Modelling composite periodic Events

Cyril Faucher
L3i, University of La Rochelle
La Rochelle, France
cyril.faucher@univ-lr.fr

Jean-Yves Lafaye
L3i, University of La Rochelle
La Rochelle, France
jean-yves.lafaye@univ-lr.fr

Frédéric Bertrand
L3i, University of La Rochelle
La Rochelle, France
frederic.bertrand@univ-lr.fr

Abstract

In this paper we consider events and time properties as first rank elements in an object model representation. A general model is given for specifying composite events on the one hand and concrete or abstract temporal properties on the other. Concrete temporal properties refer to sets of calendar date times, while abstract temporal properties are specified as kinds of predicates (which can eventually be translated into their corresponding concrete counterparts). Much interest is paid to specifying periodic events and correlatively periodic temporal expressions. We show that calendars which are obviously constituted of periodic events can be modeled *via* the event and temporal models we propose. The ways to connect events to their temporal properties and to domain model elements are also addressed. This work is part of a project[1] involving two French press agencies: AFP and RelaxNews.

Keywords. event, time, periodicity, modelling

1 Introduction

Events are central concepts among the data created and processed by press agencies. Creating such events are almost completely performed by human in charge of gathering and connecting pieces of information. From a business viewpoint, events should be inserted into thematic schedules so as to ease the planning of their media coverage. Events combine various kinds of data: descriptive (what), spatial (where) and temporal (when). As regards leisure information, events often present with repeated occurrences. So, the access to an art centre (museum, monument...) or to an art session may be more or less complex, namely: "open from 9 am to 6 pm on each day between March and September except on Sundays and during public holidays". It is a key point for press agencies to record such facts in their Information Systems so as to provide media with editing material and allow short or long term prediction of events occurrences. Consequently, events with their mutual relationship and their own properties are first rank elements for journalists. Press agencies get a main part of their income by supplying their customers with such information about the occurrence and the content of future events. In this respect, we propose a Model for complex events and a way for representing temporal data which enable press agencies to design and operate the part of their information System dedicated to managing events.
The nominal scenario consists first in analysing the dispatches for pointing out significant events and extracting the corresponding temporal information. Before being eventually recorded in the system, the extraction needs to be controlled and completed. This latter step is achieved by journalists who check and step up the information on the basis of a formal language version of the extracted information. This formal rewriting provides a text willingly close to natural

[1] ANR Contint RelaxMultiMedias 2 - 2009-2012

language, hence a facilitated validation phase [9]. Many works within the Natural Language Processing (NLP) community are dedicated to temporal knowledge extraction [16]. Our contribution comprises a metamodel for events enriched by temporal expressions (PTOM: Periodic Temporal Occurrence Model), accompanied by the equivalent formal grammar evoked above. This grammar can be processed by computers [10] whilst easily understood by (non expert) human users. In fact, we pay a special attention to (pseudo) periodic events which can be expressed intentionally - and far more expressively - than in extension as a series of concrete date times [6]. e.g.: "each monday in March 2012 instead of: (2012-03-05, 2012-03-12, 2012-03-19, 2012-03-26). Events may be complex and comply to a composite pattern.
The reminder of the paper is organized as follows: Section 2 details our motivations while Section 3 presents related works about events and temporal issues in object modelling. The standard concepts for time specification are recalled in Section 4. On this basis, Section 5 presents our own metamodels for periodic temporal objects and Section 6 treats of composite interrelated events. Since calendar elements mainly are periodic events, PTOM provides a fair way for modelling the semantics of calendars; this viewpoint is the subject of Section 7. Finally, Section 8 gives a conclusion and lists future work.

2 Motivations

Let us consider one annual French music festival, namely the "Francofolies" of La Rochelle (France), as a running example for motivating and presenting the model for complex events. Since 1985, Francofolies take place every year around July 14th and will occur as long as there are sponsors and an audience. These are the temporal properties attached to the Francofolies event. Strictly speaking, the event Francofolies actually never takes place, what happens instead is the yearly series of festivals: Francofolies1985, ..., Francofolies2011.... So, "Francofolies" is a concept aggregating other more specific events e.g., the Francofolies of each year. Moreover, each edition of "Francofolies" consists of the various performances given each day during the festival. The decomposition above is purely temporal. The temporal properties bound to a composite event give information about the time extent during which the occurrences of the components events can happen. The temporal properties of a series of sibling leaves are to be found on the parent aggregated event. The temporal properties of a simple event (i.e., one with no children) bring information about the periodicity of its instances.
From another viewpoint, the Francofolies gather several special daily performances, such as the "Francofolies's Springboard (SB)"which welcomes beginners on the stage, or the "Free Hand To..." program which lets one famous artist invite whoever he wants during his show. Alternative viewpoints may be adopted. On the one hand, for a given year, say 2011, the event "Francofolies2011" presents with a set of specific events, including SB and FreeHandTo, each having its special daily program. On the other hand, Francofolies2011 may present with a daily program, each element of the daily program containing special events such as SB or FreeHandTo. Consequently, the event metamodel must take both *temporal* and *structural* aspects into consideration what is the core of this contribution.
This example demonstrates the benefit of using intensional periodic temporal expressions instead of extensional ones. One of our goals is to store a knowledge about the future occurrences of events not yet being recorded in the database. Journalists can query this information and plan their duty. As regards the above example, the information about "Francofolies" is "each year in mid-July". Thus, each year in July, the expected Francofolies edition will appear in the journalist's diary without any need of recording from the beginning the whole (potentially

inifnite) sequence of all annual editions (see Figure 1).

