A Pattern for Periodic Intervals

María Poveda-Villalón, Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez

Ontology Engineering Group. Escuela Técnica Superior de Ingenieros Informáticos.
Universidad Politécnica de Madrid. Spain

{mpoveda, mcsuarez, asun}@fi.upm.es

Abstract. Non-convex intervals allow identifying periodic intervals with gaps between them (e.g., "every Wednesday"), while convex intervals are those that are not composed of "separate pieces" (e.g., "from 1st April 2014 to 30th April 2014"). Non-convex intervals consist intuitively of some convex subintervals with convex gaps in between. In this paper, we describe an ontology design pattern to represent *periodic intervals*, a specific case of non-convex intervals in which the period between subintervals and the duration of such subintervals are constant.

Keywords: Ontology design patterns, knowledge representation, ontology, time interval, periodic interval, interval.

1 Introduction

The temporal dimension affects almost every domain modelling or application. For example, every purchase has an ordering date and delivery, the different types of governments or wars are delimited on time by temporal points or intervals, etc. Due to its importance and presence, time theories have been developed over the years and consequently different time ontologies have been modelled [2, 6].

Time intervals can be seen as those time periods between two time points (e.g., "in which period of time the Mozart concert series happens in London? From October to February"). Time intervals can be divided into convex and non-convex. The first ones (convex intervals) are those that are not composed of "separate pieces" (e.g., a journal paper call is open during a gapless time interval); while the second ones (non-convex intervals) refer to intervals with gaps between them (e.g., every working day) [2]. In other words, non-convex intervals are those that use time units in a repetitive way or refer to recurring periods [1]. It should be noted that non-convex intervals could be periodic or aperiodic and that in this paper only periodic ones are taken into account, leaving the aperiodic ones out of the scope of this work. These periodic periods occur naturally among systems requirement description. For example, when a semantic system to manage the Football Champions League is being developed it might require representing that such a league takes place "every year from September to May."

Temporal concepts (e.g., intervals, instants, temporal units, etc.) and the relations holding between them (e.g., has end, inside, unit type, etc.) have been modelled in the context of the W3C along with the OWL-Time ontology[1] [4]. However, periodic intervals, which are a specific case of non-convex intervals where all the subintervals have the same duration and all the "gaps" between subintervals have a constant duration, are not taken into account explicitly in the OWL-Time ontology.

There are types of problems that can be solved by applying common solutions. These common solutions can be modelled as small and modularized ontologies that can be used as building blocks. These building blocks are named Ontology Design Patterns (ODPs) [3]. Taking into account this definition of ODP and the abovementioned problem, we propose an ontology design pattern called **Periodic Interval**. This pattern extends the OWL-Time ontology in order to represent periodic intervals. It should be noted that one could represent the intervals shown in Figure 1 using the OWL-Time ontology. One could represent separately the intervals I_{21}, I_{22} and I_{23} and establish that they are inside I_2 (same for I_3). However, the *Periodic Interval* pattern takes advantage of the fact that (a) the durations of I_{31}, I_{32} and I_{33} are equal and (b) the periods between them are also equal. For this reason, the proposed pattern optimizes the formalization of this kind of intervals, called periodic intervals, as there is no need to define as many subintervals as the non-convex interval is composed of.

All in all, the goal of this pattern is to represent non-convex intervals where the duration of all subintervals is constant and the period of the intervals is also uniform. Thus, this ontology design pattern allows representing periodic intervals.

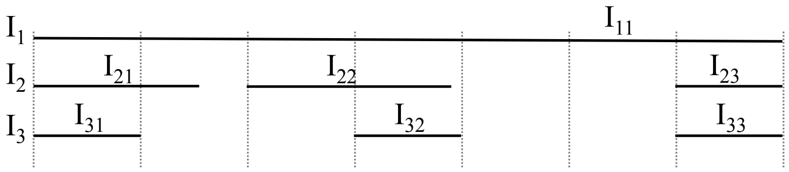

Figure 1. Convex (**I_1**) and non-convex (**I_2** and **I_3**) interval examples

The structure of the paper is the following. Section 2 describes the periodic interval pattern and its relation with the OWL-Time ontology. Section 3 presents some use cases where the proposed pattern could be applied together with a detailed example. In Section 4 we expose the related patterns and models. Finally, Section 5 presents some concluding remarks.

