Core Information Model (CoreModel)

TR-512.18

Temporal Expression

Version 1.6
January 2024

ONF Document Type: Technical Recommendation
ONF Document Name: Core Information Model version 1.6

Disclaimer

Open Networking Foundation
1000 El Camino Real, Suite 100, Menlo Park, CA 94025
www.opennetworking.org

Important note

This Technical Recommendations has been approved by the Project TST, but has not been approved by the ONF board. This Technical Recommendation is an update to a previously released TR specification, but it has been approved under the ONF publishing guidelines for 'Informational' publications that allow Project technical steering teams (TSTs) to authorize publication of Informational documents. The designation of '-info' at the end of the document ID also reflects that the project team (not the ONF board) approved this TR.

Table of Contents

List of Figures

List of Tables

Document History

Version	Date	Description of Change
1.6	January 2024	Initial Version

1 Introduction to the document suite

This document is an addendum to the TR-512 ONF Core Information Model and forms part of the description of the ONF-CIM. For general overview material and references to the other parts refer to TR-512.1.

1.1 References

For a full list of references see TR-512.1.

1.2 Definitions

For a full list of definition see TR-512.1.

1.3 Conventions

See TR-512.1 for an explanation of:

- UML conventions
- Lifecycle Stereotypes
- Diagram symbol set

1.4 Viewing UML diagrams

Some of the UML diagrams are very dense. To view them either zoom (sometimes to 400%) or open the associated image file (and zoom appropriately) or open the corresponding UML diagram via Papyrus (for each figure with a UML diagram the UML model diagram name is provided under the figure or within the figure).

1.5 Understanding the figures

Figures showing fragments of the model using standard UML symbols as well as figures illustrating application of the model are provided throughout this document. Many of the application-oriented figures also provide UML class diagrams for the corresponding model fragments (see TR-512.1 for diagram symbol sets). All UML diagrams depict a subset of the relationships between the classes, such as inheritance (i.e. specialization), association relationships (such as aggregation and composition), and conditional features or capabilities. Some UML diagrams also show further details of the individual classes, such as their attributes and the data types used by the attributes.

2 Introduction to Temporal Expression

The focus of this document is on a model of time periods. Application of this model to the other Core Model entities will allow for representation of the history of actual lifecycle changes and representation of future possibility/intention. For the latter some of the multiplicities in the model may need to be refined.

There are many models/representations of temporal expression. These have been explored to inform the development of the model described in this document.

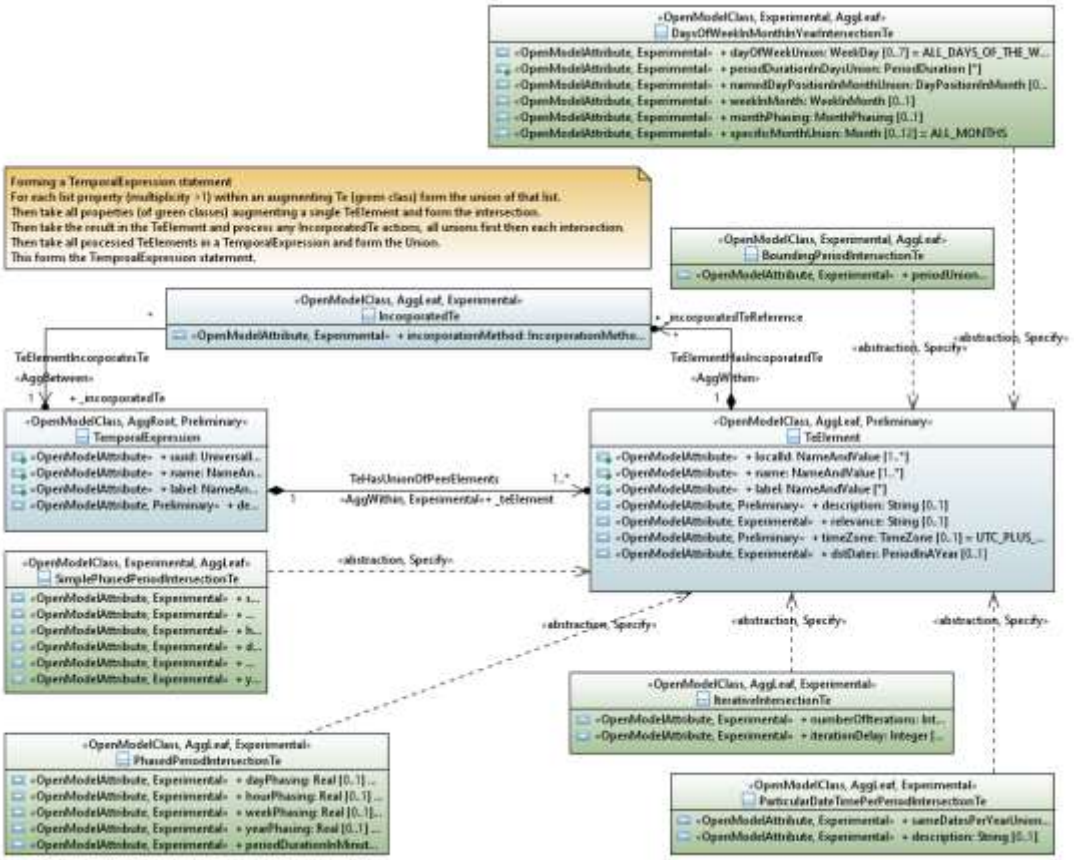
A data dictionary that sets out the details of all classes, data types and attributes is also provided (TR-512.DD).

3 Temporal Expression model detail

The model:

- Represents time using a combination of absolute calendar statements, statements of periodicity, statements of duration, statements of iteration and statements of phase.
- Provides temporal expressions in terms of combinations of minutes, hours, days, weeks, months and years.
- Accounts for time zone and DST
- Is formed around a TemporalExpression class that is built from a union of TeElements where each TeElement is formed by an intersection of definitions where each definition is applied via «specify» abstraction (such that the definition augments the TeElement).
- Supports the combination of expressions where, in general, specifications of the same sort combine by union and of different sort combine by intersection. Following the model explanation there is a section of examples to illustrate this.
- Enables a TemporalExpression to incorporate other previously defined TemporalExpressions where the incorporation method can be union, intersection or intersection with the complement of the incorporated TemporalExpression
- Has been designed such that all combinations give a defined output (i.e., no combinations are "illegal"), although some may result in no period

