype derived from ISO 8601
- `my:AbbyBirthdayGregorian` encodes the date and time according to the Gregorian calendar using separate properties for its elements
- `my:AbbyBirthdayHebrew` encodes the date and time according to the Hebrew calendar using separate properties for its elements
- `my:AbbyBirthdayUnix` encodes the birthday and time as a temporal coordinate, specifically as the integer number of seconds since the epoch 1970-01-01T00:00:00.00Z.

Listing 4 provides two examples of events in geologic time. The time of formation of the earth is given as a coordinate in the reference system that counts years backwards from the present. The time of the end of the dinosaurs is given as a named position within the international chronostratigraphic chart [5].

Listing 5 shows how the extended ontology supports the description of elements in an ordinal temporal reference system. The Cambrian Period is modeled as a `time:ProperInterval`. The beginning and end are modeled as `time:Instants`, whose positions are given as temporal coordinates in millions of years counted backwards, using the new `tplus:TimePosition` class. Topological relations with other Eons, Periods and Epochs are described using the properties provided in OWL-Time (non-exhaustively).

Using the Time-new ontology, these examples will be structurally identical, but with the prefixes `time:` and `tplus:` both replaced by `tnew:`, and `tplus:genYear`, `tplus:genMonth` and `tplus:genDay` in Listing 3 replaced by suitable datatypes defined for the Hebrew calendar.

Note that in both Time-plus and Time-new, the cases supported by OWL-Time are represented identically to the original. This was a key design goal of the new ontologies.

Listing 3 – A single event with temporal position encoded in multiple temporal reference systems

```
my:AbbyBirthday
  rdf:type time:Instant ;
  tplus:inDateTime my:AbbyBirthdayHebrew ;
  tplus:inTimePosition my:AbbyBirthdayUnix ;
  rdfs:label "Abby's birthdate"^^xsd:string ;
  time:inDateTime my:AbbyBirthdayGregorian ;
  time:inXSDDateTime "2001-05-23T08:20:00+08:00"^^xsd:dateTime ;
.
```

```
my:AbbyBirthdayGregorian
  rdf:type time:DateTimeDescription ;
  time:day "---23"^^xsd:gDay ;
  time:dayOfWeek time:Wednesday ;
  time:hour "8"^^xsd:nonNegativeInteger ;
  time:minute "20"^^xsd:nonNegativeInteger ;
  time:month "--05"^^xsd:gMonth ;
  time:timeZone [
      rdf:type tzont:TimeZone ;
      tzont:GMToffset "+8" ;
      tzont:name "AWST" ;
    ] ;
  time:unitType time:unitMinute ;
  time:year "2001"^^xsd:gYear ;
.
my:AbbyBirthdayHebrew
  rdf:type tplus:GenDateTimeDescription ;
  tplus:day "---01"^^tplus:genDay ;
  tplus:month "--03"^^tplus:genMonth ;
  tplus:hasTRS <http://dbpedia.org/resource/Hebrew_calendar> ;
  tplus:year "5761"^^tplus:genYear ;
  time:unitType time:unitDay ;
.
my:AbbyBirthdayUnix
  rdf:type tplus:TimePosition ;
  tplus:numericValue "990577200"^^tplus:Number ;
  tplus:hasTRS <http://dbpedia.org/resource/Unix_time> ;
  rdfs:label "Abby's birthdate in Unix time"^^xsd:string ;
.
```

Listing 4 – Events in geologic time encoded using a deep-time coordinate (in “years before present”), and using a named element from the geologic timescale

```
geotime:FormationOfEarth
  rdf:type time:Instant ;
  tplus:inTimePosition [
      rdf:type tplus:TimePosition ;
      tplus:numericValue "4.567e9"^^tplus:Number ;
      tplus:hasTRS <http://example.org/trs/ybp> ;
    ] ;
  rdfs:label "Formation of the earth"^^xsd:string ;
.

geotime:EndOfDinosaurs
  rdf:type time:Instant ;
  tplus:inTimePosition [
      rdf:type tplus:TimePosition ;
      tplus:nominalValue "Base of Cenozoic"^^xsd:string ;
      tplus:hasTRS <http://resource.geosciml.org/classifier/ics/ischart/> ;
    ] ;
  rdfs:label "End of the age of dinosaurs"^^xsd:string ;
.
```

Listing 5 – Elements from the geologic timescale, showing some topological relations

```
geotime:Cambrian
  rdf:type time:ProperInterval ;
  rdfs:label "Cambrian period"^^xsd:string ;
  time:hasBeginning geotime:BasePhanerozoic ;
  time:hasEnd geotime:BaseOrdovician ;
  time:intervalAfter geotime:Precambrian ;
  time:intervalBefore geotime:Mesozoic ;
  time:intervalBefore geotime:Ordovician ;
  time:intervalBefore geotime:Silurian ;
  time:intervalContains geotime:CambrianSeries2 ;
  time:intervalContains geotime:CambrianSeries3 ;
  time:intervalFinishedBy geotime:Furongian ;
  time:intervalStartedBy geotime:Terreneuvian ;
  time:intervalStarts geotime:Paleozoic ;
  time:intervalStarts geotime:Phanerozoic ;
.
```

```
geotime:BaseOrdovician
  rdf:type time:Instant ;
  tplus:inTimePosition geotime:BaseOrdovicianTime ;
  rdfs:label "Base of Ordovician Period"^^xsd:string ;
.
geotime:BaseOrdovicianTime
  rdf:type tplus:TimePosition ;
  tplus:numericValue "485.4"^^tplus:Number ;
  tplus:hasTRS <http://resource.geosciml.org/classifier/cgi/geologicage/ma> ;
  rdfs:label "Position of Base of Ordovician Period"^^xsd:string ;
.
geotime:BasePhanerozoic
  rdf:type time:Instant ;
  tplus:inTimePosition geotime:BasePhanerozoicTime ;
  rdfs:label "Base of Phanerozoic Eon"^^xsd:string ;
.
geotime:BasePhanerozoicTime
  rdf:type tplus:TimePosition ;
  tplus:numericValue "541.0"^^tplus:Number ;
  tplus:trs <http://resource.geosciml.org/classifier/cgi/geologicage/ma> ;
  rdfs:label "Position of Base of Phanerozoic Eon"^^xsd:string ;
.
```