2 Describing the Periodic Interval Pattern

As it can be observed in Figure 2 the class "Interval" defined in the OWL-Time ontology has been extended with this pattern[2] by means of the class `PeriodicInterval`. This concept has been created in order to define periodic intervals. These intervals are defined by five elements, namely, its **beginning**, its **end**,

[1] http://www.w3.org/2006/time#

[2] The URI for the periodic interval pattern is `http://odp.linkeddata.es/PeriodicInterval#`, which provides the OWL code for the proposed pattern.

the **duration of each subinterval,** the **duration of the period**, and **the number of subintervals** contained.

In order to model the beginning and end of the interval, we have reused the relationships `owl-time:hasBeginning` and `owl-time:hasEnd` already defined in the OWL-Time ontology. By taking advantage of the concepts and relations already defined in the OWL-Time ontology instead of creating new ones, we both promote the reuse of existing models and avoid the inclusion of unnecessary complexity in the pattern being developed.

The duration of the subintervals and the period between them have been modelled by means of the relationships `hasSubIntervalDuration` and `hasPeriodDuration,` respectively. Both relationships have the class `owl-time:PeriodicInterval` as domain and the class `owl-time:DurationDescription` as range. These relationships are highlighted in Figure 2 in two ways, first, identifiers for new terms are in bold letter and, second, new terms are shown with no prefix before their identifier, while the prefix "owl-time" is included to identify the terms reused from the Time Ontology.

Finally, the number of subintervals included in a given periodic interval is represented by means of the attribute `numberOfSubIntervals` defined for the class `PeriodicInterval`. This attribute is meant to ease the calculation of the total interval duration if needed.

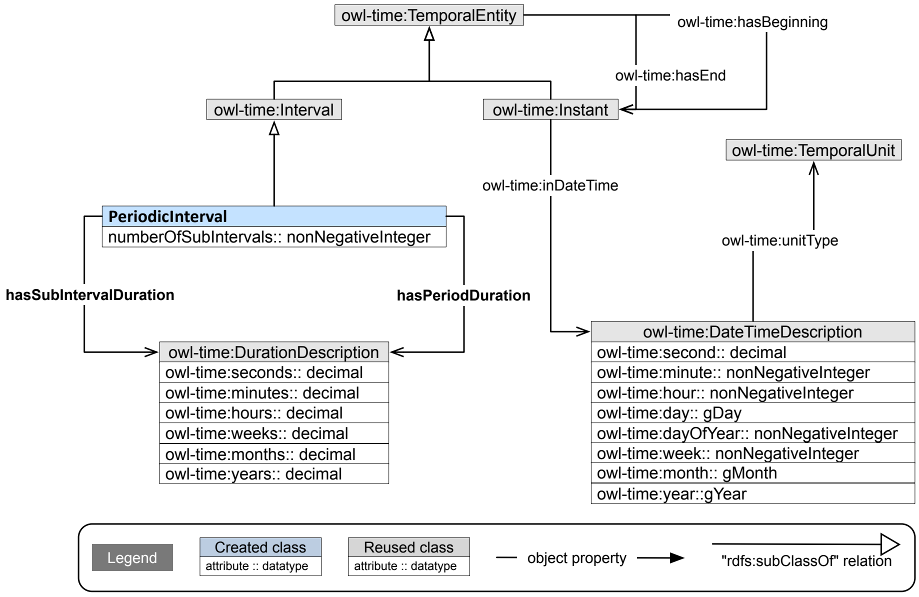

Figure 2. Periodic interval pattern

In order to clarify and support the choices made in the pattern implementation, Figure 3 shows the correspondences between the periodic interval I_3 (see Figure 1) and the proposed patterns elements. In this sense, we can observe that the interval itself would be represented as an instance of the class `PeriodicInterval`. This instance would be linked to two `owl-time:DurationDescription` instances, one representing de duration of the period (linked by the property `hasPeriodDuration`) and another one representing the duration of each subinterval (linked by the property `hasSubIntervalDuration`). The interval would be limited by its beginning and its end, which would be instances of the class `owl-time:Instant`. The given periodic interval is linked to these instants by the properties `owl-time:hasBeginning` and `owl-time:hasEnd`, respectively. It should be noted that the interval might not have a beginning or end according to the approach of infinite intervals[3].