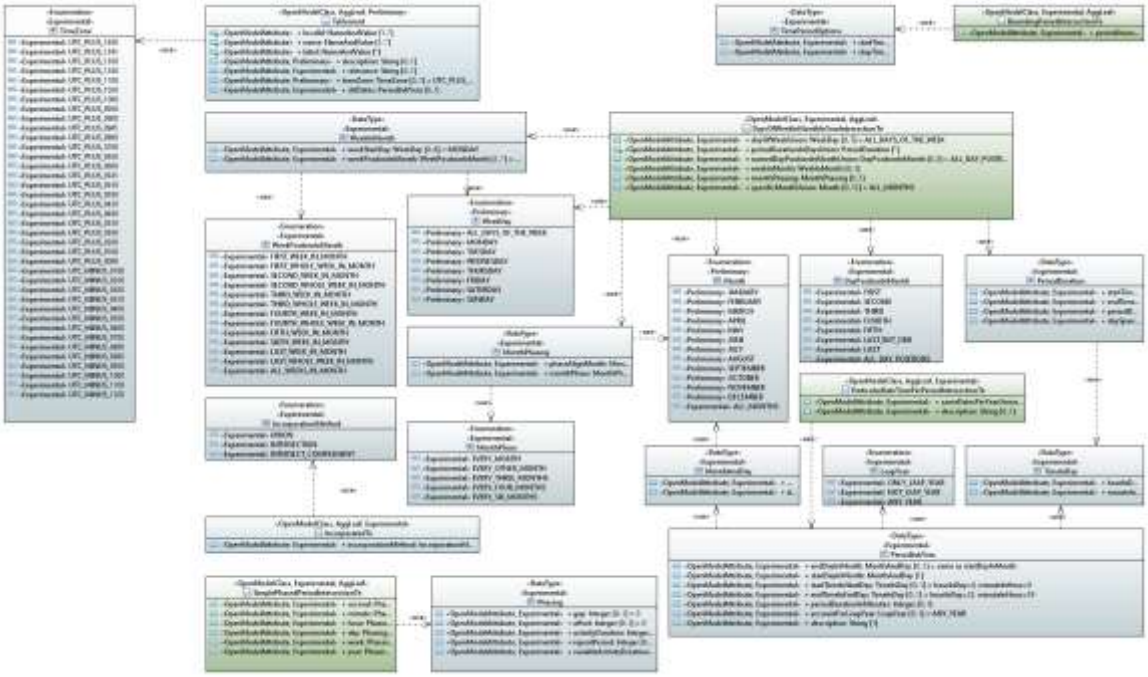
The diagram below shows the model.



CoreModel diagram: TemporalExpression-Overview

Figure 3-1 Temporal Expression Model

The diagram below shows the usage of data types in the model and exposes the detail of the model.



CoreModel diagram: TemporalExpression-DataTypes

Figure 3-2 Temporal Expression Data Types

The following sections provide definitions of the structures depicted in the diagrams above.

3.1 Temporal Expression Class Model

This section provides the temporal expression classes.

3.1.1 BoundingPeriodIntersectionTe

Qualified Name:
TemporalExpression::TemporalExpressionModel::TemporalExpressionAugments::BoundingPeriodIntersectionTe

This defines the time extent of the temporal expression element.
It covers both single shot reservations and complex reservations.
This time takes precedence over periodInDay as it is the boundary of the overall reservation.
May have various bounding periods for the same inner detail.
If more than one then only the last can be open ended etc.
This class is AggLeaf.

This class is Experimental.

Table 1: Attributes for BoundingPeriodIntersectionTe

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
periodUnion	Experimental	The start time may be any time including start of a day. If the value is absent, there is no start. The end time may be any time including the end of a day. If the value is absent there is no end. If no time is provided then the reservation is forever. If start time is after end time then the period defined is null. The effective period is the union of all listed periods.

3.1.2 DaysOfWeekInMonthInYearIntersectionTe

Qualified Name:
TemporalExpression::TemporalExpressionModel::TemporalExpressionAugments::DaysOfWeekInMonthInYearIntersectionTe

This expression collects together calendar and clock values.
The intersection of each stated property forms the temporal expression.
Note that some properties are unions withing the property.
This can be used to generate complex expressions such as..
15:00 - 16:00 and 17:00 - 18:00 on the first and third Monday and Wednesday of January and April.
This class is AggLeaf.

This class is Experimental.

Table 2: Attributes for DaysOfWeekInMonthInYearIntersectionTe

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
dayOfWeekUnion	Experimental	The days for which the expression applies. This is the union of all days listed.
periodDurationInDaysUnion	Experimental	Defines the start and end time within a day or across several days. This is the union of all periods listed.
namedDayPositionInMonthUnion	Experimental	The position of each listed dayOfWeek in the month. Is the union of all stated day positions. For some months LAST and FIFTH will be the same occurrence. Some months will not have a FIFTH occurrence of some days.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
monthPhasing	Experimental	Identifies a start month and whether every month, every other month etc. Interacts with specificMonth. The months chosen should be from the intersection of monthPhasing and specificMonth.
specificMonthUnion	Experimental	Identifies specific months. This is the union of all stated months. Note all properties in the Te grouping this property are combined by intersection. For example, this property interacts with monthPhasing such that the months chosen should be from the intersection of monthPhasing and specificMonth.
weekInMonth	Experimental	Identifies a week start day and which weeks of the month are included. The structure identifies weeks from the start of the month and from the end of the month. The structure also allows for whole weeks and for partial weeks counting.

3.1.3 IncorporatedTe

Qualified Name:
TemporalExpression::TemporalExpressionModel::TemporalExpressionCore::IncorporatedTe

Provides rules for incorporation of a referenced temporal expression.
An incorporated temporal expression is combined with the temporal expression element that owns the incorporated TE.
Note that this fragment of the model is early experimental.
This class is AggLeaf.

This class is Experimental.

Table 3: Attributes for IncorporatedTe

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
incorporationMethod	Experimental	The method for incorporation of the temporal expression.
_incorporatedTe	Experimental	The temporal expression to be incorporated.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_exclusionConflictActionAlternative	Experimental	Any relevant conflict actions. This model area requires further work.

3.1.4 IterativeIntersectionTe

Qualified Name: TemporalExpression::TemporalExpressionModel::TemporalExpressionAugments::IterativeIntersectionTe

Once all unions and intersections have been performed the result is in terms of active periods.
The IterativeTe defines the number of active periods that will be run.
This is bounded by the boundingPeriodTe.
This class is AggLeaf.

This class is Experimental.

Table 4: Attributes for IterativeIntersectionTe

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
numberOfIterations	Experimental	Once all unions and intersections have been performed the result is in terms of active periods. The number of iterations is the number of active periods that will be run.
iterationDelay	Experimental	Number of defined iterations of period that must occur prior to a period being considered active.

3.1.5 ParticularDateTimePerPeriodIntersectionTe

Qualified Name: TemporalExpression::TemporalExpressionModel::TemporalExpressionAugments::ParticularDateTimePerPeriodIntersectionTe

Defines the period(s) of activity in a year.
This class is AggLeaf.

This class is Experimental.

Table 5: Attributes for ParticularDateTimePerPeriodIntersectionTe

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
sameDatesPerYearUnion	Experimental	Defines a period of activity in a year. Is the union of all listed periods.
description	Experimental	States the purpose of the period.