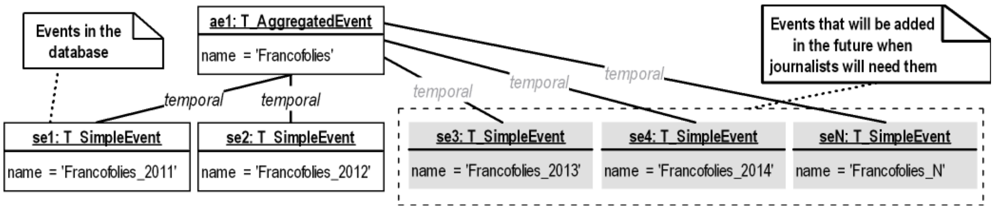


Figure 1: Events management in database

3 Related Works

Time issues have been intensively studied by many scientific communities. The present section selects some writings being in connection with the practical use cases addressed in this paper. Within the scope of software engineering, Carlson and Fowler proposed specific design patterns - namely: snapshot, effectivity and temporal object - for integrating time in Information Systems [5]. "Snapshot" specifies a starting date time after which attributes are valid. Effectivity defines the validity during a given time interval. "Temporal Object" tags an object as "changing with time" and records the whole history of the object. As a matter of fact, this can be implemented as a set of object duplicates ordered *via* a Snapshot timestamp. [15] and [4] give some other solutions to extend UML.

TOCL [18] extends OCL and provides it with linear temporal logics (LTL) elements: concepts and operators. Such assertions as "Event_A occurs until Event_B occurs" can thus be specified. Olive [12] advocates for giving events a first rank status in modelling language. The contribitions listed above show the need for integrating temporal properties into the domain model.

Let us focus now on temporal recurrence modelling. The iCalendar standard [7] shows as a textual language for specifying repeated events. It allows to launch alarms and set appointments. In iCalendar implementation, the definition of an initial event is often mandatory for defining start and end instants instead of using the recurrence rule properties, and proves to be too much constraining in practice. iCalendar does not encompass mutual relation between events.

EventsML G2[2] is a business information exchange format shared by press agencies. The International Press and Telecommunication Council (IPTC) promotes EventsML G2. The information encoded with EventsML G2 concerns events and possibly their mutual relation. EventsML G2 relies upon iCalendar to express temporal properties about event. EventsML G2 suffers the same insufficiencies as iCalendar.

Temporal properties we intend to model originate from natural language texts. Let us now consider the NLP community contributions to the subject. A model dedicated to access period specification has been designed in the domain of natural language processing (NLP) [2, 16]. This model is not rigorously an object model. The final goal is an automated selection of textual description of access periods and the final product an XML file summing up the annotations. The model does not encompass events specification. Some other works provide a taxonomy of the various manners for specifying repeated events. In particular, Mathet in [3] lists several iterators, namely: by intervals, calendar, regular, numeric, frequential, but the model remains

[2] http://www.iptc.org/site/News_Exchange_Formats/EventsML-G2/

incomplete and lacks attributes and operations.
The GLOCAL European Project[3] comes with a special event metamodel dedicated to the media domain. It comprises two packages, one for modelling the domain, and the other for modelling events plus some facilities for mapping domain elements and events. It relies upon Allen's relations. The model can also account for a causality relationship between events. On the other hand, only atomic events are modelled, i.e.: repeated events are off the point.
With respect to knowledge Engineering, Pan [13] specifies the OWL-Time ontology which can express both extensional and intensional properties. In the same vain, Yuan [17] and Perry [14] extend RDF with temporal issues grounded on the OWL-Time approach.
At last, the series of ISO 19100 which is specific to geographical information provides UML models and other precise specifications for both the geometry and the topology of time so as generic specifications for calendars and reference systems (especially ISO 19108 [11]). Repeated events are not actually addressed.
Our approach is alike that of the GLOCAL project in that we split the concerns of domain and events issues while proposing ways to model their mutual correspondence. With EventsML G2, we share the goal of treating of composite events. In some respect, PTOM extends iCalendar and ISO 10108: all rely upon the basic specification of time instants and intervals, but introduces new elements about repeated events specifying both intentional and extensional periodicity.

4 Basic Temporal Modelling: excerpt of the ISO 19108 standard

In this section, we do not intend to elaborate upon ISO 19108, but simply aim at pointing out the fundamental features which will be accessed and reused by PTOM. All such concepts can be identified by the "TM" prefix which is peculiar to the ISO 19108 standard and systematically appears in their name. Beforehand, the section expresses some justification for having chosen the ISO standard as a reference for our work.

4.1 ISO 19108, a basis for temporal modelling

Among many available standards for specifying temporal information, we selected the ISO 19108 as a basis for our work for the following conjoint reasons. It presents a UML object model being very well fitted to Model Driven Engineering techniques which are of prior interest with respect to our goals. Having a pivot object model at one's disposal, leverages the mapping of the temporal concepts in use with other application and domain oriented time specification languages, namely:

- OWL-Time for specifying an ontology including time issues, expressing logical time rules and performing formal reasoning about time properties,
- SQL-Time for relational database querying and for representing active constraints upon dates and time intervals,
- iCalendar for calendar applications that deal with concrete and periodic event, eventually bonded to calendar dates and time.

[3] `http://www.glocal-project.eu//sites/all/modules/filemanager/files/Deliverables/GLOCAL-D1.2_final.pdf`

Domain specific time models can as well be mapped to the pivot object model if necessary. This is the case within the scope of text mining when capturing the semantics of time expressions from pieces of information expressed in natural language. The ISO 19108 document is part of the ISO19100 series which applies to geographical issues. In practice, spatial aspects most often appear tightly connected with temporal information; so, no additional effort will be needed to glue temporal models and geographical ones.