6. Summary

We have introduced an extension to OWL-Time to support description of temporal position with an explicit temporal reference system, allowing use of temporal coordinate systems, ordinal temporal reference systems, and generalized calendars. The ontology uses the interval topology from OWL-Time. The new ontology is implemented as a non-intrusive extension with three new classes, eight new properties, and four datatypes, or alternatively as a replacement for OWL-Time with a smaller total element count.

The primary complications arise from the involvement of datatypes from XML Schema that are tied to the Gregorian calendar, which must be generalized for use in a wider range of applications. We define generalized types for date-fragments to match the familiar lexical productions from XSD. However, the value space for both the XSD types used in OWL-Time and the new types defined here are not supported in OWL2, so no standard reasoning support should be expected or relied upon when used in class descriptions.

Aside from the datatypes concern, the new ontologies are capable of OWL2-conformant encoding of temporal concepts used in a variety of cultural and technical settings, covering all of the temporal reference system types described in ISO 19108, but retaining exactly the same form as OWL-Time for the cases handled by the original ontology.

Acknowledgements

This work is a contribution towards the harmonization of W3C and Open Geospatial Consortium standards for geospatial data, and was supported by the CSIRO Land and Water Flagship. Michael Compton encouraged me to add the Time-new ontology to the paper, and David Ratcliffe provided advice on some of the hairy datatyping issues.

References

[1] J.F. Allen, Maintaining knowledge about temporal intervals, Commun. ACM. 26 (1983) 832–843. doi:10.1145/182.358434.

[2] D. Beckett, T. Berners-Lee, E. Prud’hommeaux, G. Carothers, RDF 1.1 Turtle, W3C Recommendation (2014). http://www.w3.org/TR/2014/REC-turtle-20140225/ (accessed May 5, 2014).

[3] P. V. Biron, A. Malhotra, XML Schema Part 2: Datatypes Second Edition, W3C Recommendation (2004). http://www.w3.org/TR/xmlschema-2/ (accessed September 9, 2014).

[4] S.J.D. Cox, S.M. Richard, A formal model for the geologic time scale and global stratotype section and point, compatible with geospatial information transfer standards,

Geosphere. 1 (2005) 119. doi:10.1130/GES00022.1.

[5] S.J.D. Cox, S.M. Richard, A geologic timescale ontology and service, Earth Sci. Informatics. 8 (2014) 5–19. doi:10.1007/s12145-014-0166-2.

[6] J.R. Hobbs, F. Pan, Time Ontology in OWL, W3C Working Draft (2006). http://www.w3.org/TR/owl-time/ (accessed February 4, 2014).

[7] ISO/TC-154, ISO 8601:2004 - Data elements and interchange formats -- Information interchange -- Representation of dates and times, (2004). http://www.iso.org/iso/catalogue_detail?csnumber=40874 (accessed September 9, 2014).

[8] ISO/TC-211, ISO 19108:2002 - Geographic information -- Temporal schema, (2002). http://www.iso.org/iso/catalogue_detail.htm?csnumber=26013 (accessed February 13, 2014).

[9] X. Ma, P. Fox, Recent progress on geologic time ontologies and considerations for future works, Earth Sci. Informatics. 6 (2013) 31–46. doi:10.1007/s12145-013-0110-x.

[10] B. Motik, B.C. Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, OWL 2 Web Ontology Language Profiles (Second Edition), W3C Recommendation (2012). http://www.w3.org/TR/owl2-profiles/ (accessed October 20, 2014).

[11] B. Motik, P.F. Patel-Schneider, B. Parsia, OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition), W3C Recommendation (2012). http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/#Datatype_Maps (accessed October 9, 2014).

[12] F. Pan, A Temporal Aggregates Ontology in OWL for the Semantic Web, in: *Proc. AAAI Fall Symp. Agents Semant. Web*, 2005: pp. 30–37. http://www.isi.edu/~hobbs/time/pub/pan-AAAI-FSS05.pdf (accessed October 10, 2014).

[13] M. Perrin, L.S. Mastella, O. Morel, A. Lorenzatti, Geological time formalization: an improved formal model for describing time successions and their correlation, Earth Sci. Informatics. 4 (2011) 81–96. doi:10.1007/s12145-011-0080-9.

[14] D. Peterson, S. (Sandy) Gao, A. Malhotra, C.M. Sperberg-McQueen, H.S. Thompson, W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation (2012). http://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/datatypes.html#gMonth (accessed October 8, 2014).

[15] Guidelines to Authors, Radiocarb. Mag. (2014). https://journals.uair.arizona.edu/index.php/radiocarbon/about/submissions#authorGuidelines (accessed September 9, 2014).