3.1.6 PhasedPeriodIntersectionTe

Qualified Name:
TemporalExpression::TemporalExpressionModel::TemporalExpressionAugments::PhasedPeriod IntersectionTe

Applied from the beginning of each continuous period. The continuous period is defined by period, daysOfWeek, periodInDay, DayInMonth and RangeEachYear.
Phasing starts at the beginning of the continuous period for the stated duration then repeated after phasing value from the beginning of the previous phasing.
If there are no phasing statement then the active period is the whole of the continuous period (every hour of every day of every week of every year, duration 60 minutes).
If no continuous period statement then phasing applies to a specific period by intersection where the intersection is defined as starting at the beginning of the period of the other intersecting element(s).
The phasings can be accumulated (essentially formed by an intersection of the properties).
For example:
- hourPhasing of 0.5 and dayPhasing of 2.0 and duration of 15 minutes means 15 minutes for every 30 minutes for a whole day every other day.
- hourPhasing of 0.5 and dayPhasing of 2.5 and duration of 15 minutes means 15 minutes for every 30 minutes for a whole day then the same for a day starting a day and a half after the end of the first period.
This class is AggLeaf.

This class is Experimental.

Table 6: Attributes for PhasedPeriodIntersectionTe

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
hourPhasing	Experimental	Phasing starts at the beginning of the continuous period for the stated duration then repeated after hourPhasing hours from the beginning of the previous phasing. A value of 1.0 means every hour. For example if the continuous period starts at 15:03 on a particular day, the duration is 90 minutes and the hourPhasing is 2.5, then - the first phase period will start at 15:03 and will last to 16:33 - the second phase period will start at 17:33 and will last to 19:03 - the third phase period will start at 20:03 and will last to 21:33 - etc. until the end of the continuous period.
dayPhasing	Experimental	Phasing starts at the beginning of the continuous period for the stated duration then repeated after dayPhasing days from the beginning of the previous phasing. A value of 1.0 means every day. For example if the continuous period starts at 15:03 on a particular day which happens to be a Monday, the duration is 90 minutes and the dayPhasing is 2.5, then - the first phase period will start at 15:03 and will last to 16:33 on that Monday - the second phase period will start at 3:03 on Wednesday and will last to 4:33 on that Wednesday - the third phase period will start at 15:03 on Friday and will last to 16:33 on that Friday - etc. until the end of the continuous period.
weekPhasing	Experimental	Phasing starts at the beginning of the continuous period for the stated duration and is then repeated after weekPhasing weeks from the beginning of the previous phasing. A value of 1.0 means every week. For example if the continuous period starts at 15:03 on a particular day which happens to be a Monday, the duration is 90 minutes and the weekPhasing is 2.5, then - the first phase period will start at 15:03 and last to 16:33 on that Monday - the second phase period will start at 3:03 on Friday and will last to 4:33 on that Friday two weeks later - the third phase period will start at 15:03 on Monday 5 weeks after the first Monday and last to 16:33 on that Monday - etc. until the end of the continuous period.
yearPhasing	Experimental	Phasing starts at the beginning of the continuous period for the stated duration then repeated after yearPhasing days from the beginning of the previous phasing. A value of 1.0 means every year. Note the challenge with leap years where the fraction of one year may not be equal to the fraction of the next.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
periodDurationInMinutes	Experimental	Duration in minutes.

3.1.7 SimplePhasedPeriodIntersectionTe

Qualified Name:
TemporalExpression::TemporalExpressionModel::TemporalExpressionAugments::SimplePhasedPeriodIntersectionTe

There is a natural intersection of periods such that there is only activity if all stated periods indicate activity.
Hence if a second phasing indicates activity every other second and a minute phasing every other minute then there is only activity every other second in every other minute.
This class is AggLeaf.

This class is Experimental.

Table 7: Attributes for SimplePhasedPeriodIntersectionTe

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
second	Experimental	Second phasing within the minute boundary.
minute	Experimental	Minute phasing within the hour boundary.
hour	Experimental	Hour phasing within the day boundary.
day	Experimental	Day phasing within the week boundary.
week	Experimental	Week phasing within the year boundary.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
year	Experimental	Year phasing within the boundingPeriod.

3.1.8 TeElement

Qualified Name:
TemporalExpression::TemporalExpressionModel::TemporalExpressionCore::TeElement

The definition of the time constraints of the temporal expression.
The temporal expression definition is formed by taking the intersection of all augmenting temporal expressions.
Note that the augmenting temporal expression has "intersection" in its name.

Inherits properties from:

- LocalClass

This class is AggLeaf.

This class is Preliminary.

Table 8: Attributes for TeElement

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
description	Preliminary	A description of the TeElement.
relevance	Experimental	The essence of the element. For example it may represent a set of non-work days. This is a free form string so relevance is not a formal parameter.
timeZone	Preliminary	Identifies the timezone that was used to construct the element. This is especially relevant when considering periods related to days as the day boundary in one timezone is in the middle of the day in another.
dstDates	Experimental	Indicates whether the time zone has Daylight Saving Time and provides the relevant detail.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_incorporatedTeReference	Experimental	Referencing TeElements that are combined with this TeElement in a TemporalExpression.

3.1.9 TemporalExpression

Qualified Name: TemporalExpression::TemporalExpressionModel::TemporalExpressionCore::TemporalExpression

The temporal expression defines the existence or operation of some entity or duration of some state, value, range, etc. of some property.
A number of properties and/or entities may follow the same temporal expression.

Inherits properties from:

- GlobalClass

This class is AggRoot.

This class is Preliminary.

Table 9: Attributes for TemporalExpression

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
description	Preliminary	A human readable summary of the temporal expression.
_teElement	Preliminary	An expression representing some periods of time as part of the temporal expression.

3.2 Temporal Expression Data Types

This section provides the temporal expression data types.

3.2.1 MonthAndDay

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::MonthAndDay

Calendar date.

Table 10: Attributes for MonthAndDay

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
month	Experimental	Month name.
dayInMonth	Experimental	Day number in month. Valid entry depends upon the specific month. An integer greater than the number of days in the month should be taken as the last day of the month. For example: - 32 January should be taken as 31 January - 93 February should be taken as 29 or 28 February depending upon whether the year is leap year or not.

3.2.2 MonthPhasing

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::MonthPhasing

Phasing across months in a year.

Table 11: Attributes for MonthPhasing

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
phaseAlignMonth	Experimental	Start month of the phasing.
monthPhase	Experimental	Specific month phasing.

3.2.3 PeriodDuration

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::PeriodDuration

Defines the time period across daysOfWeek.

Table 12: Attributes for PeriodDuration

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
endTime	Experimental	The time is the end moment of the identified hour/minute, so 23:59 is the moment before 00:00 of the next day. No endTime means end of day unless periodDuration is stated.
startTime	Experimental	The time is the beginning moment of the identified hour/minute. So 00:00 is the moment after the end of the 59th minute in the 23rd hour of the previous day. No startTime means start of day unless periodDuration and endTime are stated.
periodDuration	Experimental	Interacts with startTime and endTime. If no endTime is stated then periodDuration is number of minutes from the startTime. If endTime stated but no startTime stated then periodDuration is the number of minutes before the endTime. If startTime, endTime and periodDuration are all stated then the result is the intersection of time definitions startTime to endTime and startTime for periodDuration (i.e., whichever is shorter). If not present then assumes defined by startTime/endTime. The periodDuration may take the time beyond the end of the day or before the beginning of the day (see daySpan).
daySpan	Experimental	The daySpan is used to provide a boundary of the startTime/endTime/periodDuration. If daySpan <1 there will be no period. If daySpan =1 then there will only be a period if startTime/endTime/periodDuration does not go beyond a day boundary. This period will apply for each of the days of the week listed. If daySpan >1 then a period will be triggered within each listed day of the week and may cross into a day not listed. Via this structure it is possible to have a temporal expression that indicates a continuous period of many days starting on a specific day. Periods may merge and hence become continuous over several period definitions.