4.2 Introduction to the ISO 19108 Temporal Primitive Model

The basic class Figure 2 is TM_GeometricPrimitive with two disjoint specializations, namely TM_Instant and TM_Period which respectively indicate a zero and one dimensional geometric element on the time scale. TM_GeometricPrimitive is equipped with methods that allow computing durations and consequently distances between its instances. Ordinal relative positions between instances can be computed as well. The standard provides enumerations and datatypes which can meet the common needs of users.

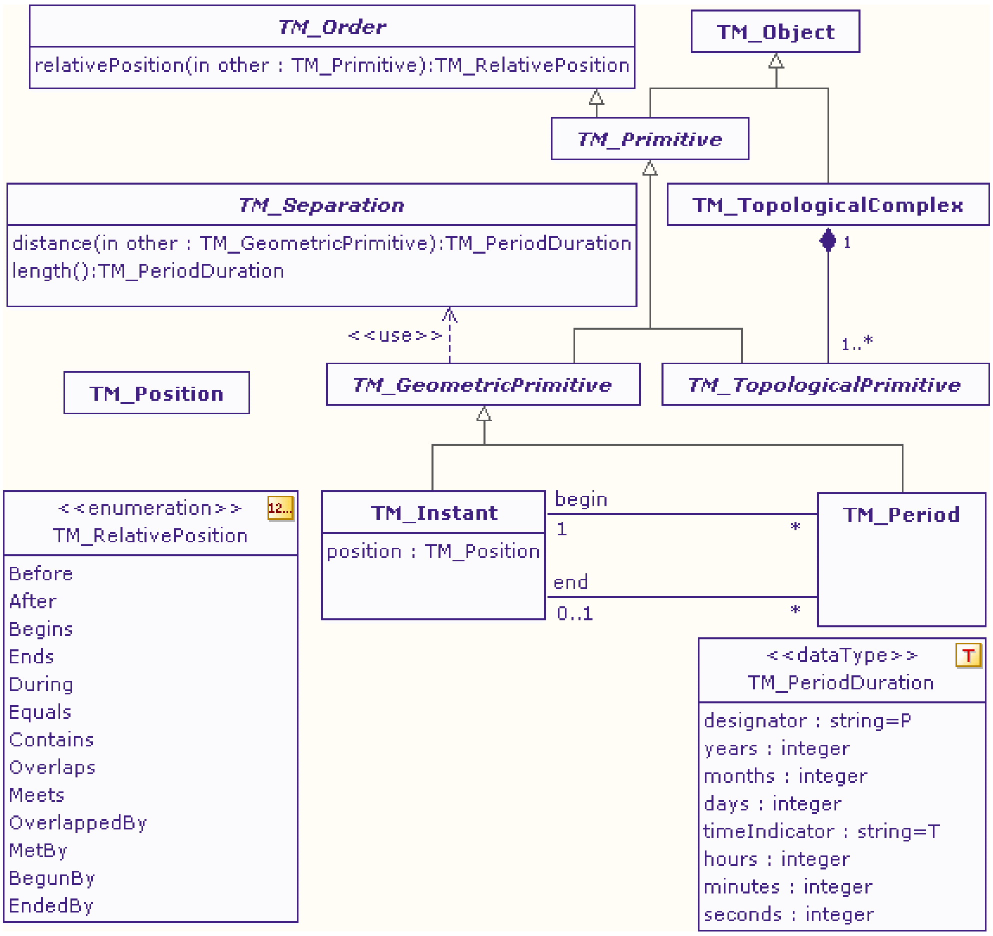


Figure 2: ISO 19108 excerpt

Distances and durations are numerical values with given temporal units which need being interpreted with respect to a given TM_ReferenceSystem. The ISO 19108 dedicates a whole section to this concern and provides all useful methods and encoding rules to specify a Reference System and implement consistent temporal computations. TM_Calendar and TM_Clocks are subclasses of TM_ReferenceSystem, and hence present with special methods which can return sets of concrete date-time values, and allow switching from one TM_ReferenceSystem to another. The standard also addresses granularity (resolution) and indeterminacy (indeterminatePosition: TM_IndeterminateValue) by providing ad hoc methods attributes and types. These latter aspects are not presented here. Of course, a set of OCL constraints are specified in the standard and ensure the consistency of any Temporal Primitive model conformant set of instances.

5 Calendar and Periodic expressions

As evoked in the introduction, and shown in Figure 3, PTOM distinguishes between concrete temporal occurrences which are expressed as extensional sets of calendar date times, and abstract (intensional) specifications of series of temporal primitives. The paper pays a special attention to periodic temporal expression since calendars essentially capture periodic milestones. An association between concrete and periodic temporal occurrences indicates connects an abstract specification with the series of calendar date times which can be computed (see the extension(start, end) method in Figure 6). The ConcreteTemporalOccurrence aggregates TM_GeometricPrimitive from the ISO Standard, and PeriodicTemporalOccurrence aggregates PeriodicRule from PTOM. In Subsection 5.1, we describe how periodic issues - which are prominent in calendars - can be reused, and then develop the specification of PeriodicRule in Subsection 5.2.