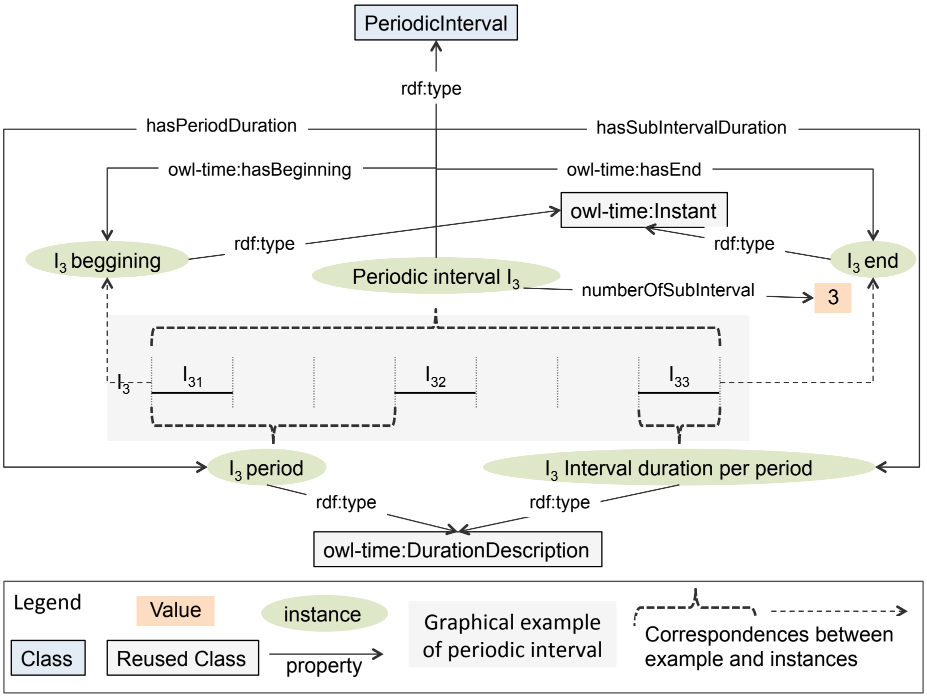

Figure 3. Graphical example of periodic interval components and relations with the periodic interval pattern elements

The classes and property declarations as well as domain and range definitions for the proposed pattern can be found in the snippet shown in Figure 4 containing the OWL code in turtle[4] serialization. The snippet contains the new classes and properties defined in the proposed pattern, as well as the axiom defined over the `PeriodicInterval` class that restricts the cardinality of the properties `owl-time:hasBeginning` and `owl-time:hasEnd` to 1.

[3] http://www.w3.org/TR/owl-time/#relations
[4] http://www.w3.org/TR/turtle/

```
@prefix : <http://odp.linkeddata.es/PeriodicInterval#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix time-owl: <http://www.w3.org/2006/time#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .


:PeriodicInterval rdf:type owl:Class ;
       rdfs:label "Periodic interval"@en ;
       rdfs:comment "a periodic interval is a non-convex interval in
which all the sub-intervals have the same duration and constant period
between sub-intervals."@en ;
       rdfs:subClassOf time:Interval ,
        [ rdf:type owl:Restriction ;
              owl:onProperty time:hasBeginning ;
              owl:maxCardinality "1"^^xsd:nonNegativeInteger
       ] ,
        [ rdf:type owl:Restriction ;
              owl:onProperty time:hasEnd ;
              owl:maxCardinality "1"^^xsd:nonNegativeInteger
       ].

:hasPeriodDuration rdf:type owl:ObjectProperty ;
       rdfs:label "has period duration"@en ;
       rdfs:comment "period between each pair of subintervals of a
given periodic interval"@en ;
       rdfs:domain PeriodicInterval2:PeriodicInterval ;
       rdfs:range time:DurationDescription .

:hasSubIntervalDuration rdf:type owl:ObjectProperty ;
       rdfs:label "has subinterval duration"@en ;
       rdfs:comment "duration of each sub-interval of a given periodic
interval"@en ;
       rdfs:domain PeriodicInterval2:PeriodicInterval ;
       rdfs:range time:DurationDescription .

:numberOfSubIntervals rdf:type owl:DatatypeProperty ;
       rdfs:label "number of sub-intervals"@en ;
       rdfs:comment "number of sub-intervals contained in the periodic
interval"@en ;
       rdfs:domain PeriodicInterval2:PeriodicInterval ;
       rdfs:range xsd:nonNegativeInteger .
```

Figure 4. Snippet of the turtle encoding for the periodic interval ODP

3 Conceivable Use Cases and Example

The Periodic Interval pattern was first created in order to support some requirements defined by the mIO! ontology network[5] [5], a context ontology in the mobile environment. Some examples of functional requirements involving periodic intervals defined for the mIO! ontology are:

- *The ontology should be able to represent non-convex intervals (e.g. every Monday).*
- *The ontology should be able to answer the following question "What is the period of the interval "every Tuesday of 2010? Answer: 7 days".*

[5] http://www.oeg-upm.net/index.php/ontologies/82-mio-ontologies

As these types of requirements are very common across domains, the solution proposed in this work could be understood as being applicable to a recurrent problem and could be considered, therefore, an ontology design pattern.