3.2.4 PeriodInAYear

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::PeriodInAYear

This may be combined with other structures as an intersection.
However, this is an unlikely approach.
More likely there will be an exclusion rule.
This is equivalent to BoundingPeriod with no year defined, hence repeats every year.
Can be intersected with other structures. If no other structures intersected then the period is continuous.

Table 13: Attributes for PeriodInAYear

Attribute Name	**Lifecycle Stereotype (empty = Mature)**	**Description**
startDayInMonth	Experimental	Specific day in year that the event starts.
startTimeInStartDay	Experimental	Defines the time in the start day that the period starts. The time is the start moment of the identified hour/minute, so 00:00 is the moment that the day starts, i.e., the moment after the moment that the previous day ends.
endDayInMonth	Experimental	Specific day in year that the event ends.
endTimeInEndDay	Experimental	Defines the time in the end day that the period ends. The time is the end moment of the identified hour/minute, so 23:59 is the moment before 00:00 of the next day.
periodDurationInMinutes	Experimental	An alternative to endDay/endTime. Number of minutes from the startDayInMonth and startTimeInDay. If not present then assumes defined by endDay/endTime..
accountForLeapYear	Experimental	Indicates whether the period should account for leap years.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
description	Experimental	Description of the period.

3.2.5 Phasing

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::Phasing

Table 14: Attributes for Phasing

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
gap	Experimental	Indicates the gap between active periods. The value: - 0 means repeated every period (no gap) - 1 means gap of one period (and hence, if activityDuration is 1, every other period) - etc. If gap is used with repeatPeriod then there is guaranteed gap between active periods.
offset	Experimental	The offset of the start of the active period from the start of the next larger period. So for minutes, offset of 1 means start periodicity 1 minute after the hour.
activityDuration	Experimental	The length of the active period.
repeatPeriod	Experimental	Period of activity repeat. An activity of duration 1 and repeatPeriod 1 is essentially continuous. The repeatPeriod is an alternative to gap.
variableActivityDuration	Experimental	The duration of activity is at least the activityDuration but may be more up to the repeatPeriod boundary.

3.2.6 TimeInDay

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::TimeInDay

Clock time as per 24 hour clock.

Table 15: Attributes for TimeInDay

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
hourInDay	Experimental	Clock time hours.
minuteInHour	Experimental	Clock time minutes.

3.2.7 TimePeriodOptions

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::TimePeriodOptions

This data type specifies a time period.

Table 16: Attributes for TimePeriodOptions

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
startTime	Experimental	This attribute defines the start time of the time period.
stopTime	Experimental	This attribute defines the stop time of the time period.

3.2.8 WeekInMonth

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::WeekInMonth

Table 17: Attributes for WeekInMonth

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
weekStartDay	Experimental	The start day of the week (defining the week boundary).
weekPositionInMonth	Experimental	All relevant week positions in the month.

3.3 Enumeration Types

3.3.1 DayPositionInMonth

3.3.1.1 DayPositionInMonth

Qualified Name:
TemporalExpression::TemporalExpressionModel::DataTypes::DayPositionInMonth

There are a number of occurrences of each named day in a month.
This property identifies which of the occurrences should be active.

Applied stereotypes:

- Experimental

Contains Enumeration Literals:

- FIRST:
  - The first occurrence of the specific named day in the month (which may be the first day of the month).
  - Applied stereotypes:
    - Experimental
- SECOND:
  - The second occurrence of the specific named day in the month.
  - Applied stereotypes:
    - Experimental
- THIRD:
  - The third occurrence of the specific named day in the month.

    - Applied stereotypes:
        - Experimental
- FOURTH:
    - The fourth occurrence of the specific named day in the month.
    - Applied stereotypes:
        - Experimental
- FIFTH:
    - The fifth occurrence of the specific named day in the month.
    - Applied stereotypes:
        - Experimental
- LAST:
    - The last occurrence of the specific named day in the month.
    - Applied stereotypes:
        - Experimental
- LAST_BUT_ONE:
    - The last but one occurrence of the specific named day in the month.
    - Applied stereotypes:
        - Experimental
- ALL_DAY_POSITIONS:
    - All day positions in the month.
    - Applied stereotypes:
        - Experimental

3.3.2 IncorporationMethod

3.3.2.1 IncorporationMethod

Qualified Name:
TemporalExpression::TemporalExpressionModel::DataTypes::IncorporationMethod

Applied stereotypes:
- Experimental

Contains Enumeration Literals:
- UNION:
    - The temporal expressions to be combined are unioned (essentially added together) such that the expression corresponds to all time periods referenced.
    - Applied stereotypes:
        - Experimental
- INTERSECTION:
    - The temporal expressions to be combined are intersected such that the resulting temporal expression includes only time periods that referenced by both definitions.
    - Applied stereotypes:
        - Experimental
- INTERSECT_COMPLEMENT:

- The resulting temporal expression will not include the definition of the referenced TemporalExpression, i.e., Intersection with the complement of the referenced TemporalExpression.
  Using a simple integer equivalent as an example, if an integer list expression is 1, 3, 5, 7, 9 and the expression to be incorporated via INTERSECTION_COMPLEMENT is 3, 4, 5 then the result will be 1, 7, 9.
  This is because the complement of 3, 4, 5 is 1, 2, 6, 7, 8, 9, 10, 11, ... and the intersection of 1, 2, 6, 7, 8, 9, 10, 11,... with 1, 3, 5, 7, 9 is 1, 7, 9.
- Applied stereotypes:
  - Experimental

3.3.3 LeapYear

3.3.3.1 LeapYear

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::LeapYear

How to deal with leap years.

Applied stereotypes:
- Experimental

Contains Enumeration Literals:
- ONLY_LEAP_YEAR:
  - Only applies during leap years.
  - Applied stereotypes:
    - Experimental
- NOT_LEAP_YEAR:
  - Does not apply to leap years.
  - Applied stereotypes:
    - Experimental
- ANY_YEAR:
  - Not sensitive to leap year.
  - Applied stereotypes:
    - Experimental

3.3.4 Month

3.3.4.1 Month

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::Month

Month of the year.