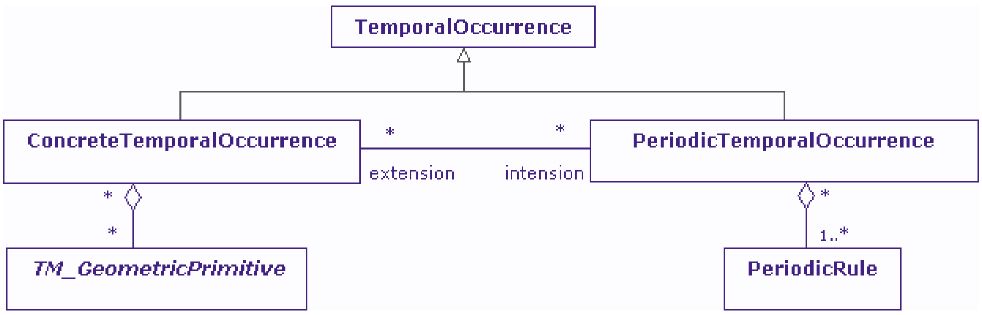


Figure 3: Concrete and Periodic Temporal Occurrences

5.1 Calendar Periodic Descriptor

A calendar provides built in concepts which represent periodic events such as named days and months. March may either be referred by its name (i.e., March), or by its rank in the year (the 3rd month). This also applies to days in a week (modulo the location context). Besides, rank is the only direct way to identify a week in a calendar month or in a year, or to indicate a year in a century or at last a century in an era.

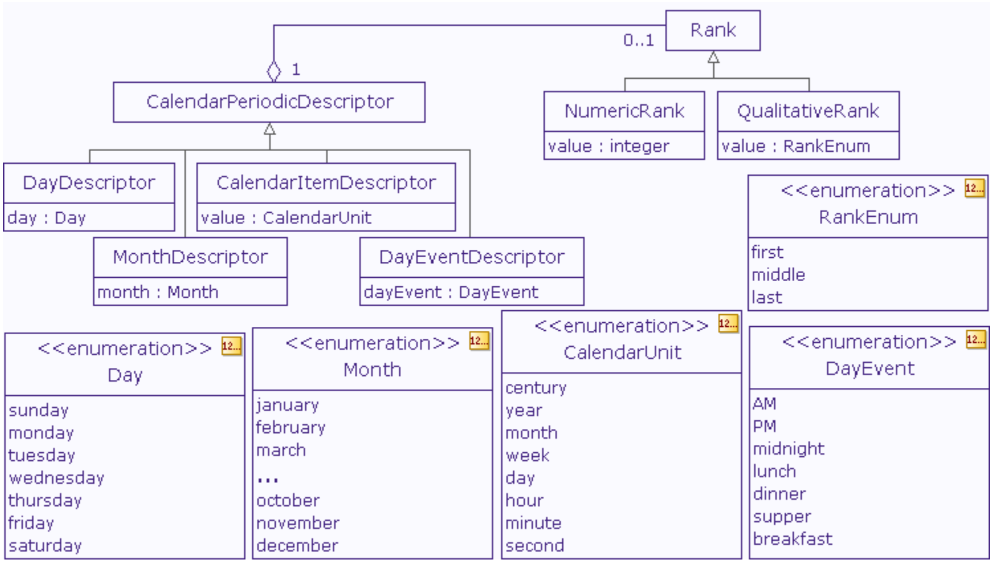


Figure 4: Periodic Calendar Descriptor

The model depicted in Figure 4 allows to specify elementary CalendarPeriodicDescriptor on the basis of the common calendar semantics, and then to aggregate them in an AbsoluteTemporalExpression. For sake of brevity, we shall better illustrate the semantics and show few examples, than provide the complete formal specification which includes a series of OCL constraints. The syntax for describing the instances is willingly implicit but straightforward and thus far much concise than a usual complete instance diagram.

```
"Each second week of each year"
{ { CalendarItemDescriptor.value = week;
      NumericalRank.value = 2 }
   { CalendarItemDescriptor.value = year }  }
"Each week of the third month of each year"
{ { CalendarItemDescriptor.value = week }
      { CalendarItemDescriptor.value = month;
        NumericalRank.value = 3 }
    { CalendarItemDescriptor.value = year }  }
```

Additional constraints upon the use of rank, named entities and granularity ensure temporal expressions are well formed [8]. The above specifications have a resolution of one week. In case more precise information about the days when the event in hand does occur were available, a DayDescriptor should be specified. Time would be treated accordingly. The model also allows specifying imprecise occurrences with DayEvent. The metamodel is extensible, in the sense that new literals can be added to genuine enumerated types such as DayEvent and while CalendarDescriptor can be extended by inheritance.

5.2 Temporal Periodic Rule Expression

Temporal properties are attached to events *via* an association between Event and PeriodicTemporalOccurrence. Hence, PeriodicTemporalOccurrences can be reused by various Events. A PeriodicRule specifies a simple periodic occurrence pattern (i.e., with a single frequency). The

PeriodicTemporalOccurrence Class allows to specify multi-frequential occurrence patterns, by aggregating PeriodicRules. Here, aggregation means adding the frequencies of all components. Let us notice that in Figure 5, the group of classes with a colored background permits to build complex periodic temporal patterns, either instant or interval, by combining PeriodicCalendarDescriptor instances. (Nb.: PeriodicRelativePosition is mentioned but not addressed here).