The Periodic Interval pattern can be used to express the following situations:

- Logic programming courses at *Universidad Politécnica de Madrid* (UMP), which take place every Wednesday from 9:00 to 11:00 during the second semester (from February to June).
- A pharmacy whose opening hours are from 9:00 to 21:00 every day.
- General medical check-ups for babies which take place when babies are 2 months old, 4 months old, 6 months old, 8 month old, 10 months old and 12 months old. Each check-up has a duration of half an hour.
- Summer courses at *Universidad Politécnica de Madrid* (UPM) which are hold every year during the first 20 days of July.
- The celebration of the *Nuestra Señora del Carmen* day in Santiago de Compostela (Galicia) every 16th of July.
- The episodes of each season of Games of Thrones, which are aired every Sunday during spring from 2011 until 2016. More precisely, we could represent that each episode of the 1st season was aired each Sunday from April 1, 2012 to June 3, 2012.

In order to give a more practical view of the proposed pattern, a graphical description of the Games of Thrones example (Figure 5) is provided, together with its turtle encoding (Figure 6). It should be noted that we take the day of emission as subinterval duration for the sake of clarity, instead of the exact duration of each episode. However, it could be represented in a more granular and detailed way.

In this sense, Figure 5 shows a graphical representation of the triples that would need to be generated to implement the example according to our pattern, while Figure 6 contains the actual RDF triples.

In Figure 5 and Figure 6, the namespace of the instances is indicated by the prefix “ex”. It can be observed that the main entity is the individual `ex:firstSeasonGameOfThrones` that represents the periodic interval to be described. This instance is linked to its beginning (`ex:fistSeasonGameOfThronesBeginning`) and its end (`ex:fistSeasonGameOfThronesEnd`), which are time instants determined by the dates “1st of April, 2012” (`ex:1april2012`) and “3rd of June, 2012” (`ex:3june2012`), respectively.

The periodic interval `ex:firstSeasonGameOfThrones` is also related to the instance representing the subintervals duration (`ex:subIntervalDurationGoT`), which has a duration of 1 day, and to the instance representing the period duration (`ex:periodGoT`), which has a duration of 1 week.

Finally, it is stated that the periodic interval `ex:firstSeasonGameOfThrones` contains 10 subintervals by means of the attribute `numberOfSubIntervals`.