Applied stereotypes:
- Preliminary

Contains Enumeration Literals:

- JANUARY:
  - January.
  - Applied stereotypes:
    - Preliminary
- FEBRUARY:
  - February.
  - Applied stereotypes:
    - Preliminary
- MARCH:
  - March.
  - Applied stereotypes:
    - Preliminary
- APRIL:
  - April.
  - Applied stereotypes:
    - Preliminary
- MAY:
  - May.
  - Applied stereotypes:
    - Preliminary
- JUNE:
  - June.
  - Applied stereotypes:
    - Preliminary
- JULY:
  - July.
  - Applied stereotypes:
    - Preliminary
- AUGUST:
  - August.
  - Applied stereotypes:
    - Preliminary
- SEPTEMBER:
  - September.
  - Applied stereotypes:
    - Preliminary
- OCTOBER:
  - October.
  - Applied stereotypes:
    - Preliminary
- NOVEMBER:
  - November.
  - Applied stereotypes:
    - Preliminary

- DECEMBER:
  - December.
  - Applied stereotypes:
    - Preliminary
- ALL_MONTHS:
  - All months in the year.
  - Applied stereotypes:
    - Experimental

3.3.5 MonthPhase

3.3.5.1 MonthPhase

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::MonthPhase

Whether the event occurs every month or on some other basis.

Applied stereotypes:
- Experimental

Contains Enumeration Literals:
- EVERY_MONTH:
  - All months (no gaps).
  - Applied stereotypes:
    - Experimental
- EVERY_OTHER_MONTH:
  - A one month gap.
    The named month will be the first month.
    The next month will not be active then the following month will be etc.
  - Applied stereotypes:
    - Experimental
- EVERY_THREE_MONTHS:
  - A two month gap.
    The named month will be the first month.
    The next two months will not be active then the following month will be etc.
  - Applied stereotypes:
    - Experimental
- EVERY_FOUR_MONTHS:
  - A three month gap.
    The named month will be the first month.
    The next three months will not be active then the following month will be etc.
  - Applied stereotypes:
    - Experimental
- EVERY_SIX_MONTHS:

- A five month gap.
  The named month will be the first month.
  The next five months will not be active then the following month will be etc.
- Applied stereotypes:
  - Experimental

3.3.6 TemporalDirection

3.3.6.1 TemporalDirection

Qualified Name: TemporalExpression::ExclusionConflictAction::TemporalDirection

With respect to normal flow of time.

Applied stereotypes:

- Experimental

Contains Enumeration Literals:

- NEXT:
  - Forward in time to the next valid period.
  - Applied stereotypes:
    - Experimental
- PREVIOUS:
  - Reverse in time to the previous valid period.
  - Applied stereotypes:
    - Experimental
- NEXT_AND_PREVIOUS:
  - Either forward or backward in time to the next valid period.
  - Applied stereotypes:
    - Experimental

3.3.7 TimeZone

3.3.7.1 TimeZone

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::TimeZone

List of all standard time zones.

Applied stereotypes:

- Experimental

Contains Enumeration Literals:

- UTC_PLUS_1400:
  - UTC +14.
  - Applied stereotypes:

        - Experimental
- UTC_PLUS_1345:
    - UTC +13:45.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_1300:
    - UTC +13.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_1200:
    - UTC +12.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_1100:
    - UTC +11.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_1030:
    - UTC +10:30.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_1000:
    - UTC +10.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0930:
    - UTC +9:30.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0900:
    - UTC +9.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0845:
    - UTC +8:45.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0800:
    - UTC +8.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0700:
    - UTC +7.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0630:

    - UTC +6:30.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0600:
    - UTC +6.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0545:
    - UTC +5:45.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0530:
    - UTC +5:30.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0500:
    - UTC +5.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0430:
    - UTC +4:30.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0400:
    - UTC +4.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0330:
    - UTC +3:30.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0300:
    - UTC +3.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0200:
    - UTC +2.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0100:
    - UTC +1.
    - Applied stereotypes:
        - Experimental
- UTC_PLUS_0000:
    - UTC +0.
    - Applied stereotypes:

        - Experimental
- UTC_MINUS_0100:
    - UTC -1.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0200:
    - UTC -2.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0300:
    - UTC -3.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0330:
    - UTC -3:30.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0400:
    - UTC -4.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0500:
    - UTC -5.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0600:
    - UTC -6.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0700:
    - UTC -7.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0800:
    - UTC -8.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0900:
    - UTC -9.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_0930:
    - UTC -9:30.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_1000:

    - UTC -10.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_1100:
    - UTC -11.
    - Applied stereotypes:
        - Experimental
- UTC_MINUS_1200:
    - UTC -12.
    - Applied stereotypes:
        - Experimental

3.3.8 WeekDay

3.3.8.1 WeekDay

Qualified Name: TemporalExpression::TemporalExpressionModel::DataTypes::WeekDay

Named day of week.

Applied stereotypes:

- Preliminary

Contains Enumeration Literals:

- ALL_DAYS_OF_THE_WEEK:
    - All the days of the week.
    - Applied stereotypes:
        - Preliminary
- MONDAY:
    - Monday.
    - Applied stereotypes:
        - Preliminary
- TUESDAY:
    - Tuesday.
    - Applied stereotypes:
        - Preliminary
- WEDNESDAY:
    - Wednesday.
    - Applied stereotypes:
        - Preliminary
- THURSDAY:
    - Thursday.
    - Applied stereotypes:
        - Preliminary
- FRIDAY:
    - Friday.

    - Applied stereotypes:
        - Preliminary
- SATURDAY:
    - Saturday.
    - Applied stereotypes:
        - Preliminary
- SUNDAY:
    - Sunday.
    - Applied stereotypes:
        - Preliminary

3.3.9 WeekPositionInMonth

3.3.9.1 WeekPositionInMonth

Qualified Name:
TemporalExpression::TemporalExpressionModel::DataTypes::WeekPositionInMonth

Week position in month recognizing that week and month boundaries are not necessarily coincident.

Applied stereotypes:
- Experimental

Contains Enumeration Literals:
- FIRST_WEEK_IN_MONTH:
    - The month start boundary occurs in the week or at the start of the week.
      At least one day of the week is in the month.
      For example, the last day of the week is the first day of the month.
    - Applied stereotypes:
        - Experimental
- FIRST_WHOLE_WEEK_IN_MONTH:
    - The month start boundary occurs at or before the week start boundary but there are less than seven days between the month start and week start.
      All days of the week are in the month.
      This may be the same as the FIRST_WEEK_IN_MONTH, but is more often the SECOND_WEEK_IN_MONTH.
    - Applied stereotypes:
        - Experimental
- SECOND_WEEK_IN_MONTH:
    - The week directly after the FIRST_WEEK_IN_MONTH.
    - Applied stereotypes:
        - Experimental
- SECOND_WHOLE_WEEK_IN_MONTH:
    - The week directly after the FIRST_WHOLE_WEEK_IN_MONTH.
    - Applied stereotypes:

        - Experimental
- THIRD_WEEK_IN_MONTH:
    - The week directly after the SECOND_WEEK_IN_MONTH.
    - Applied stereotypes:
        - Experimental
- THIRD_WHOLE_WEEK_IN_MONTH:
    - The week directly after the SECOND_WHOLE_WEEK_IN_MONTH.
    - Applied stereotypes:
        - Experimental
- FOURTH_WEEK_IN_MONTH:
    - The week directly after the THIRD_WEEK_IN_MONTH.
    - Applied stereotypes:
        - Experimental
- FOURTH_WHOLE_WEEK_IN_MONTH:
    - The week directly after the THIRD_WHOLE_WEEK_IN_MONTH.
    - Applied stereotypes:
        - Experimental
- FIFTH_WEEK_IN_MONTH:
    - The week directly after the FOURTH_WEEK_IN_MONTH.
    - Applied stereotypes:
        - Experimental
- SIXTH_WEEK_IN_MONTH:
    - The week directly after the FIFTH_WEEK_IN_MONTH.
    - Applied stereotypes:
        - Experimental
- LAST_WEEK_IN_MONTH:
    - The month end boundary occurs in the week or at the end of the week.
      At least one day of the week is in the month.
      For example, the first day of the week is the last day of the month.
    - Applied stereotypes:
        - Experimental
- LAST_WHOLE_WEEK_IN_MONTH:
    - The month end boundary occurs at or after the end of week boundary but there are less than seven days between the month end and the end of the week.
      All days of the week are in the month.
    - Applied stereotypes:
        - Experimental
- ALL_WEEKS_IN_MONTH:
    - All weeks in the month.
    - Applied stereotypes:
        - Experimental

3.4 Further detail

Some of the structures use in the model have a somewhat complex interaction. These are discussed in the sections below.

3.4.1 Period duration

The periodDurationInDays has several interacting properties. The interaction is explained below.

Table 18: periodDurationInDays property interaction

startTime explicit	endTime explicit	periodDuration explicit	daySpan explicit	Meaning
Don't care	Don't care	None	0	No period
None	None	None	None	Starts at beginning of listed day and ends at end of listed day.
None	None	None	1	Starts at beginning of listed day and ends at end of listed day
None	None	None	2+	Starts at beginning of listed day and ends at end of day 1+ days later
Specific	None	None	1..*	Starts at specific time in day listed and ends at end of day as per daySpan
None	Any specific	None	1..*	Starts at start of day listed and ends at specific time as per daySpan.
Before end	After start	None	1	Simple duration within each day of week listed
After end	Before start	None	1	Starts at

startTime explicit	endTime explicit	periodDuration explicit	daySpan explicit	Meaning
				beginning of day of week. Ends at end time starts again at start time and ends at end of day
Specific	Specific	None	2+	Starts at start time in day of week listed ends 2+ days layer at end time.
Specific	Don't care	Specific	Don't care	Starts at start time in day of week listed and stops after duration.
None	Specific	Specific	Don't care	Starts duration before end time in day listed and stops at end time in day listed.
None	None	Specific	Don't care	Starts at start of day listed and ends after duration

4 Examples of Combination

The followings sections illustrate usage of the structure for an arbitrary set of definitions.

4.1 Union within a property

Consider the following examples:

- dayOfWeekUnion (in DaysOfWeekInMonthInYearIntersectionTe): if the list has MONDAY and TUESDAY then this means for both Monday and Tuesday.
- periodUnion (in BoundingPeriodIntersectionTe): If there are two members of the list:
    - Start: 1 January 2023 00:00:00 (Sunday) End: 31 March 2023 23:59:59 (Friday)
    - Start 1 June 2023 00:01:00 (Thursday) and End: 30 June 2023 22:59:59 (Friday)

  Then the expression is "active" for both periods, but clearly not between 1 April 2023 00:00:00 to 1 June 2023 00:00:59

4.2 Intersection between properties in augments

Take the properties above. If these were applied to a TeElement, the intersection results in every Monday and Tuesday between 1 January 2023 00:00:00 (Sunday) End: 31 March 2023 23:59:59 as well as between 1 June 2023 00:01:00 (Thursday) 30 June and 2023 22:59:59 (Friday).

So, this would include Monday 2 January, Tuesday 3 January, Monday 9 January etc. as well as Monday 5 June, Tuesday 6 June etc.

Now consider properties from PhasedPeriodIntersectionTe. If hourPhasing is set to 2 and periodDurationInMinutes is set to 15, the intersection of these two means the first 15 minutes every other hour throughout each continuous period starting at the beginning of the period[1].

If this is also applied to the TeElement this would result in the TeElement being valid for Monday 1 January 00:00:00 to 00:14:59 (last moment of), then 02:00:00 to 02:14:59 through Monday and similarly through Tuesday as that is part of the continuous period. This would be the same for each following Monday/Tuesday pair.

All other properties intersect in a similar way.

4.3 Union of TeElements into a TemporalExpression

Consider two TeElements, one as defined above and another that was built similarly but was for Tuesday and Wednesday from 1 January 2023 00:00:00 to 31 January 2023 23:59:59 with a PhasedPeriodIntersectionTe of every hour for 30 minutes. If these two TeElements are applied to the same TemporalExpression, TeA, via TeHasUnionOfPeerElements association, this would result in Monday being exactly as defined in the previous section.

[1] note that the current structure only allows the phasing to start at the beginning of a continuous period. The model allows for further augments of structures that allow greater flexibility (assuming that the general union/intersection pattern is followed). A structure could be developed similar to the current PhasedPeriodIntersectionTe that has offset structures that can “delay” the start after the continuous period boundary for each of the phasings.

But the Tuesdays in January would be active for the first 30 minutes of every hour as would the Wednesdays. From the end of January only Monday and Tuesday would be active for the first 15 minutes of every other hour as before.

Again, all other properties can be treated in this way.

4.4 Applying an IncorporatedTe

Consider a further TemporalExpression (TeB) that is defined with one TeElement and no BoundingPeriodIntersection so that it applies for all time. If that TeElement has (in DaysOfWeekInMonthInYearIntersectionTe):

- specificMonthUnion of JANUARY
- dayOfWeekUnion indicating TUESDAY
- namedDayPositionInMonthUnion indicating SECOND

This TemporalExpression (TeB) is active for the whole of the second Tuesday (i.e., the first moment of 00:00:00 to the last moment of 23:59:59) in January (for any year).

Consider another TemporalExpression (TeC) that has a TeElement that is defined for all time (i.e., has no constraints), but also has two IncorporatedTes.

The first has incorporationMethod set to INTERSECTION and refers to TeA and the second has incorporationMethod set to INTERSECT_COMPLEMENT and refers to TeB.

The complement of TeB is active for all time other than the whole of the second Tuesday in January (when it is inactive). When the intersection of this with TeA is formed this results in the same definition as the raw TeA other than for the second Tuesday in January (10 January) when the result is inactive.

So TeC would be active as defined for TeA for each Tuesday in January other than 10 January when it would be inactive.

5 Using the temporal model

5.1 Application to elements of the domain model

Temporal variations can occur at any level of a model:

- An instance may be created at a particular time then deleted at another
- An instance may become visible at a particular time then vanish at another time and then reappear later etc.
- A property of an instance can take a new value at a particular time.
- A property can appear in an instance at a particular time
- A member of a list can change position at a particular time
- Etc.

Distilling from the above, there appear two distinct considerations:

- Presence/existence (at a place)
- Value

Value at a time can be stated in terms of solely the value that has changed in the context of a sparce positional structure or in the context of other unchanging close values. For example, the value of one element of a list may change, this could be expressed in isolation as a new value against the list position or could be expressed as part of the whole list of values (in a sparce positional structure).