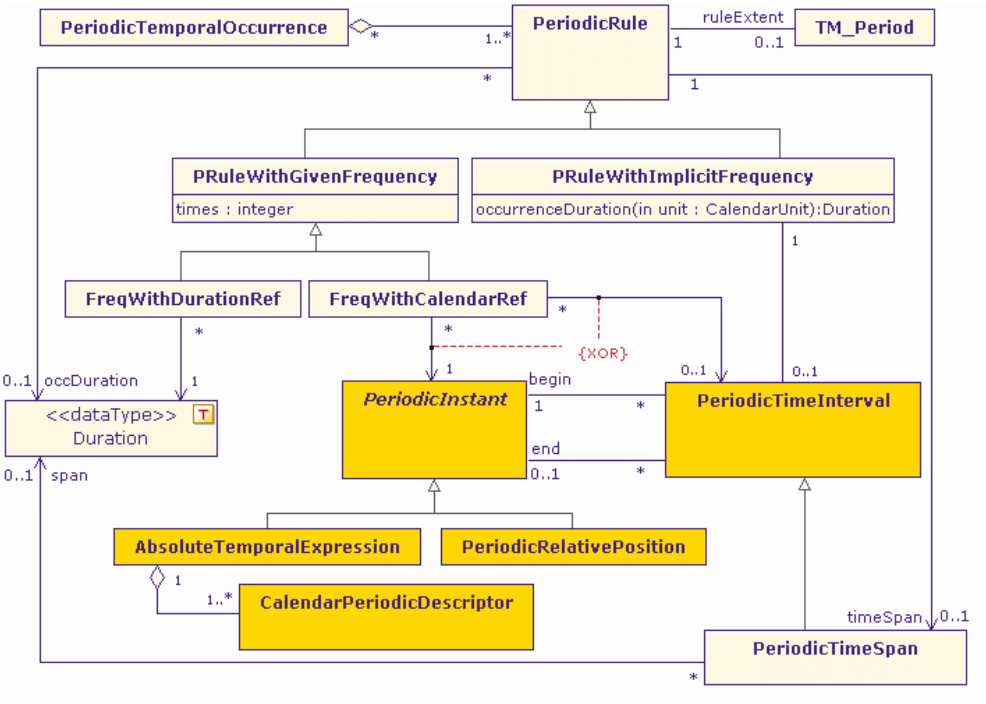


Figure 5: Periodic Rule

PeriodicInstant and PeriodicTimeInterval are the entry points to these constructs. In some respect, the pair of classes, namely TM_Instant and TM_Period could appear similar to the other pair made of PeriodicInstant and PeriodicTimeInterval. The latter can be viewed as an extension of the former, passing from concreteness to abstraction and from static to periodic. Of course, the former is more complex and induces the specification of consistency constraints. For instance, PTOM imposes (*via* OCL constraints) that for a given PeriodicTimeInterval, the PeriodicInstant respectively referred by begin and end roles should have equal frequencies. There are actually three manners for defining a PeriodicRule. Basically, either a frequency is explicitly defined (PRuleWithGivenFrequency), or it remains implicit and refers to a PeriodicTimeInterval. Once a frequency is given, defining the corresponding time unit may be done with a FreqWithDurationRef or with a FreqWithCalendarRef. This eventually makes three solutions, each being discussed below.

1. FreqWithDurationRef uses common calendar units as defined in the Duration datatype which specializes TM_PeriodDuration. Such a specification is for instance: "three times every two months".

2. FreqWithCalendarRef uses PeriodicInstant and allows constructs such as "four times each Sunday". The expressiveness of PeriodicInstant permits even more complex and accurate constructs e.g., "three times every week of the third month of each year".

3. PRuleWithImplicitFrequency directly reuses the PeriodicTimeInterval class which intrinsically is periodic, hence needing no external frequency specification. The independent specification of "begin" and "end" of the PeriodicTimeInterval permits to tackle cases when the event in hand is periodic with a possibly varying duration e.g., "begins on the 15th of every month and ends on the last day of every month".

Unlike the 1st and 2nd cases, the phase of the phenomenon as well as its frequency and duration are explicitly specified in the 3rd case. PTOM also addresses the case of events that occur periodically during series of periods which themselves are periodic. This is for instance the case in the following sentence: "Each year, Event E occurs once a week from June to August". Here, "once" is the explicit frequency, "week" is the corresponding CalendarPeriodicDescriptor, and "June to August" is a sliding window called PeriodicTimeSpan, that filters the prior partial specification "once a week". The PeriodicTimeSpan is a PeriodicTimeInterval whose end is either specified or derived from the knowledge of both the role "span" and the mandatory begin. In order to restrain the period during which a PeriodicRule applies, a ruleExtent is needed. It is a constant time interval (TM_Period), which discards all occurrences of the associated PeriodicRule that do not lay inside the ruleExtent. Within some respect, the ruleExtent property can indicate an approximate duration of the PeriodicRule considered as a whole (Convex hull of all its occurrences). It must be clearly distinguished from the *occDuration* which expresses the duration of only one occurrence. PTOM allows adding exceptions to TemporalOccurrences. TemporalExceptions are defined with the same syntax as TM_Primitive, PeriodicRules or FeatureRelativePosition. The only constraint states that TemporalExceptions cannot be embedded within one another. The semantics is that the occurrences specified in the TemporalException are withdrawn (if they exist) from those of the corresponding TemporalOccurrence.
The models for PeriodicTemporalOccurrence and PeriodicRule have been tested upon a set of 513 actual temporal expressions collected among dispatches from various press agencies. Tested means we automatically transformed the annotated text provided by an extractor into instances of PTOM classes. The output is a list of intentional specifications for access periods [10] which can be found at the website given in reference[4].

6 Event Metamodel

As evoked in the motivations, we need to define 2 kinds of relations between events. The first one concerns the *temporal* aspect, the second one the *structural*. Figure 6 below shows how we specialize the composite pattern so as to treat these remarks.