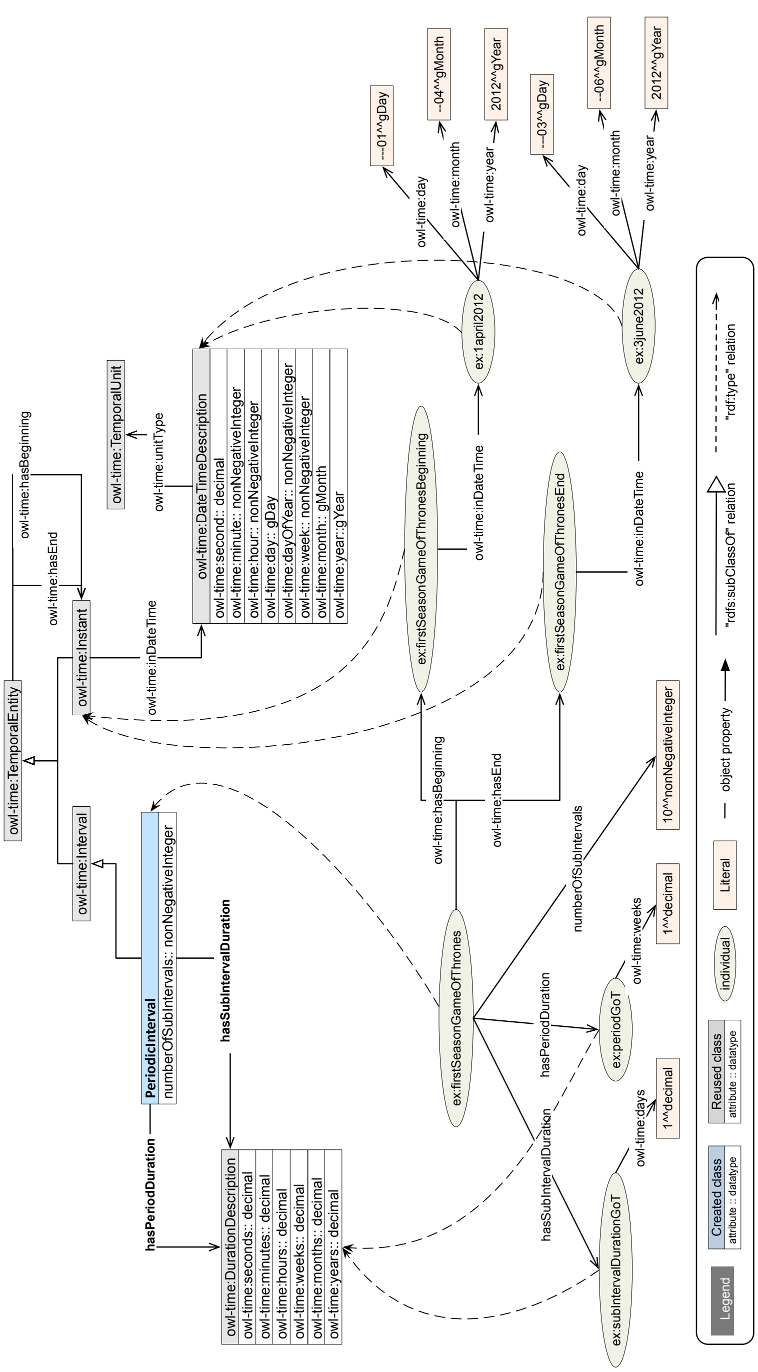

Figure 5. Graphical example of periodic interval representing the releases of the first season of Game of Thrones

```
@prefix : <http://odp.linkeddata.es/PeriodicInterval#> .
@prefix ex:
       <http://odp.linkeddata.es/PeriodicIntervalExampleGoT#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix time-owl: <http://www.w3.org/2006/time#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

ex:firstSeasonGameOfThrones rdf:type :PeriodicInterval ,
                                     owl:NamedIndividual ;
       rdfs:label "Releases of the first season of Game of
Thrones"@en;
       :numberOfSubIntervals "10"^^xsd:nonNegativeInteger ;
       time:hasBeginning ex:firstSeasonGameOfThronesBeginning ;
       time:hasEnd ex:firstSeasonGameOfThronesEnd ;
       :hasPeriodDuration ex:periodGoT ;
       :hasSubIntervalDuration ex:subIntervalDurationGoT .

ex:firstSeasonGameOfThronesBeginning rdf:type
                      owl:NamedIndividual , time:Instant ;
       rdfs:label "Beginning of the first season of Game of
Thrones"@en ;
       time:inDateTime ex2:1april2012 .

ex:1april2012 rdf:type owl:NamedIndividual ,
                      time:DateTimeDescription ;
       rdfs:label "1st of april, 2012"@en ;
       time:day "---01"^^xsd:gDay ;
       time:month "--04"^^xsd:gMonth ;
       time:year "2012"^^xsd:gYear ;
       time:unitType time:unitDay .

ex:firstSeasonGameOfThronesEnd rdf:type owl:NamedIndividual ,
                                     time:Instant ;
       rdfs:label "End of the first season of Game of Thrones"@en ;
       time:inDateTime ex:3june2012 .

ex:3june2012 rdf:type owl:NamedIndividual ,
              time:DateTimeDescription ;
       rdfs:label "3rd of june, 2012"@en ;
       time:day "---03"^^xsd:gDay ;
       time:month "--06"^^xsd:gMonth ;
       time:year "2012"^^xsd:gYear ;
       time:unitType time:unitDay .

ex:subIntervalDurationGoT rdf:type owl:NamedIndividual ,
                                     time:DurationDescription ;
       rdfs:label "subinterval duration of releases of Game of
Thrones"@en ;
       time:days 1 .

ex:periodGoT rdf:type owl:NamedIndividual ,
                      time:DurationDescription ;
       rdfs:label "period of releases of Game of Thrones"@en ;
       time:weeks 1 .
```

Figure 6. Snippet of the turtle encoding for the example of periodic interval representing the releases of the first season of Game of Thrones

This example shows that the Periodic Interval pattern could be used to generate a dataset with temporal information, for example, about the episodes of different seasons of television series (such as Game of Thrones programmed in HBO (Home Box Office)). Such RDF dataset could be published and shared within the HBO web page[6].