Presence is more complex as it requires:

- create and remove (and potentially hide and expose) of an identified unit, including structure and creation of an identifier, at a place in an existing structure
- move and replication of an identified unit (and potentially change of identifier of a unit, e.g., where the identification is the position and there is a requirement to change the position).

As a consequence, in the most complex cases, the temporal model statements will apply at many/all levels of the model. It may also be necessary to separate presence from value for all levels

The figure below shows a generalized representation of the application/positioning of the temporal expression.

It is expected that at each application there will be a single referenced temporal expression that will, as necessary, incorporate other temporal expressions to form the appropriate complexity of temporal expression.

There may be several distinct changes over time such that there are several distinct representations of the structure/values, each with its own temporal expression.

There may be a statement of value/structure that applies to any time where stated temporal expressions are not true.

It appears that there is a need for an idempotent add (i.e., "add if not already present" as opposed to "add another") and corresponding remove.

There is a clear challenge adding and removing by list position as opposed to invariant id as there may be race conditions with competing adds and removes.

The expression could be in terms of presence/absence after a specific time as opposed to add/remove at a moment in time.

The model could be extended with considerations of time to achieve and an indication of jeopardy if a temporal expectation cannot be met.

- E.g., If a connectivity-service is not realized for a period then is to be realized for a period and the time to realize is longer than the time left before the service should be realized.
- E.g., if a future realization (with resources already acquired) is failed (operational state disabled) and the time to repair is longer than the time remaining before the schedule becomes active.

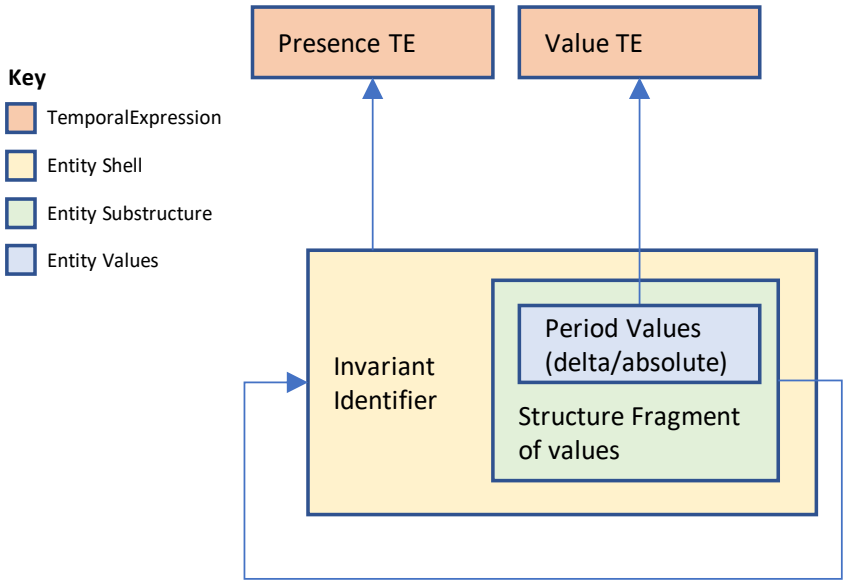


Figure 5-1 TemporalExpresion being applied to an entity – Pattern

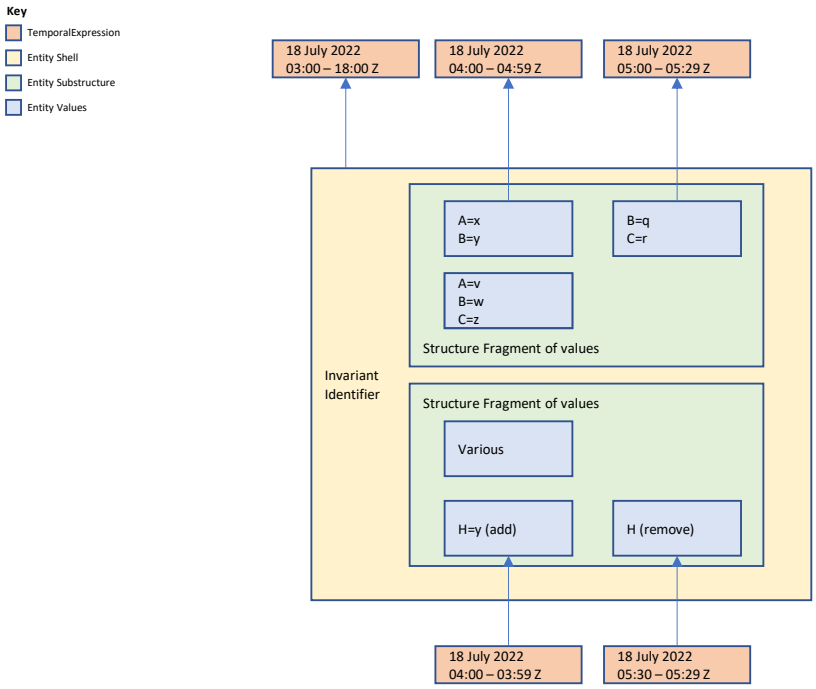


Figure 5-2 TemporalExpresion being applied to an entity – Rough Expansion

5.2 Application to deal with plan deviations and plan alternatives

The temporal model is applied to the modeling of both actual history and possible future. Considering possible future, there are several degrees:

- Committed future
- Temporary deviation from commitment
- Alternative futures

The following subsections consider each of the above.

5.2.1 Committed future

This is a single thread of time where, at any particular point in time, there is no conflict between resource usages in the same way there is no conflict in usage in live solution, i.e., each unit of resource can be used only once.

Unlike the live solution, some resources may not have a defined state (distinct from known state, the resource must be in a state even if it is not known). The representation of committed future provides a set of constraints that restrict what is intended.

The committed future is a statement of a progression of outcomes that is a realization solution to the outcomes requested, usually with more detail that was provided in the request. It is possible to:

- delay the specification of resource configuration state to the moment when it is required
- plan explicit resource configuration state ahead of time

In both cases there is a chance that the resources will not be available at the moment they are required to support the desired outcome, however, in many cases an appropriately detailed plan that is continually evaluated for achievability will provide a greater likelihood of success than a delayed specification where resources may just not be available. Clearly, continual evaluation comes at a cost.

5.2.2 Temporary deviation from commitment

This can apply:

- "Now"
- From some point in the future

The deviation may have a clear end date/time or may have uncertainty in its end (and possibly even its start). The deviation may be fully defined or may have temporal variation with uncertainty etc. Clearly, there may be a deviation from the deviation etc.

Any deviation will have some associated definition of progression from non-deviated state to deviated state and also back to non-deviated state.

5.2.3 Alternative futures

Beyond any commitment, there may be some knowledge of potential futures and there may be several potential futures each of which may only be partially defined and each of which will probably have temporal variation with uncertainty.

As time progresses, some part of the alternative futures will become a commitment "collapsing" much of the uncertainty into a specific set of details. Eventually that committed future will become the present (and then the past) and will be fully resolved into real instances.

6 Specific examples

This section sets out examples of use of the temporal expression for common everyday activities (as opposed to telecoms scheduling) to help clarify the model usage in a familiar context.