[4] `http://relaxmultimedia2.univ-lr.fr/ap2tom/benchmark.html`

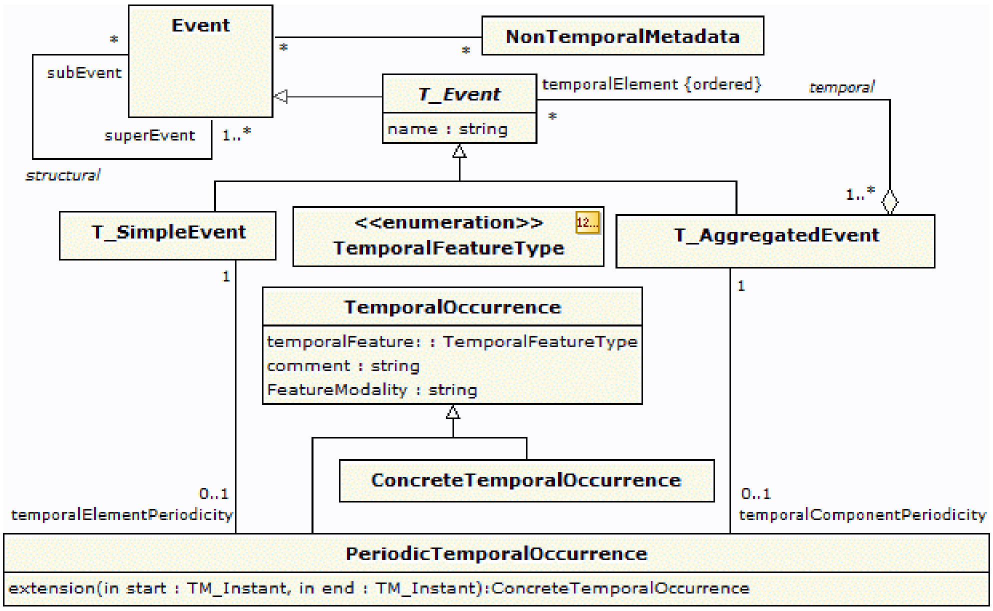
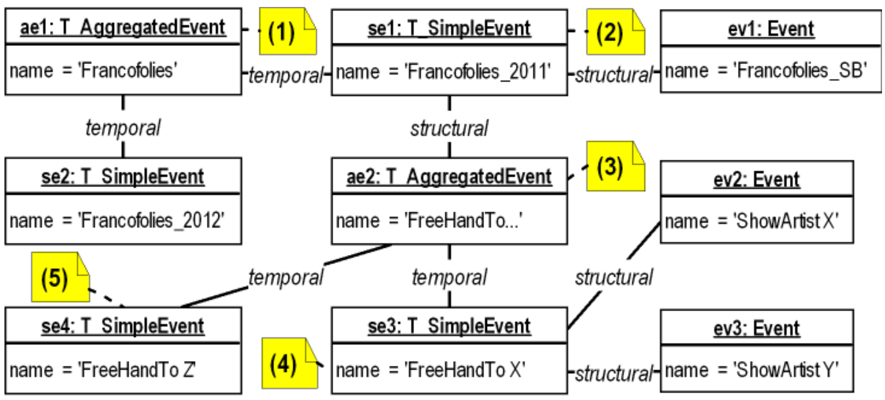


Figure 6: Event Metamodel

The TemporalOccurrence class has two attributes that respectively recall in brief and in detail which is the temporal feature in hand. For instance, we specify the access to Francofolies performances. A temporal event (T_Event) is a kind of Event and stands as the central class in the model with two disjoint subclasses. T_SimpleEvent is a leaf. T_AggregatedEvent gathers T_Events according to a temporal viewpoint. In the composite pattern, T_Events are bound to the PeriodicTemporalOccurrence Class which is the entry point to the temporal model.
PeriodicTemporalOccurrence specializes TemporalOccurrence and inherits attributes that identify the special kind of temporal property which is considered e.g., Open, Closed, Restricted etc. Two specific associations are respectively set between PeriodicTemporalOccurrences and simple and aggregated events. They depict the temporal properties of the components of a T_Event which are explicitly specified in case of a T_AggregatedEvent, and which are not in case of a T_SimpleEvent. It clearly appears that constraints have to be verified so as to ensure that the temporal properties of any T_AggregatedEvent are consistent along its lineage.
A PeriodicTemporalOccurrence is a formal abstract specification for a series of concrete calendar equivalent counterparts which can in general be computed *via* the extension(start,end) method. As seen with the Francofolies example, it is necessary that the model can account for the structural decomposition of Events. This information is carried by the reflexive "structural" aggregation on the Event Class. Structural sub events indeed are constrained by the temporal properties of their super events, but - unlike the case of the temporal relationship - such sub events are not occurrences of the super event temporal pattern. Other information (Identification, Spatial, Quality, Portrayal, Distribution...) are represented here by the informal NonTemporalMetaData rectangle, and are precisely described in the ISO 19115 standard.
Figure 7 gives an excerpt of the instance diagram corresponding to the second viewpoint quoted above about the running example. The example we present contains both intentional and extensional definitions. In our approach, each event is an instance of the Event class. This is especially the case for both objects named "Fancofolies" and "Fancofolies_2011". Besides, the

temporal link drawn between "Francofolies" and "Francofolies_2011" indicates that the latter is one special occurrence of the PeriodicTemporalOccurrence attached to the former. This kind of modelling allows to tackle the specification of embedded periodic events. As a matter of fact, any object can appear as an instance of Event hence inheriting of temporal (resp. possibly *structural*) properties *via* the *temporal* (resp. *structural*) association.



(1) T_AggregatedEvent.temporalComponentPeriodicity = **each July, starts on 1985**
(2) PeriodicTemporalOccurrence.extension = **July 2011**
(3) T_AggregatedEvent.temporalComponentPeriodicity = **each day, from 2011-07-11 to 2011-07-15**
(4) PeriodicTemporalOccurrence.extension = **2011-07-11**
(5) PeriodicTemporalOccurrence.extension = **2011-07-13**

Figure 7: Excerpt of one possible instance diagram for the "Francofolies" case study

Figure 7 presents an example which illustrates the use of *temporal* and *structural* links in the Event metamodel. Periodic temporal properties are shown as notes attached to objects and bear a numbered reference to their textual specification given downwards. It also permits to view and check the consistency of the various periodic rules specified across the model:

- se3 (rule 4) and se4 (rule 5) T_SimpleEvents respectively occurring on "2011-07-13" and "2011-07-11" both satisfy the periodic rule of their parent T_AggregatedEvent ae2 (rule 3), namely "each day, from 2011-07-11 to 2011-07-15",
- the periodic rule for ae2 which is a structural sub-event of se1 complies the parent specification (rule 2) on "July 2011",
- eventually, the rule for se1 itself complies to the temporal property of the root Event named Francofolies (rule 1): once "each July from 1985".