4 Related Modelling Practices

The issue of representing dates and times using numbers, as well as describing time intervals and recurring time intervals, has been presented in ISO 8601[7]. In a more practical way, temporal notions (e.g., intervals, instants, temporal units, etc.) have been modelled in the context of the W3C by means of the OWL-Time ontology, as mentioned in Section 1. This ontology allows representing proper intervals, that is, intervals whose extremes are different (e.g., 2013-2014 is a proper interval, while 2013-2013 is not a proper interval). A set of time ontologies was presented in [7].

To represent a time interval, which is the time between two time points, we could use the Time Interval pattern[8]. This pattern models information about the date of a time interval as well as the starting and end time of the interval.

In addition, time intervals have been modelled in the Timeline Ontology[9], which can be seen as an extension of the OWL-Time ontology. This ontology describes intervals (same as in OWL-Time) and includes different types of intervals (abstract, discrete, relative, etc.).

However, to the best of our knowledge, none of the aforementioned models (ontologies and patterns) allow the representation of periodic intervals, that is, intervals in which the period between its subintervals and the duration of such subintervals is constant.

5 Conclusions and Future Work

Temporal notions commonly appear in almost every domain, so models that semantically represent such notions are needed. One of the most used semantic models for representing temporal concepts is the W3C OWL Time ontology. This ontology models intervals, instants, temporal units, and so on, as well as the relations holding between them. However, the concept of periodic intervals, useful to represent temporal information, such as the celebration of St. James day in Santiago de Compostela (Galicia) every 25th of July, is not explicitly modelled within such an ontology.

For this reason, we propose an ontology design pattern called Periodic Interval. This pattern provides a mechanism to represent a specific case of non-convex intervals where the subintervals contained have the same duration and a constant period. This pattern reuses well-defined and known concepts and relationships related to time theory as it extends the OWL-Time ontology.

[6] http://www.hbo.com/#/game-of-thrones/

[7] http://www.iso.org/iso/home/standards/iso8601.htm

[8] http://ontologydesignpatterns.org/wiki/Submissions:TimeInterval

[9] http://motools.sourceforge.net/timeline/timeline.html

Main line of future work is to generate the necessary rules in order to transform a periodic interval (defined according to the proposed pattern) into the equivalent subintervals (according to the non-convex definition of intervals in the Time ontology). It is also planed to analyse whether other types of intervals, for example those that include irregular periods and gaps, could be represented by sibling patterns to the one proposed in this work.

Acknowledgments. This work has been partially supported by the Spanish projects *mIO!* (CENIT-2008-1019) and 4V (TIN2013-46238-C4-2-R) funded by the Spanish Ministry of Economy and Competitiveness in Spain. We are very grateful to Mariano Fernández-López for his support with time theories and to Elena Montiel-Ponsoda for her feedback.

References

1. Ladkin, P.B., *Time Representation: A Taxonomy of Interval Relations*. Proceedings of AAAI-86. 1986. http://aaaipress.org/Papers/AAAI/1986/AAAI86-060.pdf
2. Fernández-López, M., Gómez-Pérez, A. (2004). *Searching for a time ontology for semantic web applications*.
3. Gangemi, A., & Presutti, V. (2009). *Ontology design patterns*. In Handbook on Ontologies (pp. 221-243). Springer Berlin Heidelberg.
4. Hobbs, J. R. and Pan, F. *Time Ontology in OWL*. W3C Working Draft 27 September 2006. http://www.w3.org/TR/owl-time/
5. Poveda, M., Suárez-Figueroa, M.C., García-Castro, R., Gómez-Pérez, A. *A Context Ontology for Mobile Environments*. Proceedings of CIAO 2010. Lisbon, Portugal. 11 October 2010.
6. Fernández-López, M., Gómez-Pérez, A., Suárez-Figueroa, M.C. *Methodological guidelines for reusing general ontologies*. Data & Knowledge Engineering (ISSN: 0169-023X) Volume: 86 (242-275). July 2013.
7. Fernández-López, M., Gómez-Pérez, A. *Searching for a Time Ontology for Semantic Web Applications*. In Vari A, Vieu L (eds.), 3th International Conference on Formal Ontology in Information Systems (FOIS). Turin, Italy. Pp 331-441. 2004.