6.1 Garden Waste Collection

This example uses several of the structures to show how a regular waste collection activity may be suspended for a week due to a public holiday. It is likely that the waste collection would be reschedules in the week of the holiday. This is not detailed in the example, but the opportunity is highlighted.

The figure below sets an instance model.

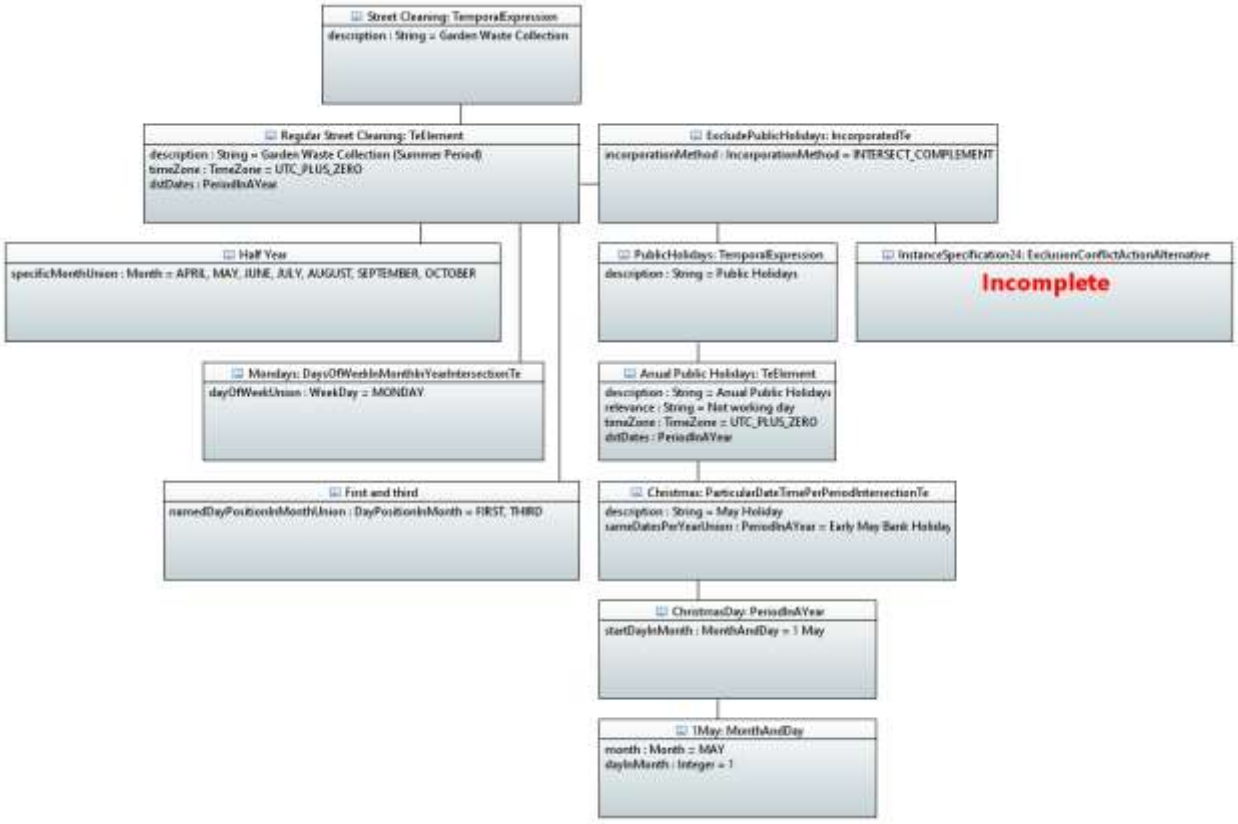


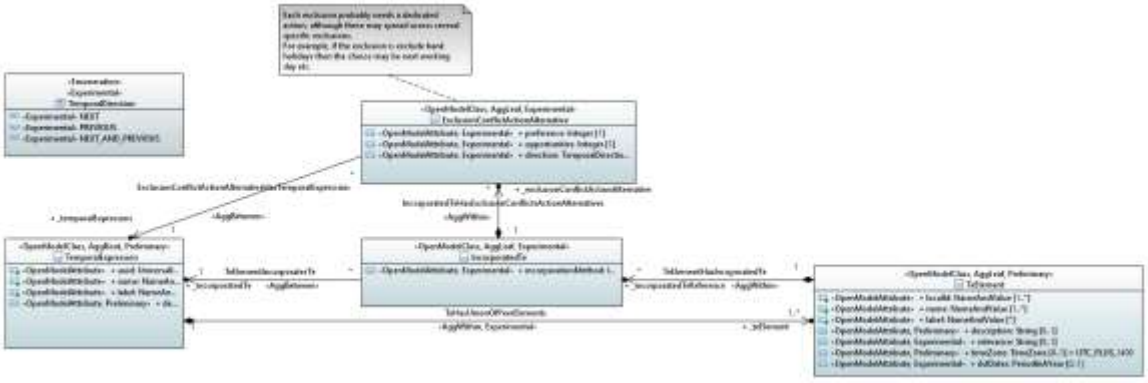
Figure 6-1 Garden waste collection example

7 Further work

Further work will be carried out in this area in a subsequent release of the model.

7.1 Exclusion conflict action

One area touched on in the model is that of exclusion conflict action. This covers the cases where there is a need for regularity that is prevented due to some other schedule (for example, no refuse collection on public holidays). There is a need to reschedule an activity at some other time that provides a close-to-regular behavior. This reschedule may need alternative to allow for other conflicts etc. The model includes an early sketch of a solution.



CoreModel diagram: TemporalExpression-ExclusionConflictActionAlternative

Figure 7-1 Exclusion conflict action alternative

7.1.1 ExclusionConflictActionAlternative

Qualified Name:
TemporalExpression::ExclusionConflictAction::ExclusionConflictActionAlternative

The rules that apply when an exclusion from one part of a temporal expression causes disruption to a fundamental sequence from another part of the temproal expression.
Note that this class is early experimental. The usage detail needs to be developed further.
This class is AggLeaf.

This class is Experimental.

Table 19: Attributes for ExclusionConflictActionAlternative

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
preference	Experimental	It is possible that a specific conflict action cannot overcome the conflict. There may be several alternative conflict actions. The conflict actions should be run in preference order with the lowest value integer being the most preferred and hence the first to run. Same preference value actions will be run in an arbitrary order.
opportunities	Experimental	Indicates how many alternatives in the temporal direction can be tried. For example, if the exclusion is of a public holiday Monday and the normal occurrence is Monday then the alternative action will activate. If the referenced TemporalExpression offers every week day and temporal direction is NEXT, then the first opportunity will be Tuesday. If the opportunities is set to 1 then that is all that may be tried. But it may not be available for some other exclusion reason and hence no alternative will be available and the period will be skipped. However, if the opportunities is set to 2 then the next but one day can also be tried (i.e., Wednesday in the example) and that may be available. And so on...

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
direction	Experimental	Indicates in which temporal direction the conflict action applies.
_temporalExpression	Experimental	The temporal expression that deals with the conflict providing an alternative to the necessary regular cycle that was disrupted.

End of Document