7 Calendars, PTOM and Event Metamodel

Many concepts in PTOM and in ISO 19108 rely on the seminal definitions of what is a calendar. Annex D in ISO 19108 is an informal text dedicated to describing Calendars which expresses a common intuitive knowledge and experience. We intend to provide an example of how the event metamodel joint with PTOM can result in expressing the semantics of a Calendar (e.g., Gregorian calendar). The task is hard when the goal is to build a complete stand alone specification.

We shall only give a brief survey of our approach here. The basic idea is that many issues in a calendar are periodic events. Hence, using the event metamodel is straightforward; so is as well, the use of PTOM for specifying the temporal properties of these events. Figure 8 shows an excerpt of the instance model issued from the Event metamodel with PTOM. Names of temporal most interesting events are printed in bold. YEAR is a T_AggregatedEvent whose components are the various years, each being identified by its number. YEAR happens 100 times a century (PeriodicRule). Similarly, FEBRUARY happens only once a year and aggregates the instances of every second month in the year (e.g., FEBRUARY2010). These are *temporal* aspects.

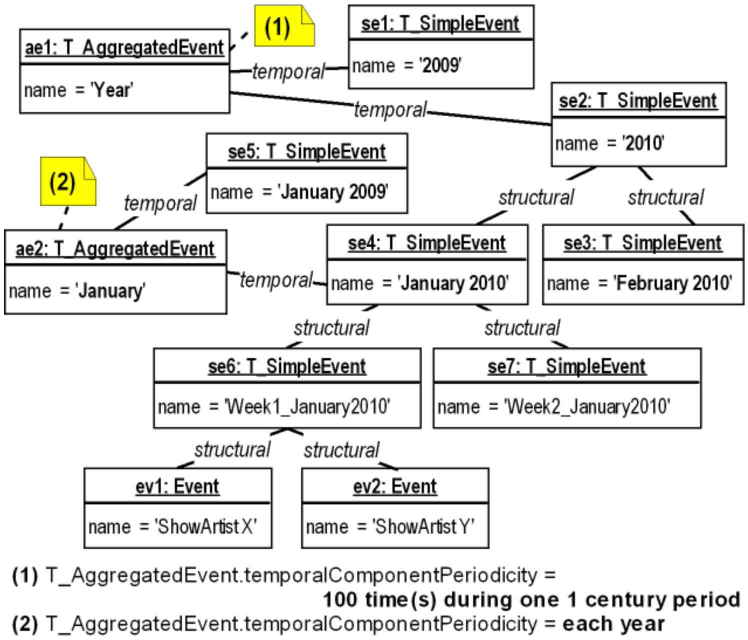


Figure 8: Excerpt of the calendar specification using both PTOM and the Event metamodel

From a *structural* point of view, FEBRUARY2010 is a component of year 2010. In the same manner FEBRUARY2010 aggregates several DAYS. Using the TM_RelativePosition enumeration (see Figure 2) PeriodicRules allow to specify such properties as: "JANUARY Meets FEBRUARY" and "FEBRUARY Meets MARCH"... which are the essence of the Calendar topology (provided a natural extension of the "Meet" Allen's relation [1] to periodic events). CALENDAR_MONTH (not shown in the diagram) is defined as a T_AggregatedEvent with a frequency of 12 for a CalendarPeriodicDescriptor with value "year". The corresponding temporalElements (see role in Figure 6) are the series of named periodic months e.g., JANUARY, FEBRUARY, etc. Let us insist upon the fundamental difference between CALENDAR_MONTH which is a complex event and the element in the CalendarPeriodicDescriptor namely "month" which indicates a duration unit. The knowledge that a month's duration is 28, 29, 30 or 31 days is a matter of solving granularity issues. Nonetheless, information about the duration of events can be recorded by specifying the value of occurrenceDuration in PeriodicRules. As an additional example, the Calendar model can be refined by introducing such Events as Leap Year. Figure 9 gives a rough view of the way to do so. In fact, the definition of a Leap Year periodicity is more complex, but the goal here is simply to outline the solution.

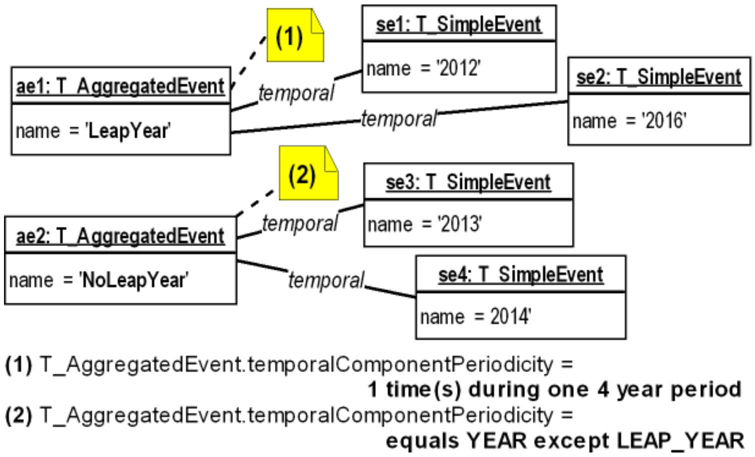


Figure 9: Defining "Leap Year" events

Though of frequent and much common use, calendars embed some non straightforward issues, resulting from the ill synchronization between calendar elements and actual astronomical phenomena. One obvious example quoted in Figure 9 is the case of leap years. Another is the fact that some weeks may overlap two consecutive years or months. More precisely, a week is assigned to a unique year (resp. month), which is the one when it starts. Consequently, some days in week W1 assigned to year Y1 (resp. month M1) may belong to year Y2 (resp. month M2). This of course can only happen when Y1 *meets* Y2 (resp. M1 *meets* M2) in the Allen's relation sense, and concern the last week of a year (resp. of a month).

8 Conclusion and future work

Aside, but in correlation with a rich lot of works upon time processing, we focused on the need for specifying a general metamodel for treating of concrete leisure and cultural events. i.e., events that have a complex structure, which are located in space and time, are connected with one another, are related to some context and several key personalities. Our scope is limited to structural and temporal property modelling, but entry points are set in our models to merge other kinds of information.

Our contribution is a metamodel (PTOM) for complex periodic events which is specified and made operational *via* model driven engineering techniques which allow a satisfactory and rational management of complexity. We indicated how the Gregorian calendar could be specified with PTOM. In fact the periodicity of calendar elements is easily mastered *via* PTOM elements. As an instance, "year" is an event which starts on January 1st, and ends on December 31st. All elements in the calendar can be inductively built accordingly.

Our works presently are in progress in three main directions. First, building a framework for connecting PTOM with pre-existing given UML business models and then friendly adding periodical properties to business classes and instances. Second, extending OCL syntax so as to allow the specification and computation of formal temporal properties in OCL constraints upon UML model elements. Third, develop temporal reasoning on the basis of a PTOM model of a calendar semantics.

Acknowledgements This work is granted by the French National Research Agency (ANR-Contint, RelaxMultiMedias 2 project).

References

[1] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] Delphine Battistelli. *Linguistique et recherche d'information : la problématique du temps*. Traitement de l'information. Hermes Science Lavoisier, 2011.

[3] Gérard Becher, Patrice Enjalbert, Estelle Fievé, Laurent Gosselin, François Lévy, and Gérard Ligozat. Rapport technique du projet OGRE. *CoRR*, abs/cs/061, 2006.

[4] Jordi Cabot, Antoni Olivé, and Ernest Teniente. Representing Temporal Information in UML. In *UML'03*, Lecture Notes in Computer Science, pages 44–59. Springer Berlin / Heidelberg, 2003.

[5] Andy Carlson, Sharon Estepp, and Martin Fowler. Temporal Patterns. In Harrison Foote, editor, *Pattern Languages of Program Design 4*, pages 241–262. Addison-Wesley, 1999.

[6] Rudolf Carnap. *Meaning and Necessity*. University of Chicago Press, 1947.

[7] B. Desruisseaux. Internet Calendaring and Scheduling Core Object Specification (iCalendar) - RFC5545, 2009.

[8] Cyril Faucher, Samnang Chea, Frédéric Bertrand, and Jean-Yves Lafaye. Validation sémantique d'objets à l'aide d'un modèle de référence et de contraintes. In *IDM 2011*, pages 109–113, Lille, France, 2011.

[9] Cyril Faucher, Jean-Yves Lafaye, Frédéric Bertrand, and Charles Teissèdre. Modélisation et reformulation d'expressions temporelles extraites de textes en langage naturel. In *AFADL 2010 (10es Journées Francophones Internationales sur les Approches Formelles dans l'Assistance au Développement de Logiciels)*, pages 213–216, Futuroscope-Poitiers, France, 2010.

[10] Cyril Faucher, Charles Teissèdre, Jean-Yves Lafaye, and Frédéric Bertrand. Temporal Knowledge Acquisition and Modeling. In *EKAW 2010 - Knowledge Engineering and Knowledge Management by the Masses*, volume 6317 of *LNCS (LNAI)*, pages 371–380, Lisbon, Portugal, 2010. Springer-Verlag.

[11] ISO. Text of 19108 Geographic information - Temporal schema, 2002.

[12] Antoni Olivé and Ruth Raventós. Modeling events as entities in object-oriented conceptual modeling languages. *Data & Knowledge Engineering - Special issue: ER 2004*, 58(3):243–262, 2006.

[13] Feng Pan and Jerry R Hobbs. Temporal Aggregates in OWL-Time. In *FLAIRS 2005*, pages 560–565, 2005.

[14] Matthew Perry, Farshad Hakimpour, and Amit P Sheth. Analyzing theme, space, and time: an ontology-based approach. In *GIS*, pages 147–154, 2006.

[15] Marianthi Svinterikou and Babis Theodoulidis. TUML: A Method for Modelling Temporal Information Systems. In Matthias Jarke and Andreas Oberweis, editors, *Advanced Information Systems Engineering*, volume 1626 of *Lecture Notes in Computer Science*, pages 456–461. Springer Berlin / Heidelberg, 1999.

[16] Charles Teissèdre, Delphine Battistelli, and Jean-Luc Minel. Resources for Calendar Expressions Semantic Tagging and Temporal Navigation through Texts. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 3572–3577, Valletta, Malta, 2010. European Language Resources Association (ELRA).

[17] May Yuan. Modeling Semantical, Temporal and Spatial Information in Geographic Information Systems. In Max Craglia and Helen Couclelis, editors, *Geographic Information Research: Bridging the Atlantic*, pages 334–347. Taylor & Francis, 1996.

[18] Paul Ziemann and Martin Gogolla. An Extension of OCL with Temporal Logic. In Jan Jurjens, Maria Victoria Cengarle, Eduardo Fernandez-Medina, Bernhard Rumpe, and Robert Sandner,

editors, *Critical Systems Development with UML - Proceedings of the UML'02 workshop*, volume TUM-I0208, pages 53–62, 2002.